

monatlich

Compute mit

COMMODORE & SCHNEIDER

6/86

VC-64, VC-20, C-16/116, CPC-464

3,80 DM
33 öS
3,80 sFr

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

Aktuell +++ Aktuell +++ Aktuell +++ Aktuell

Endlich ist es soweit!

Ab sofort auch alle C-16/116 mit Checksummen. Bei richtiger Eingabe der Programmzeilen sehen Sie immer am Ende der Zeile eine Checksumme, die Ihnen über evtl. Fehler der Programmzeile Auskunft gibt. Nähere Erläuterungen zur Handhabung und zum Einsatz des Checksummers finden Sie auf S. 17.

Programmfehler haben keine Chance mehr.



Hobbytronic '86

Was war auf der diesjährigen Hobbytronic in Dortmund zu sehen? Wir waren für Sie vor Ort



und haben uns ein wenig umgesehen.

Zahlreiche Besucher waren auf dem Ausstellungsgelände um sich Anregungen im Hobby-Bereich zu holen oder um ganz einfach zu sehen, mit welchem Hobby man seine Freizeit sinnvoll nutzen kann. Wer seine Freizeit mit Computern verbringt, der sollte auf Seite 44 weiterlesen.

Der C-64 im neuen Gewand

Die Gerüchte wollten nicht verstummen, daß Commodore seine 64'er Produktion gedrosselt habe, um den Verkauf des PC 128 anzukurbeln. Tatsächlich war über Ostern so gut wie kein C-64 mehr zu bekommen, weil Commodore seine Lager leergehäumt hatte und in den Produktionsstätten die Bänder für einen neuen C-64 umgestellt wurden. Einen neuen C-64 also, der allerdings in seinen Funktionen ganz der alte ist, lediglich in einem neuen Gewand. Aus produktionstechnischen Gründen wurde allerdings die Platine modifiziert und modernisiert. Für den Benutzer sollen sich dadurch allerdings keinerlei Veränderungen ergeben. Auch das Gerücht, es würde ein zusätzliches Betriebssystem integriert, erwies sich als falsch. Wahr ist, das nachgeladen werden muß und dem Benutzer eine GEM-ähnliche Arbeitsstruktur bietet. Genaueres veröffentlichen wir in der nächsten Ausgabe, wenn wir ein Testgerät in den Händen gehabt haben.

Leser- + Meckerecke - Report - Software-Abo - Software-Service
Bücher - Werkstatt - Assembler-Kurs - Software-Reviews - Tips & Tricks
Eddy - Affenwurf - Karate-Kid - Q-Bert - Starcommand - u. v. m.

sie
ist
da...

die neue Generation

Ausgabe März/April, Nr. 3/86 - 6S 50 / sfr 6,- / DM 6,-

aktueller software Markt

ERSTE COMPUTER-SOFTWARE FACHZEITSCHRIFT TESTS UND VORSTELLUNGEN

Über **100** Programme **IM TEST!**

• Action Games

• Adventure

• Anwender

Fundgrube

**12 Seiten
Kleinanzeigen**

Do it yourself



Branchenlösung
für
Friseure

Mitmachen

SCHACHECKE



Quiz

CRI

Reise ins Software-
Land »England«
zu gewinnen

Autoren
**Was müssen Programm-
Autoren beachten?**

Gefragte Software

- Textverarbeitung
- Basic-Compiler
- Spiele-Generator
- Profi-Printer
- Sekretärin

- 100 Seiten Programme im Test
- der Software-Markt auf einen Blick
- speziell gesuchte Software kann direkt beim Hersteller bestellt werden
- 12 Seiten Kleinanzeigen als Fundgrube
- für alle Anwender von Heimcomputern

Tronic-Verlag GmbH • Am Stad 35 • 3440 Eschwege • Telefon: 05651 / 3 00 11



der Computer-Zeitschriften

Report

Was gibts Neues auf dem Software-Markt	26
Hobbytronic '86	44
Amstrad kauft die Vertriebsrechte für Sinclair	45
Probierwochen	45

Bücher

Computer – mal anders betrachtet	54
Computerspiele und Computergrafik	54
Computergrafische Experimente mit Pascal	55

Tips & Tricks

für C 16/116	20
für Schneider	48

Kleinanzeigen

	14
--	----

Software-Service

	32
--	----

Assembler-Kurs

Teil 13	39
---------	----

Werkstatt

Music-Master für C-64	9
Invers-Utility C-16/116	20
Data-Maker für VC-20	30
Disksort für CPC 464	60

Software

Commodore

The little Puck goes home (C-64)	3
Eddy (C-64)	6
Checksummer OC für C-16/116	17
Affenwurf (C-16/116)	18
Lohnsteuerjahresausgleich (C-16/116)	22
Karate-Kid (VC-20)	28
Q-Bert (VC-20)	35

Schneider

Mini-Forth	41
Starcommand	50

Die nächste Ausgabe von

Compute mit

liegt wieder ab

16. Juni

bei Ihrem Zeitschriftenhändler
für Sie bereit.

Impressum

„Compute mit“
erscheint monatlich im Tronic-Verlag, 3440 Eschwege

Redaktion:

Axel Credé (verantwortlich)
Chefredakteur: Hartmut Wendt
Siegfried Görk, Manfred Kleimann, Bernd Zimmermann
Frank Brall, Otfried Schmidt, Uwe Knerim

Gesamtherstellung:

Druckhaus Dierichs Kassel, Frankfurter Str. 168, 3500 Kassel

Vertrieb:

Inland (Groß-, Einzel- und Buchhandelsbuchhandel) sowie Österreich und Schweiz
Verlagsunion
Friedrich-Bergius-Straße 20
6200 Wiesbaden
Telefon 06121/2660

Anfragen nicht an Vertrieb oder Druckerei, sondern nur an den Verlag!

Erscheinungsweise:

Erstverkaufstag von „Compute mit“ jeweils Mitte des Monats.

Urheberrecht:

Alle in „Compute mit“ veröffentlichten Beiträge sind urheberrechtlich geschützt. Alle Rechte, auch Übersetzungen, vorbehalten.

Reproduktionen jeder Art (Fotokopien, Microfilm, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.

Aus ihrer Veröffentlichung kann nicht geschlossen werden, daß die beschriebenen Lösungen oder Bezeichnungen frei von Schutzrechten sind.

Bezugspreis:

Einzelheft 3,80 DM
Abonnement: Inland 42,- DM im Jahr (12 Ausgaben)
Ausland: Europa 52,- DM
ohne Kassetten!

Autoren, Manuskripte:

Der Verlag nimmt Manuskripte und Software zur Veröffentlichung gerne entgegen.

Sollte keine andere Vereinbarung getroffen sein, so gehen wir davon aus, daß Sie mit einem Honorar von 120,- DM pro abgedruckter Seite im Heft einverstanden sind.

Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger.

Rücksendung erfolgt nur gegen Erstattung der Kosten. Zusendungen von Software zur Veröffentlichung sollten folgendes enthalten:

Kopierfähige Kassette oder Diskette mit dem Programm (Computer-Bezeichnung), von Drucker erstelltes Listing oder Serie von Bildschirmfotos (keine Schreibmaschinenlistings), evtl. Bildschirmfotos von einem Probelauf und ausführliche Programmbeschreibung (Erklärung der Variablenliste, Beschreibung des Bildschirmaufbaues, Farbe, Grafik usw.). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

Anzeigenpreise:

Bitte Mediaunterlagen anfordern.

Anzeigenverwaltung:

Tronic-Verlag GmbH,
Am Stad 35, 3440 Eschwege
Telefon 05651/30011
Telefax 05651/30014

Anzeigenleitung:

Inland: H. Wendt
Ausland: M. Kleimann

Geschicklichkeitsspiel mit Action:

Die erlebnisreiche Wanderung des „Little Puck“ durch die abenteuerlichen Wege und Straßen, Gräben und Höhlen

The little Puck goes home

mit dem Commodore 64

Ladeanweisungen:

Tippen Sie zuerst das Hexdata-Ladeprogramm ab und speichern Sie dieses. Geben Sie danach im Direktmodus

POKE43,1:POKE44,78:POKE19968,0:NEW

ein und starten Sie das Ladeprogramm mit RUN. Am Ende jeder Data-Zeile steht zum Erkennen von Tippfehlern eine Prüfsumme. Evtl. Prüfsummenfehler werden mit der Zeilennummer ausgegeben. Treten keine Fehler auf, wird das erzeugte Maschinenprogramm auf den gewünschten Massenspeicher (Geräteadresse in Zeile 300 beachten!) abgesaved. Dieses

Maschinenprogramm kann nun jederzeit mit LOAD"PUCK (16400)" Geräteadresse,1 geladen und mit SYS16400 gestartet werden.

Spielgeschehen:

Der Spieler schlüpft in die Rolle des kleinen Puck, der auf dem Bildschirm gut an seinen roten Haaren zu erkennen ist. Es gilt, diesen kleinen Puck, begleitet von einer schönen Melodie, durch Höhlen und Gräben nach Hause zu führen. Dabei muß er Vögeln, Spinnen, Pilzen u.v.m. ausweichen und darauf achten, daß er beim Springen nicht an seinen Kopf stößt. Um von einem Bild ins Nächste zu gelangen, muß der Spieler den kleinen Puck von

der Ausgangsposition links zum rechten Bildschirmrand führen. Haben Sie das geschafft, ohne eines der drei Leben des kleinen Puck zu verlieren, erscheinen noch weitere 9 Level. Ab dem 10. Bild beginnt das Spiel von vorn, allerdings mit erhöhter Geschwindigkeit.

Steuerung:

Zur Steuerung des kleinen Puck benötigen Sie einen Joystick in Port 2. Bei Druck auf den Feuerknopf erscheint das Titelbild und das Spiel kann beginnen.

Und nun viel Spaß beim Abtippen!!!

```

10 REM *****
20 REM *
30 REM * THE LITTLE PUCK GOES HOME *
40 REM *
50 REM *****
60 :
70 REM (C)1986 BY PATRICK URBAN
80 :
90 :
100 DIMH(75):FORI=0TO9
110 H(48+I)=I:H(65+I)=I+10:NEXT
120 FORI=15808TO19759:READA$
130 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))
140 D=H(H)*16+H(L):S=S+D:POKEI,D
150 A=A+1:IFA<20THENNEXT:A=-1
160 PRINT"ZEILE:";1000+Z;
170 READV:Z=Z+1:IFV=STHEN190
180 PRINT"PRUEFSUMMENFEHLER (SPACE) !";999
+Z
185 STOP
190 IFA<0THEN300
200 S=0:A=0:PRINT:NEXT
250 :
280 :
300 SYS57812 "PUCK (SPACE) (16400)",B
310 REM GERAETEADRESSE ^
320 POKE175,77:POKE174,47

```

```

<15>
<117>
<110>
<137>
<55>
<118>
<208>
<138>
<148>
<149>
<171>
<1>
<190>
<29>
<113>
<243>
<49>
<84>
<73>
<233>
<195>
<52>
<83>
<108>
<33>
<125>
330 POKE194,61:POKE193,192
340 SYS62954 : REM SAVE-ROUTINE
350 END
400 :
450 :
480 :
1000 DATA 00,14,00,00,65,00,01,55,40,01,
D7,40,01,D7,40,01,55,40,05,65, 1087
1001 DATA 50,15,56,54,59,55,55,57,FE,D9,
67,BF,D5,55,55,65,55,96,55,15, 2207
1002 DATA 55,54,04,14,10,04,14,10,04,14,
10,04,14,10,04,14,10,04,14,10, 409
1003 DATA 04,14,10,79,00,14,00,00,65,00,
01,55,40,01,D7,40,01,D7,40,01, 993
1004 DATA 55,40,05,65,50,15,56,54,59,55,
55,57,FE,D9,67,BF,D5,55,55,65, 2121
1005 DATA 55,96,55,15,55,54,10,41,04,10,
41,04,10,41,04,10,41,04,10,41, 931
1006 DATA 04,10,41,04,10,41,04,C9,00,00,
00,00,00,00,00,00,00,00,00,00, 375
1007 DATA 00,00,00,00,00,00,00,00,00,00,01,
05,50,05,15,55,19,55,55,55,6A, 583
1008 DATA A9,05,2A,A0,00,2A,A0,00,0A,A0,
00,0A,A0,00,02,A0,00,02,A0,00, 1242
1009 DATA 00,80,00,00,00,00,00,00,00,00,
00,EB,AA,AA,AA,AA,AA,AA,AA,AA, 1720
1010 DATA AA,AA,AA,AA,AA,AA,AA,AA,AA,AA,
AA,AA,AA,AA,AA,AA,AA,AA,AA, 3400

```


programme

[illegible]

1048 DATA 1D,1D,3E,20,20,3E,20,20,3E,20,20,50,20,55,20,43,20,4B,20,20, 903	<33>
1049 DATA 3C,20,20,3C,20,20,3C,0D,1F,11,11,11,1D,1D,1D,1D,1D,1D, 623	<255>
1050 DATA 1D,1D,1D,1D,1D,1D,1D,1D,4F,4F,45,53,20,48,4F,4D,45,0D,90,11, 1037	<55>
1051 DATA 11,11,11,11,11,1D,1D,1D,1D,1D,1D,1D,1D,28,43,29,31,39,38,36, 681	<181>
1052 DATA 20,42,59,20,50,41,54,52,49,43,4B,20,55,52,42,41,4E,8D,55,41, 1492	<64>
1053 DATA 20,D2,FF,EB,E0,8C,D0,FS,A9,00,8D,13,D4,A9,F0,8D,14,D4,A9,0A, 3048	<187>
1054 DATA 85,FB,85,FD,A9,05,85,A4,A9,83,85,A5,A9,00,85,FC,A9,10,8D,12, 2785	<180>
1055 DATA D4,AD,1E,D0,AD,1F,D0,A9,00,8D,1E,D0,8D,1F,D0,AD,00,DC,C9,70, 2669	<146>
1056 DATA B0,F9,4C,FS,45,AD,10,D0,29,01,F0,0F,AD,00,D0,C9,2C,90,14,A9, 2466	<11>
1057 DATA 2B,8D,00,D0,4C,47,42,AD,00,D0,C9,14,B0,05,A9,15,8D,00,D0,A5, 2092	<13>
1058 DATA FC,C9,01,D0,03,4C,46,43,C9,04,D0,03,4C,61,43,C9,02,D0,03,4C, 2024	<116>
1059 DATA 82,43,C9,05,D0,03,4C,AF,43,C9,03,D0,03,4C,DD,43,C9,06,D0,03, 2129	<136>
1060 DATA 4C,0B,44,AD,00,DC,C9,71,B0,0B,A2,11,8E,12,D4,4C,07,43,C9,7B, 2065	<254>
1061 DATA B0,03,4C,BE,42,C9,7C,B0,03,4C,91,42,60,AD,00,D0,D0,0B,AD,10, 2184	<158>
1062 DATA D0,49,01,8D,10,D0,CE,00,D0,C6,FB,A5,FB,D0,0D,A9,0A,85,FB,A9, 2879	<211>
1063 DATA FD,8D,FB,87,20,ED,42,60,C9,05,D0,05,A9,FC,8D,FB,07,60,AD,00, 2585	<52>
1064 DATA D0,C9,FF,D0,0B,AD,10,D0,49,01,8D,10,D0,EE,00,D0,C6,FB,A5,FB, 3027	<124>
1065 DATA D0,0D,A9,0A,85,FB,A9,FF,8D,FB,07,20,ED,42,60,C9,05,D0,05,A9, 2623	<187>
1066 DATA FE,8D,FB,07,60,A9,32,8D,0B,D4,A9,00,8D,0D,D4,A9,01,8D,0C,D4, 2396	<34>
1067 DATA A9,80,8D,0B,D4,A9,81,8D,0B,D4,60,AE,01,D0,86,FD,C9,68,B0,0C, 2682	<166>
1068 DATA A9,FF,8D,FB,07,A9,02,85,FC,4C,82,43,C9,6C,B0,0C,A9,FD,8D,FB, 2957	<91>
1069 DATA 07,A9,03,85,FC,4C,DD,43,A9,01,85,FC,AD,01,D0,49,FF,8D,0F,D4, 2561	<204>
1070 DATA A9,10,8D,12,D4,A9,11,8D,12,D4,CE,01,D0,60,CE,01,D0,AD,01,D0, 2421	<201>
1071 DATA 49,FF,8D,0F,D4,38,A5,FD,E9,1E,18,CD,01,D0,D0,04,A9,04,85,FC, 2641	<160>
1072 DATA 60,EE,01,D0,AD,01,D0,49,FF,8D,0F,D4,A5,FD,CD,01,D0,A5,FD,CD, 3076	<38>
1073 DATA 01,D0,D0,09,A9,00,85,FC,A9,10,8D,12,D4,60,CE,01,D0,AD,00,D0, 2428	<252>
1074 DATA C9,FF,D0,0B,AD,10,D0,49,01,8D,10,D0,EE,00,D0,AD,01,D0,49,FF, 2664	<118>
1075 DATA 8D,0F,D4,A5,FD,38,E9,1E,18,CD,01,D0,D0,04,A9,05,85,FC,60,EE, 2648	<63>
1076 DATA 01,D0,AD,00,D0,C9,FF,D0,0B,AD,10,D0,49,01,8D,10,D0,EE,00,D0, 2544	<184>
1077 DATA AD,01,D0,49,FF,8D,0F,D4,A5,FD,CD,01,D0,D0,09,A9,00,85,FC,A9, 2850	<211>
1078 DATA 10,8D,12,D4,60,CE,01,D0,AD,01,D0,49,FF,8D,0F,D4,AD,00,D0,D0, 2565	<209>
1079 DATA 0B,AD,10,D0,49,01,8D,10,D0,CE,00,D0,A5,FD,38,E9,1E,18,CD,01, 2225	<87>
1080 DATA D0,D0,04,A9,06,85,FC,60,EE,01,D0,AD,00,D0,D0,0B,AD,10,D0,49, 2590	<155>
1081 DATA 01,8D,10,D0,AD,01,D0,49,FF,8D,0F,D4,CE,00,D0,A5,FD,CD,01,D0, 2690	<253>
1082 DATA D0,09,A9,00,85,FC,A9,10,8D,12,D4,60,13,12,20,2A,20,54,4B,45, 1791	<21>
1083 DATA 20,4C,49,54,54,4C,45,20,50,55,43,4B,20,2A,20,20,4C,49,56,45, 1275	<79>
1084 DATA 53,20,20,20,20,4C,45,56,45,4C,20,20,20,20,A9,93,20,D2,FF,A9, 1697	<57>

1085 DATA 1E,20,D2,FF,A2,00,86,A6,A9,11, 20,D2,FF,A6,A6,E6,A6,E0,0F,D0, 2847	<226>
1086 DATA F3,A2,00,86,A6,A9,12,20,D2,FF, A9,A6,20,D2,FF,A6,A6,E6,A6,E0, 3173	<73>
1087 DATA FF,D0,F3,A9,00,85,A6,A9,A6,20, D2,FF,A6,A6,E6,A6,E0,8E,D0,F3, 3551	<193>
1088 DATA A2,00,86,A6,8D,34,44,20,D2,FF, E6,A6,A6,A6,E0,2A,D0,F2,A2,00, 2874	<41>
1089 DATA 86,A6,A9,A6,20,D2,FF,E6,A6,A6, A6,E0,28,D0,F3,A5,A5,8D,1C,04, 3078	<146>
1090 DATA A9,E6,8D,E7,07,A9,05,8D,E7,DB, A9,00,8D,10,D0,85,FC,A9,15,8D, 2793	<33>
1091 DATA 00,D0,A9,FF,8D,F8,07,A9,80,8D, 01,D0,A9,01,8D,15,D0,8D,17,D0, 2587	<21>
1092 DATA 8D,1C,D0,8D,1D,D0,8D,1B,D0,A9, 00,8D,25,D0,8D,1F,D0,A9,02,8D, 2378	<150>
1093 DATA 26,D0,A9,08,8D,27,D0,A9,10,8D, 12,D4,60,A9,10,8D,12,D4,A9,11, 2205	<8>
1094 DATA 8D,12,D4,20,3F,45,20,3F,45,EE, 01,D0,AD,01,D0,49,FF,8D,0F,D4, 2224	<36>
1095 DATA AD,01,D0,C9,AA,D0,EB,C6,A5,A9, 10,8D,12,D4,A9,00,85,FC,60,A4, 2926	<209>
1096 DATA A4,A2,FF,CA,D0,FD,88,D0,F8,60, 20,5E,44,AD,1E,D0,A9,B3,8D,26, 3064	<127>
1097 DATA 04,A9,FB,8D,FA,07,A9,95,8D,05, D0,A9,5B,8D,04,D0,A9,05,8D,15, 2443	<232>
1098 DATA D0,8D,1C,D0,A9,01,8D,29,D0,A9, 00,85,A6,8D,1E,D0,4C,9E,45,A5, 2460	<17>
1099 DATA A6,D0,0F,EE,04,D0,AD,04,D0,C9, FF,D0,14,A9,01,85,A6,60,CE,04, 2683	<44>
1100 DATA D0,AD,04,D0,C9,5A,D0,05,A9,00, 85,A6,60,60,A9,00,8D,1E,D0,20, 2337	<93>
1101 DATA 25,42,20,7B,45,20,3F,45,AD,1E, D0,29,01,F0,25,20,11,45,A5,A5, 1669	<2>
1102 DATA C9,80,D0,0E,AD,1E,D0,A9,00,8D, 15,D0,8D,1E,D0,4C,3D,41,AD,1E, 2333	<119>
1103 DATA D0,A9,00,8D,15,D0,8D,1E,D0,4C, 4A,45,20,7B,45,20,3F,45,AD,10, 1922	<86>
1104 DATA D0,F0,0A,AD,00,D0,C9,2B,D0,03, 4C,3F,47,20,3F,45,4C,9E,45,20, 2003	<97>
1105 DATA 5E,44,A9,B1,8D,26,04,A9,07,8D, 15,D0,8D,17,D0,8D,1C,D0,A9,A9, 2324	<250>
1106 DATA 8D,03,D0,8D,05,D0,A9,0D,8D,2B, D0,8D,29,D0,A9,64,8D,02,D0,A9, 2456	<220>
1107 DATA E6,8D,04,D0,A9,00,8D,1E,D0,8D, 1E,D0,A9,FA,8D,F9,07,8D,FA,07, 2724	<61>
1108 DATA A9,00,8D,1E,D0,20,25,42,20,3F, 45,20,3F,45,20,3F,45,AD,1E,D0, 1586	<23>
1109 DATA 29,01,F0,27,20,11,45,A5,A5,C9, B0,D0,0E,AD,1E,D0,A9,00,8D,15, 2110	<204>
1110 DATA D0,8D,1E,D0,4C,E1,41,AD,1E,D0, A9,00,8D,15,D0,A9,00,8D,1E,D0, 2451	<100>
1111 DATA 4C,F3,45,AD,10,D0,C9,01,D0,0A, AD,00,D0,C9,2B,D0,03,4C,83,46, 2318	<54>
1112 DATA 4C,30,46,20,5E,44,A9,B2,8D,26, 04,A9,03,8D,15,D0,8D,1C,D0,A9, 2006	<62>
1113 DATA 09,8D,28,D0,A9,F9,8D,F9,07,AD, 1E,D0,A9,00,8D,1E,D0,A9,FF,8D, 2737	<18>
1114 DATA 02,D0,A9,5A,8D,03,D0,A9,00,85, A7,4C,F0,46,AD,02,D0,D0,10,AD, 2456	<78>
1115 DATA 10,D0,49,02,8D,10,D0,AD,1E,D0, A9,00,8D,1E,D0,CE,02,D0,A5,A7, 2371	<7>
1116 DATA D0,0F,CE,03,D0,AD,03,D0,C9,50, D0,13,A9,01,85,A7,60,EE,03,D0, 2547	<132>
1117 DATA AD,03,D0,C9,82,D0,04,A9,00,85, A7,60,20,25,42,A9,00,8D,1E,D0, 2175	<95>
1118 DATA 20,B6,46,20,3F,45,20,B6,46,20, 3F,45,20,B6,46,20,3F,45,AD,1E, 1547	<128>
1119 DATA D0,29,01,F0,1A,20,11,45,AD,1E, D0,A9,00,8D,1E,D0,8D,15,D0,A5, 2128	<111>
1120 DATA A5,C9,B0,D0,03,4C,E1,41,4C,83, 46,AD,10,D0,29,01,F0,0A,AD,00, 2258	<192>
1121 DATA D0,C9,2B,D0,03,4C,4A,45,4C,F0, 46,20,5E,44,A9,0F,8D,15,D0,8D, 2157	<37>

1122 DATA 1C,D0,A9,0D,8D,17,D0,8D,29,D0,8D,2A,D0,A9,B4,8D,26,04,A9,09, 2281	<210>
1123 DATA 8D,2B,D0,A9,F9,8D,F9,07,A9,FA,8D,FA,07,8D,FB,07,A9,5A,8D,03, 2823	<56>
1124 DATA D0,A9,FF,8D,02,D0,A9,64,8D,04,D0,A9,E6,8D,06,D0,A9,A9,8D,05, 2843	<141>
1125 DATA D0,8D,07,D0,AD,1E,D0,A9,00,8D,1E,D0,A9,00,8D,1E,D0,20,25,42, 2206	<183>
1126 DATA 20,B6,46,20,3F,45,20,B6,46,20,3F,45,20,B6,46,20,3F,45,20,3F,45,AD,1E, 1547	<136>
1127 DATA D0,29,01,F0,1A,20,11,45,AD,1E,D0,A9,00,8D,1E,D0,8D,15,D0,A5, 2128	<119>
1128 DATA A5,C9,B0,D0,03,4C,3D,41,4C,3F,47,AD,10,D0,29,01,F0,0A,AD,00, 2027	<58>
1129 DATA D0,C9,2B,D0,03,4C,DF,47,4C,90,47,20,5E,44,20,E8,47,4C,A0,48, 2161	<74>
1130 DATA A2,00,86,A6,A9,11,20,D2,FF,E6,A6,A6,A6,E0,0D,D0,F3,A9,90,20, 2906	<206>
1131 DATA D2,FF,A9,12,20,D2,FF,A2,00,86,A6,A9,20,20,D2,FF,E6,A6,A6, 3037	<25>
1132 DATA E0,2B,D0,F3,A9,13,20,D2,FF,A9,11,20,D2,FF,A9,11,20,D2,FF,A9, 2935	<134>
1133 DATA 12,20,D2,FF,A2,00,86,A6,A9,20,20,D2,FF,E6,A6,A6,A6,E0,A0,D0, 2995	<143>
1134 DATA F3,60,AD,06,D0,D0,10,AD,10,D0,49,0B,8D,10,D0,AD,1E,D0,A9,00, 2373	<96>
1135 DATA 8D,1E,D0,CE,06,D0,A5,AB,D0,0A,A9,F7,8D,FB,07,A9,01,85,AB,60, 2732	<11>
1136 DATA A9,F8,8D,FB,07,A9,00,85,AB,60,AD,0B,D0,D0,10,AD,10,D0,49,10, 2481	<58>
1137 DATA 8D,10,D0,AD,1E,D0,A9,00,8D,1E,D0,CE,0B,D0,A5,A9,D0,0A,A9,F7, 2714	<20>
1138 DATA 8D,FC,07,A9,01,85,A9,60,A9,F8,8D,FC,07,A9,00,85,A9,60,20,3A, 2442	<106>
1139 DATA 48,4C,6A,4B,A9,00,85,AB,85,A9,A9,95,8D,07,D0,8D,09,D0,A9,32, 2349	<121>
1140 DATA 8D,06,D0,D0,A9,AA,8D,0B,D0,A9,B5,BD,26,04,A9,07,8D,2A,D0,8D,2B, 2335	<167>
1141 DATA D0,A9,F7,8D,FB,07,8D,FC,07,A9,0B,8D,10,D0,A9,19,8D,15,D0,8D, 2670	<83>
1142 DATA 1C,D0,A9,00,8D,1E,D0,A9,00,8D,1E,D0,20,25,42,20,9A,4B,20,3F, 1820	<180>
1143 DATA 45,20,9A,48,20,3F,45,20,3F,45,AD,1E,D0,29,01,F0,1A,20,11,45, 1492	<26>
1144 DATA AD,1E,D0,A9,00,8D,1E,D0,8D,15,D0,A5,A5,C9,B0,D0,03,4C,3D,41, 2449	<52>
1145 DATA 4C,DF,47,AD,10,D0,29,01,F0,0A,AD,00,D0,C9,2B,D0,03,4C,67,49, 2147	<219>
1146 DATA 4C,DF,48,92,20,DF,A9,20,20,20,20,DF,A9,20,20,20,DF,A9,20, 2013	<105>
1147 DATA 20,20,20,DF,A9,20,20,20,20,DF,A9,20,20,20,DF,A9,20,20,DF,A9,20,20, 1624	<207>
1148 DATA 20,DF,A9,20,A2,00,86,A6,8D,2B,49,20,D2,FF,E6,A6,A6,A6,E0,29, 2713	<44>
1149 DATA D0,F2,60,20,5E,44,20,E8,47,A9,B6,8D,26,04,20,54,49,A9,95,8D, 2257	<39>
1150 DATA 07,D0,A9,FF,8D,06,D0,A9,F7,8D,FB,07,A9,09,8D,15,D0,8D,1C,D0, 2734	<153>
1151 DATA A9,07,8D,2A,D0,AD,1F,D0,AD,1E,D0,A9,00,8D,1E,D0,8D,1F,D0,A9, 2487	<14>
1152 DATA 00,8D,1E,D0,8D,1F,D0,20,25,42,20,3A,48,20,3F,45,20,3A,48,20, 1414	<244>
1153 DATA 3F,45,20,3A,48,20,3F,45,AD,1E,D0,29,01,F0,1A,20,11,45,AD,1E, 1498	<202>
1154 DATA D0,A9,00,8D,1E,D0,8D,15,D0,A5,A5,C9,B0,D0,03,4C,3D,41,4C,67, 2425	<38>
1155 DATA 49,AD,1F,D0,29,01,F0,20,20,11,45,AD,1F,D0,AD,1E,D0,A9,00,8D, 2050	<168>
1156 DATA 1E,D0,8D,1F,D0,8D,15,D0,A5,A5,C9,B0,D0,03,4C,3D,41,4C,67,49, 2360	<222>
1157 DATA AD,10,D0,29,01,F0,0A,AD,00,D0,C9,2B,D0,03,4C,1B,4A,4C,9F,49, 2007	<171>
1158 DATA 20,5E,44,20,E8,47,A9,B7,8D,26,04,A9,0B,8D,15,D0,8D,1C,D0,A9, 2160	<242>


```

1159 DATA F9,8D,F9,07,A9,F7,8D,FB,07,A9,
09,8D,2B,DO,A9,07,8D,2A,DO,A9, 2764 <119>
1160 DATA 00,85,A6,85,A7,85,A8,A9,B4,8D,
02,DO,A9,FF,8D,06,DO,A9,95,8D, 2838 <200>
1161 DATA 07,DO,A9,5A,8D,03,DO,A9,00,8D,
1E,DO,A9,00,8D,1E,DO,20,25,42, 2057 <125>
1162 DATA 20,3A,48,20,3F,45,20,
B6,46,20,3A,48,20,3F,45,20,B6, 1434 <211>
1163 DATA 46,20,3F,45,AD,1E,DO,29,01,FO,
1A,20,11,45,A9,00,8D,15,DO,8D, 1751 <50>
1164 DATA 1E,DO,AD,1E,DO,A5,A5,C9,B0,DO,
03,4C,3D,41,4C,1B,4A,AD,10,DO, 2340 <90>
1165 DATA 29,01,FO,0A,AD,00,DO,C9,2B,DO,
03,4C,B5,4A,4C,60,4A,20,5E,44, 1899 <176>
1166 DATA A9,B8,8D,26,04,A9,0F,8D,15,DO,
BD,1C,DO,A9,F9,8D,F9,07,A9,FB, 2702 <190>
1167 DATA 8D,FA,07,A9,FA,8D,FB,07,A9,5A,
BD,03,DO,A9,FF,8D,02,DO,8D,04, 2747 <96>
1168 DATA DO,A9,95,8D,05,DO,A9,A9,8D,07,
DO,A9,14,8D,06,DO,A9,08,8D,10, 2452 <197>
1169 DATA DO,A9,09,8D,2B,DO,A9,01,8D,29,
DO,A9,0D,8D,2A,DO,A9,00,8D,1E, 2248 <175>
1170 DATA DO,85,A6,85,A7,85,A8,85,A9,A9,
01,85,A6,A9,09,8D,17,DO,A9,00, 2598 <142>
1171 DATA 8D,1E,DO,20,25,42,20,B6,46,20,
7B,45,20,3F,45,20,B6,46,20,3F, 1565 <249>
1172 DATA 45,20,B6,46,20,3F,45,AD,1E,DO,
29,01,FO,1A,20,11,45,A9,00,8D, 1664 <79>
1173 DATA 15,DO,8D,1E,DO,AD,1E,DO,A5,A5,
C9,B0,DO,03,4C,3D,41,4C,B5,4A, 2470 <41>
1174 DATA AD,10,DO,29,01,FO,0A,AD,00,DO,
C9,2B,DO,03,4C,6C,4B,4C,1A,4B, 1961 <28>
1175 DATA 20,5E,44,A9,B9,8D,26,04,A9,3F,
BD,15,DO,8D,1C,DO,A9,27,8D,17, 2082 <65>
1176 DATA DO,A9,FB,8D,FB,07,8D,FC,07,A9,
FA,8D,F9,07,8D,FA,07,8D,FD,07, 3039 <24>
1177 DATA A9,07,8D,2A,DO,8D,2B,DO,A9,0D,
BD,2B,DO,8D,29,DO,8D,2C,DO,A9, 2482 <36>
1178 DATA 95,8D,07,DO,8D,09,DO,A9,FF,8D,
06,DO,8D,0B,DO,A9,A9,8D,03,DO, 2689 <160>

```

```

1179 DATA 8D,05,DO,8D,0B,DO,A9,4B,8D,02,
DO,A9,A5,8D,04,DO,A9,FF,8D,0A, 2571 <122>
1180 DATA DO,A9,10,8D,10,DO,A9,00,8D,1E,
DO,85,A6,85,A7,85,A8,85,A9,4C, 2584 <201>
1181 DATA 44,4C,13,11,11,11,11,11,11,
11,9D,9D,12,96,20,20,11,9D,9D, 1176 <188>
1182 DATA 20,20,11,9D,9D,20,20,11,9D,9D,
20,20,11,9D,9D,20,20,11,9D,9D, 1580 <102>
1183 DATA 20,20,11,9D,9D,20,20,11,9D,9D,
20,20,13,1C,11,11,11,11,9D,20, 1158 <141>
1184 DATA 11,9D,9D,20,20,11,9D,9D,9D,20,
20,20,11,9D,9D,9D,9D,20,20, 1720 <201>
1185 DATA 20,9D,9D,11,11,1F,20,11,9D,20,
11,9D,20,11,9D,20,20,25,42,20, 1228 <78>
1186 DATA 9A,4B,20,3F,45,20,9A,4B,20,3F,
45,20,3F,45,AD,1E,DO,29,01,FO, 1669 <252>
1187 DATA 1A,20,11,45,A9,00,8D,15,DO,8D,
1E,DO,AD,1E,DO,A5,A5,C9,B0,DO, 2388 <122>
1188 DATA 03,4C,3D,41,4C,6C,4B,AD,10,DO,
29,01,FO,0A,AD,00,DO,C9,2B,DO, 1986 <7>
1189 DATA 03,4C,8B,4C,4C,44,4C,20,5E,44,
A9,B0,8D,26,04,A9,B1,8D,25,04, 1764 <115>
1190 DATA A2,00,86,A6,8D,E6,4B,20,D2,FF,
E6,A6,A6,A6,E0,32,DO,F2,A2,00, 3067 <51>
1191 DATA 86,A6,8D,18,4C,20,D2,FF,E6,A6,
A6,A6,E0,2C,DO,F2,A9,05,8D,15, 2868 <43>
1192 DATA DO,8D,1C,DO,A9,FF,8D,04,DO,A9,
95,8D,05,DO,A9,FB,8D,FA,07,A9, 3021 <204>
1193 DATA 01,8D,29,DO,85,A6,A9,00,8D,1E,
DO,A9,00,8D,1E,DO,20,25,42,20, 1953 <12>
1194 DATA 7B,45,20,3F,45,20,7B,45,20,3F,
45,20,3F,45,AD,1E,DO,29,01,FO, 1601 <14>
1195 DATA 1A,20,11,45,A9,00,8D,15,DO,8D,
1E,DO,AD,1E,DO,A5,A5,C9,B0,DO, 2388 <130>
1196 DATA 03,4C,3D,41,4C,8B,4C,AD,10,DO,
29,01,FO,0E,AD,00,DO,C9,2B,DO, 2022 <95>
1197 DATA 07,C6,A4,C6,A4,C6,F3,45,4C,DF,
4C,FF, 1749 <30>

```

Superspaß mit EDDY für alle C-64 Freaks

Für EDDY lohnt es wirklich, sich die Finger wund zu schreiben. Das Programm erinnert etwas an die Minerspiele, muß aber den Vergleich mit anderen Spielen, die in Basic geschrieben sind, nicht scheuen.

Nach dem Starten werden zunächst der Zeichensatz kopiert und die Sprites eingelesen. Wenn dann das Titelbild erscheint, kann man mit der F1-Taste den Level (1-10) einstellen. Mit der Feuertaste wird noch der Joystickport bestimmt und das Spiel beginnt.

EDDY muß nun in den verschiedenen Ebenen die zwölf Geldsäcke einsammeln, ohne die vorgegebene Zeit zu überschreiten. In den höheren Levels muß EDDY über Totenköpfe und Löcher springen, um an die Geldsäcke heranzukommen.



Gehen Sie mit 'Eddy' auf Punktejagd!

Durch die zufällige Verteilung der Totenköpfe entstehen immer wieder neue, knifflige Spielsituationen, die das ganze Geschick des Spielers erfordern.

Ist das Spielfeld abgeräumt, so wird der Rest der Zeit in Punkte gutgeschrieben. Hat er seine drei Leben verloren, ist das Spiel beendet.

Übrigens kann man mit der RUN/STOP Taste das Spiel unterbrechen und wieder fortsetzen.

Noch ein Tip: EDDY sollte so oft wie möglich springen. Er wird dadurch erheblich schneller.

```

0 REM *** EDDY *** BY J.RESE *** <229>
1 S=54272:V=53248:GOTO20 <181>
2 POKES+4,0:POKES+24,15:POKES+5,2:POKES+ <82>
6,0:POKES+1,30:POKES,1:POKES+4,129:RETUR <155>
N <170>
3 POKES+4,0:POKES+24,15:POKES+5,2:POKES+ <82>
6,0:POKES+1,50:POKES,1:POKES+4,129:RETUR <88>
N <138>
20 POKE788,52:IFPEEK(2)=255THEN80 <207>
30 POKE53280,6:POKE53281,6:PRINT" <58>
DOWN10 WHITE)", "(SPACE3)PLEASE(SPACE)WAI <198>
T" <75>
40 FORP=0T07:FORP=0T062:READA1:POKE(248+ <109>
P)*64+T,A1:NEXTT,P <22>
75 GOSUB9000 <179>
80 POKE53272,(PEEK(53272)AND240)OR12 <243>
90 S*="HOME RVSOFF DOWN11 RIGHT32 LIG.B <26>
REEN)" <151>
91 H*="HOME RVSOFF DOWN6 RIGHT32 LIG.BL <67>
UE)":GOTO5000 <72>
95 FORT=0T024:POKES+T,0:NEXT:POKES+24,15 <27>
POKES+5,9:POKES+6,100:POKES,1:POKES+4,3 <247>
3 <124>
96 FORT=0T02:FORE=20T080STEP2:POKES+1,E: <212>
NEXT:NEXT:POKES+6,0:SC=0 <216>
100 POKEV+16,0:POKEV+21,0:L0=0:L1=0:E1=0 <233>
R=R1:IFR1>10THENR=R1-10 <223>
110 QNRGDSUB800,810,820,830,840,850,860, <186>
870,880,890,900 <104>
111 IFR1>10THENF=0:IFR1>20THEN5000 <95>
120 A$="(RVSON)":FORT=0T014:A$=A$+CHR$(Z <101>
1)+CHR$(Z2):NEXT <204>
130 POKEV+33,0:POKEV+32,0:PRINT" <131>
CLEAR":TAB(30):"RVSON MGREY":* <180>
140 FORT=0T011:PRINT"DOWN LEFT2"& <122>
LEFT2":*":NEXT <140>
150 PRINT"HOME RVSOFF":TAB(33):"DOWN4 <68>
LIG.BLUE":HIGH: <22>
DOWN3":PRINTTAB(32):HI <205>
160 PRINTTAB(33):"DOWN2 LIG.GREEN":SCORE <95>
: <204>
DOWN:PRINTTAB(32):SC <131>
170 PRINTTAB(33):"DOWN2 LIG.RED":LEVEL: <180>
DOWN:PRINTTAB(33):R1 <122>
180 PRINTTAB(33):"DOWN2 WHITE":TIME: <68>
DOWN:PRINTTAB(33):"RVSON LIG.RED SPACE Y <22>
ELLOW SPACE2 GREEN SPACE3 RVSOFF": <205>
185 FORT=0T0V1:POKE1097+T*2,42:POKES+109 <95>
7+T*2,7:NEXT <186>
190 PRINT"HOME":POKE646,F:FORT=0T05:P <104>
RINTA$: <101>
DOWN3":NEXT:PRINTA$: <204>
200 FORT=0T010:IFX(T)=0THEN250 <131>
210 E=0:FORP=0T0D(T) <180>
220 POKEV(T)+E,107:POKEV(T)+E+5,7:POKEV <122>
(T)+E+1,115:POKEV(T)+E+5+1,7:E=E+40:NEXT <140>
230 NEXT <68>
250 IFLO=1THENGOSUB400 <22>
260 FORT=0T010:IFL(T)=0THEN300 <205>
270 FORE=0T02:POKEL(T)+E,32:NEXT:NEXT <95>
300 FORT=1144T01944STEP160:FORE=1T02 <186>
310 P=INT(RND(1)*29)+T:IFPEEK(P)<>32ORPE <122>
EK(P+40)=32THEN310 <140>
320 POKEP,0:POKEP+5,10:NEXT:NEXT <68>
330 FORT=1144T01944STEP160:FORE=1T0E1:IF <22>
E1=0THEN500 <205>
340 P=INT(RND(1)*29)+T:IFPEEK(P)<>32ORPE <95>
EK(P+40)=32THEN340 <186>
350 POKEP,94:POKEP+5,1:NEXT:NEXT:GOTO500 <122>
400 FORT=1184T01824STEP160:FORE=1T0L1 <205>
410 P=INT(RND(1)*27)+T:P0=PEEK(P):P1=PEE <95>
K(P+1):P2=PEEK(P+2) <99>
420 IFP0=107ORP1=107ORP2=107ORP0=115ORP1 <186>
=115ORP2=115THEN410 <144>
430 POKEP,32:POKEP+1,32:POKEP+2,32:NEXT: <144>
NEXT:RETURN
500 FORT=0T010:X(T)=0:L(T)=0:D(T)=0:NEXT
POKE2040,252:POKEV+1,221:POKEV,132

```

```

520 POKEV+28,1:POKEV+21,1:X=132:Y=221:C= <156>
1958:Z1=252:B=2040:F1=107:F2=115 <71>
550 M(0)=32:M(1)=101:M(2)=116:M(3)=117:M <222>
(4)=97:M(5)=246:M(6)=234:M(7)=231 <87>
560 M(8)=160:M=1902:M1=8:M2=0:K=0 <163>
590 FORT=0T024:POKES+T,0:NEXT:POKE198,0 <99>
600 U=0:X1=0:J=PEEK(J0):IF(JAND16)=0THEN <39>
GOSUB1000 <34>
610 IF(JAND8)=0ANDX<252THENX1=4:Z1=250:U <179>
=1:C=C+1:GOTO630 <244>
620 IF(JAND4)=0ANDX>20THENX1=-4:Z1=252:U <58>
=1:C=C-1 <30>
630 X=X+X1:POKEV,X:POKEB,Z1+U:IFU=1THENG <254>
OSUB3 <164>
640 IF(JAND2)=0THENGOSUB700 <187>
650 IF(JAND1)=0THENGOSUB750 <77>
660 M2=M2+1:IFM2/3=INT(M2/3)THENPOKEM,M <22>
(M1):M1=M1-1:IFM1<0THENM1=8:M=M-1 <248>
670 IF(PEEK(C+40)=32ANDPEEK(C)<>F1ANDPEE <177>
K(C)<>F2)ORPEEK(C)=940RM=1896THENGOTO700 <22>
0 <248>
680 X=X+X1:POKEV,X:POKEB,Z1:IFU=1THENGOS <177>
UB2 <22>
690 IFPEEK(C)=0THENPOKEC,32:SC=SC+20:GOS <248>
UB6000:PRINTB$:SC=K+K+1:IFK=12THEN6500 <177>
691 GETA$:IFA$="(CTRLC)"THENPOKE53280,12 <22>
POKE198,0:WAIT198,1:POKE53280,0:POKE198 <248>
,0 <177>
695 GOTO600 <22>
700 IF(PEEK(C)<>F1ANDPEEK(C)<>F2)OR(PEEK <248>
(C+40)<>F1ANDPEEK(C+40)<>F2)THENRETURN <177>
710 FORT=1T04:Y=Y+4:POKEV+1,Y:POKEB,254: <22>
GOSUB3:FORF=1T050:NEXT <248>
720 Y=Y+4:POKEV+1,Y:POKEB,255:GOSUB2:FOR <177>
F=1T050:NEXT:NEXT:POKEB,Z1:C=C+160:RETUR <22>
N <248>
750 IF(PEEK(C)<>F1ANDPEEK(C)<>F2)OR(PEEK <177>
(C+40)<>F1ANDPEEK(C+40)<>F2)THENRETURN <22>
760 FORT=1T04:Y=Y+4:POKEV+1,Y:POKEB,254: <248>
GOSUB3:FORF=1T050:NEXT <177>
770 Y=Y+4:POKEV+1,Y:POKEB,255:GOSUB2:FOR <22>
F=1T050:NEXT:NEXT:POKEB,Z1:C=C+160:RETUR <248>
N <177>
800 X(0)=1145:D(0)=20:X(1)=1171:D(1)=20: <177>
X(2)=1151:D(2)=4:X(3)=1165:D(3)=4 <124>
801 X(4)=1791:D(4)=4:X(5)=1805:D(5)=4:X <0>
(6)=1318:D(6)=12 <61>
802 Z1=178:Z2=177:F=5:RETURN <21>
810 X(0)=1147:D(0)=8:X(1)=1169:D(1)=8:X <17>
(2)=1478:D(2)=4:X(3)=1627:D(3)=8 <21>
811 X(4)=1649:D(4)=8:Z1=91:Z2=92:F=10:E1 <21>
=1:RETURN <21>
820 X(0)=1149:D(0)=4:X(1)=1327:D(1)=4:X <109>
(2)=1478:D(2)=4:X(3)=1629:D(3)=4 <56>
821 X(4)=1807:D(4)=4:Z1=93:Z2=93:F=7:E1= <231>
1:RETURN <171>
831 L(0)=1184:L(1)=1344:L(2)=1504:L(3)=1 <237>
664:L(4)=1824:L(5)=1211:L(6)=1371 <156>
832 Z1=94:Z2=99:E1=2:F=6:RETURN <135>
840 X(0)=1158:D(0)=8:X(1)=1467:D(1)=4:X <252>
(2)=1489:D(2)=4:X(3)=1638:D(3)=8 <206>
845 L(0)=1192:L(1)=1352:L(2)=1672:L(3)=1 <165>
832:L(4)=1203:L(5)=1363:L(6)=1683 <80>
846 L(7)=1843:F=14:Z1=43:Z2=43:E1=2:RETU <157>
RN
850 X(0)=1147:D(0)=4:X(1)=1329:D(1)=4:X <157>
(2)=1467:D(2)=4:X(3)=1649:D(3)=4 <157>
851 X(4)=1787:D(4)=4:Z1=40:Z2=41:F=8:L0= <157>
1:L1=1:E1=2:RETURN <157>
860 X(0)=1169:D(0)=4:X(1)=1318:D(1)=4:X <157>
(2)=1467:D(2)=4:X(3)=1638:D(3)=4 <157>
861 X(4)=1809:D(4)=4:Z1=33:Z2=33:F=2:E1= <157>
3:L0=1:L1=2:RETURN

```



```

870 X(0)=1155:D(0)=4:X(1)=1321:D(1)=4:X(
2)=1475:D(2)=4:X(3)=1641:D(3)=4 <252>
871 X(4)=1795:D(4)=4;Z1=63;Z2=63;F=11;E1
=3;L0=1;L1=3;RETURN <70>
880 X(0)=1146:D(0)=20;X(1)=1170:D(1)=20:
F=4;E1=4;L0=1;L1=3;Z1=61;Z2=61;RETURN <142>
890 X(0)=1158:D(0)=4;X(1)=1330:D(1)=4;X(
2)=1478:D(2)=4;X(3)=1626:D(3)=4 <97>
891 X(4)=1798:D(4)=4;F=3;E1=4;L0=1;L1=3:
Z1=44;Z2=44;RETURN <182>
900 RETURN <21>
1000 POKES+4,0;POKES+2,15;POKES+5,9;POKE
B+6,68;POKES,1;POKES+4,33;IFPEEK(B)=250T
HENGOTO1050 <121>
1005 IFX<68THENPOKES+4,0;RETURN <179>
1010 POKEB,248;FORT=1TO6;Y=Y-1;X=X-4;POK
EV+1,Y;POKEV,X;POKES+1,T*20 <120>
1020 FORF=1TO20:NEXT;NEXT <210>
1030 FORT=6TO18STEP-1;Y=Y+1;X=X-4;POKEV+1
,Y;POKEV,X;POKES+1,T*20 <6>
1040 FORF=1TO20:NEXT;NEXT;C=C-6;POKES+4,
0;B=2;RETURN <236>
1050 IFX>204THENPOKES+4,Z;RETURN <144>
1055 POKEB,249;FORT=1TO6;Y=Y-1;X=X+4;POK
EV+1,Y;POKEV,X;POKES+1,T*20 <148>
1060 FORF=1TO20:NEXT;NEXT <251>
1070 FORT=6TO18STEP-1;Y=Y+1;X=X+4;POKEV+1
,Y;POKEV,X;POKES+1,T*20 <26>
1080 FORF=1TO20:NEXT;NEXT;C=C+6;POKES+4,
0;B=2;RETURN <4>
5000 PRINT"RVSON GREEN CLEAR SPACE40";
;POKEV+33,0;POKEV+32,0 <185>
5010 PRINT" {SPACE8} {SPACE4} {SPACE
4} {SPACE15}"; <149>
5020 PRINT" {SPACE8} | {SPACE4} | {SPACE
4} | {SPACE15}"; <10>
5030 PRINT" {SPACE8} | {SPACE2} | {SPA
CE} | {SPACE} {SPACE} {SPACE9}"; <63>
5040 PRINT" {SPACE8} | {SPACE2} | {SPA
CE} | {SPACE} | {SPACE} | {SPACE9}"; <73>
5050 PRINT" {SPACE8} | {SPACE} | {SPA
CE} | {SPACE} | {SPACE9}"; <161>
5060 PRINT" {SPACE8} {SPACE} {SPA
CE} {SPACE} | {SPACE9}"; <253>
5070 PRINT" {SPACE27} | {SPACE9}"; <77>
5080 PRINT" {SPACE SHIFTSPACE SPACE25} {
SHIFTSPACE SPACE8}"; <54>
5090 PRINT"RVSOFF"; <83>
5100 PRINT"DOWN WHITE SPACE4}WRITTEN(SP
ACE)BY{SPACE}JOACHIM{SPACE}RESE{SPACE}IN
{SPACE}1984";V1=2;R=1;R1=1;P=1 <192>
5101 POKEV,40;POKEV+2,40;POKEV+16,3;POKE
V+4,50;POKEV+6,50 <218>
5102 POKEV+1,170;POKEV+3,220;POKEV+5,170
;POKEV+7,220;POKEV+28,31;POKEV+39,5 <84>
5103 POKEV+40,5;POKEV+41,5;POKEV+42,5;PO
KEV+37,7;POKEV+38,2 <1>
5104 FORT=0TO3;POKE2040+T,250;NEXT;POKEV
+21,31;Y=252;Y1=251;POKE198,0 <78>
5110 PRINT"HOME RVSOFF DOWN17 YELLOW",
" {SPACE4}LEVEL:";R1;" {LEFT SPACE2}" <14>
5120 PRINT," {SPACE2 DOWN3 LIG.RED RVSOFF
3} <49>
5130 PRINT," {SPACE2} | {SPACE}PRESS{SPACE}
FIRE{SPACE} |" <72>
5140 PRINT," {SPACE2} " :GOSU
B5200 <57>
5150 PRINT" {HOME DOWN17 YELLOW}", " {SPACE
4}LEVEL:";R1;" {LEFT SHIFTSPACE SPACE}" <82>
5160 PRINT,"DOWN3 LIG.RED SPACE2 RVSON"
<162>
5170 PRINT," {SPACE2 RVSON}| {SPACE}PRESS{
SPACE}FIRE{SPACE}|" <74>
5180 PRINT," {SPACE2 RVSON}

```

```

";B0SUB5200:G0T05110
5200 JO=56321;J=PEEK(JO):IF(JAND16)=0THE
N95
5210 JO=56320;J=PEEK(JO):IF(JAND16)=0THE
N95
5220 BETA$:IFA$="{F1}"THENR1=R1+1:IFR1>1
OTHENR1=1
5230 POKE2042,Y1:POKE2043,Y1:Y1=Y1+1:IFY
1=252THENY1=250
5235 POKE2040,Y:POKE2041,Y:Y=Y+1:IFY=254
THENY=252
5240 P=P+.2:IFP>=3THENP=1
5245 ONINT(P)B0SUB5300,5310
5250 RETURN
5300 PRINT"({HOME RVSOFF DOWN14})","({MGREY
SPACE2})HIGHSCORE:";HI,"({LEFT SPACE}":RE
TURN
5310 PRINT"({HOME RVSOFF DOWN14})","({DGREY
SPACE})LAST({SPACE})SCORE:";SC,"({LEFT SPAC
E5})":RETURN
6000 POKES+11,0:POKES+24,15:POKES+12,9:P
OKES+13,0:POKES+8,50:POKES+7,5:POKES+11,
17
6001 IFSC>HITHENHI=SC:PRINTH$;HI
6002 RETURN
6500 POKES+4,0:POKES+24,15:POKES+5,9:POK
ES+6,68:POKES,1:POKES+4,33
6510 IFM=1896THENPOKES+6,0:R1=R1+1:G0T01
00
6520 POKEM,M(M1):M1=M1-1:IFM1<0THENM=M-1
:M1=B
6530 FORT=50T0200STEP8:POKES+1,T:NEXT:SC
=SC+5:PRINTS$;SC:IFSC>HITHENHI=SC:PRINT
H$;HI
6540 B0T06510
7000 IFPEEK(B)>248ANDPEEK(B)<252THENPOKE
B,249:G0T07010
7005 POKEB,248
7010 POKES+4,0:POKES+24,15:POKES+5,9:POK
ES+6,255:POKES,1:POKES+4,33
7020 E=200:FORT=YT0230:E=E-1:POKEV+1,T:P
OKES+1,E:NEXT:POKES+4,0
7030 FORT=0T024:POKES+T,0:NEXT:POKES+24,
15:POKES+5,9:POKES+6,255:POKES,1:POKES+4
,33
7040 FORT=99T020STEP-2:POKES+1,T:NEXT:PO
KES+6,0:VI=VI-1:IFVI=-1THEN7050
7045 G0T0100
7050 FORT=1T01000:NEXT
7051 PRINT"({HOME RVSOFF CYAN DOWN8 RIGHT
Z RVSON}:",{RIGHT16}"
7052 PRINT"({RIGHT7 RVSON}|{SPACE}|{
SPACE}|{SPACE}|{RIGHT16}"
7055 PRINT"({RIGHT7 RVSON}|{SPACE}|{
SPACE}|{SPACE}|{SPACE}|{RIGHT16}"
7060 PRINT"({RIGHT7 RVSON}|{SPACE}|{SP
ACE}|{SPACE}|{SPACE2}|{SPACE}|{RIGHT16
}"
7070 PRINT"({RIGHT7 RVSON}|{SPACE}|{SPA
CE}|{SPACE}|{SPACE}|{SPACE}|{RIGHT16
}"
7075 PRINT"({RIGHT7 RVSON}|{SPACE}|{SPACE
}|{SPACE}|{SPACE}|{SPACE}|{RIGHT16}"
7080 PRINT"({RIGHT7 RVSON}|{SPACE}|{SPA
CE}|{SPACE}|{SPACE}|{SPACE2}|{RIGHT16
}"
7090 PRINT"({RIGHT7 RVSON}|{
"
7100 FORT=0T024:POKES+T,0:NEXT
7110 POKES+24,15:POKES+5,9:POKES+6,255:P
OKES,1:POKES+4,17
7120 FORT=40T020STEP-1:POKES+1,T:FOR=1T
0100:NEXT:NEXT:POKES+6,0
7130 FORT=1T0500:J=PEEK(JO):IF(JAND16)<>

```

```

OTHENNEXT                                <81>
7140 B0T05000                            <50>
9000 POKE2,255:POKE56334,PEEK(56334)AND2
54:POKE1,PEEK(1)AND251:E=12288         <233>
9010 FORT=1T02000:POKEE+T,PEEK(V+T):NEXT <12>
9020 POKE1,PEEK(1)OR4:POKE56334,PEEK(563
34)OR1                                    <68>
9030 READA1:IFA1=-1THENRETURN            <211>
9040 FORI=0T07:READP:POKEE+A1*8+I,P:NEXT
:60T09030                                <180>
10000 DATA3,192,0,15,240,0,13,112,0,5,11
2,0,5,112,0,2,128,0,10,160,0          <209>
10010 DATA43,168,0,255,168,0,255,168,0,4
2,168,0,42,168,0,42,168,0,63,252,0    <122>
10020 DATA207,240,0,207,240,0,255,240,0,
255,240,0,0,0,0,0,0,0,0,0,0           <22>
10030 DATA3,192,0,15,240,0,13,112,0,13,8
0,0,13,80,0,2,128,0,10,160,0,42,232,0 <2>
10040 DATA42,255,0,42,255,0,42,168,0,42,
168,0,42,168,0,63,252,0,15,243,0       <155>
10050 DATA15,243,0,15,255,0,15,255,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0              <198>
10060 DATA3,192,0,15,240,0,13,112,0,13,8
0,0,13,80,0,2,128,0,10,160,0,42,232,0 <32>
10070 DATA42,255,0,42,255,0,42,168,0,42,
168,0,42,168,0,63,252,0,15,240,0,3,192,0 <22>
10080 DATA3,192,0,3,192,0,3,192,0,3,240,
0,3,240,0                                <111>
10090 DATA3,192,0,15,240,0,13,112,0,13,8
0,0,13,80,0,2,128,0,10,160,0,42,232,0 <62>
10100 DATA42,232,0,42,232,0,42,232,0,42,
232,0,42,232,0,63,252,0,15,240,0      <132>
10110 DATA15,240,0,12,48,0,12,48,0,12,48
,0,15,60,0,15,60,0                      <150>
10120 DATA3,192,0,15,240,0,13,112,0,5,11
2,0,5,112,0,2,128,0,10,160,0,43,168,0 <213>
10130 DATA255,168,0,255,168,0,42,168,0,4
2,168,0,42,168,0,63,252,0,15,240,0    <124>
10140 DATA3,192,0,3,192,0,3,192,0,3,192,
0,15,192,0,15,192,0                    <24>
10150 DATA3,192,0,15,240,0,13,112,0,5,11
2,0,5,112,0,2,128,0,10,160,0,43,168,0 <243>
10160 DATA43,168,0,43,168,0,43,168,0,43,
168,0,43,168,0,63,252,0,15,240,0      <12>
10170 DATA15,240,0,12,48,0,12,48,0,12,48
,0,60,240,0,60,240,0                    <184>
10180 DATA3,192,0,15,240,0,15,240,0,15,2
40,0,15,243,0,3,195,0,10,163,0,42,171,0 <147>
10190 DATA234,171,0,234,171,0,234,168,0,
234,168,0,234,168,0,255,252,0,15,240,0 <174>
10200 DATA3,192,0,3,192,0,3,240,0,3,240,

```

0,15,0,0,15,0,0	<224>
10210 DATA3,192,0,15,240,0,15,240,0,15,240,0,207,240,0,195,192,0,202,160,0	<207>
10220 DATA234,168,0,234,171,0,234,171,0,42,171,0,42,171,0,42,171,0,63,255,0	<251>
10230 DATA15,240,0,3,192,0,3,192,0,15,192,0,15,192,0,0,240,0,0,240,0	<41>
11000 DATA0,126,60,102,223,231,251,231,126	<151>
11010 DATA158,255,255,0,255,255,0,255,255,255	<243>
11020 DATA155,255,255,7,30,120,224,255,255	<250>
11030 DATA156,255,255,224,120,30,7,255,255	<0>
11040 DATA157,255,255,189,153,153,189,255,255	<201>
11050 DATA171,255,255,255,195,195,255,255,255	<187>
11060 DATA168,255,255,195,0,60,255,255,255	<22>
11070 DATA169,255,255,255,60,0,195,255,255	<47>
11080 DATA94,60,255,153,219,126,36,60,24	<157>
11090 DATA163,255,255,195,199,206,220,248,241	<118>
11100 DATA164,255,255,195,227,115,59,31,143	<108>
11110 DATA165,241,248,220,206,199,195,255,255	<166>
11120 DATA166,143,31,59,115,227,195,255,255	<104>
11130 DATA42,60,126,255,153,255,195,102,60	<82>
11140 DATA161,255,255,7,7,240,240,255,255	<104>
11150 DATA191,255,153,153,255,255,153,153,255	<165>
11160 DATA189,255,255,240,96,6,15,255,255	<57>
11170 DATA172,255,24,24,255,255,24,24,255	<127>
11180 DATA218,231,231,231,231,231,255,255,255	<243>
11190 DATA240,0,0,0,7,15,31,31,31	<140>
11200 DATA238,0,0,0,224,240,248,248,248	<197>
11210 DATA237,31,31,31,15,7,0,0,0	<37>
11220 DATA253,248,248,248,240,224,0,0,0	<99>
11230 DATA211,255,255,255,15,15,255,255,255,-1	<198>
<p>ENDE DES LISTINGS</p>	

Werkstatt:

Music-Master

Mit dem Programm 'MUSIC-MASTER' können Sie wie in professionellen Spielen dreistimmige unabhängige Hintergrundmusik programmieren und in eigenen Basicprogrammen verwenden.

Um ein Musikstück spielen zu können, müssen zuerst die Noten eingegeben werden.

Im Listing stehen dort bereits die Noten für ein kleines zweistimmiges Demostück. Die Noten stehen in Datazeilen und wer-

den in folgendem Format eingegeben:
'D5,4' bedeutet z. B. **Note D, Oktave 5**. Die folgende **4** gibt die **Notenlänge** an. Pausen werden mit '**P.Länge**' eingegeben.
Eine Möglichkeit der Staffelung der Notenlängen wäre z. B.:

16 – für eine ganze Note
8 – für eine halbe Note
4 – für eine viertel Note
2 – für eine achtel Note
1 – für eine sechzehntel Note



Diese Werte sind jedoch nicht verbindlich; es können genauso gut auch andere Werte verwendet werden.

Abgeschlossen wird die Noteneingabe der einzelnen Stimmen mit 'ENDE'.

Um die verschiedenen SID-Register, wie z.B. die Wellenform zu setzen, muß man die entsprechenden Werte in den Zeilen 86-95 einsetzen.

Für die Wellenform muß je nach Wunsch 17 für Dreieck, 33 für Sägezahn, 65 für

Rechteck, 129 für Pauschen oder 0 für keine Wellenform geschrieben werden. Bei Rechteck muß zusätzlich das Tastenverhältnis im Format Lowbyte-Highbyte angegeben werden.

Die Hüllkurve setzt sich aus 2 Werten zusammen:

Wert 1: 16 * Attackzeit + Decayzeit

Wert 2: 16 * Sustainpegel + Releasezeit

Für Attack, Decay, Sustain und Release können Werte von 0-15 eingesetzt werden.

Die Startadresse in Zeile 86 gibt den Beginn des MC-Programms an und kann nach den jeweiligen Bedürfnissen abgeändert werden.

Die Angabe bei Tempo in Zeile 89 gibt die Spielgeschwindigkeit an. Der Wert kann zwischen 1 (schnell) und 255 (langsam) liegen.

In Zeile 88 gibt man an, welche Stimmen eingeschaltet werden. 1=an, 0=aus. Man kann dann z.B. Stimme 1 und 2 für Musik benutzen während Stimme 3 für Geräusche frei bleibt. Wenn man alle Noten- und Parameterdaten eingegeben hat, kann man das Programm mit 'RUN' starten. Nun werden die Noten- und Parameterwerte eingelesen und in den Speicher geschrieben, was einige Minuten dauern kann.

Am Ende werden die Start- und Endadresse des Notenspeichers angezeigt, die man sich notieren sollte.

Die Musik kann jetzt probeweise mit 'SYS Startadresse' gestartet werden.

Um die Musik für eigene Programme verwendbar zu gestalten und um die Einlesezeit der Daten zu verkürzen, muß man den Datagenerator einladen. Vorher ist es jedoch notwendig den IRQ-Vektor mit 'STOP/RESTORE' zurückzusetzen. Auf die Fragen nach Start- und Endadresse gibt man die notierten Werte ein. Der Datagenerator generiert nun die notwendigen Datenzeilen und eine Einleseroutine mit zeilenweiser Prüfsumme. Danach löscht er sich selbst.

Die wichtigsten Register:

Music-Master belegt die Zeropageadressen 176-181, sowie die Register 828-858 im Kassettenspeicher.

Adresse

835 Tempo

848 Wellenform Stimme 1

849 Wellenform Stimme 2

850 Wellenform Stimme 3

851 Stimme 1 ein/aus:

1=ein, 0=aus

852 Stimme 2 ein/aus

853 Stimme 3 ein/aus

Während die Musik spielt kann man durch Ändern dieser Register z.B. das Tempo verändern oder mit 'POKE 848,0' Stimme 1 stummschalten oder einzelne Stimmen ganz abschalten und so für andere Soundeffekte zu benutzen.

```

1 REM *****
2 REM **
3 REM ** MUSIC-MASTER **
4 REM **
5 REM ** (C) 1985 **
6 REM **
7 REM ** K. SELIGER **
8 REM **
9 REM *****
10 :
11 :
12 REM *** KLANGPARAMETER EINLESEN ***
13 DIM PA(20)
14 FORA=1TO20:READPA(A):NEXTA
15 :
16 :
17 REM *** NOTENPARAMETER EINLESEN ***
18 DIM NO$(96),WE(96)
19 FORA=1TO96:READNO$(A):NEXT
20 FORA=1TO96:READWE(A):NEXT
21 :
22 :
23 REM *** MC-ROUTINE EINLESEN ***
24 ZE=119:BA=PA(1)
25 FORA=1TO49
26 FORB=0TO8:READDA:POKEBA+B,DA:SU=SU+DA
27 :NEXTB
28 READSD:IFSD<>SUTHENPRINT"DATAFEHLER(S
29 :PACE)IN(SPACE)ZEILE"ZE:END
30 ZE=ZE+1:BA=BA+9:SU=0
31 :
32 :
33 REM *** NOTENWERTE EINLESEN ***
34 FORB=1TO3
35 READNO$:IFNO$="ENDE"THENNO(B+1)=463+P
36 :A(1)+ZE:AN(B)=AN:AN=0:NEXTB:GOTO52
37 WE=1
38 FORA=1TO96
39 IFNO$(A)=NO$THENWE=WE(A)
40 :NEXTA
41 IFWE=1THENPRINT"SYNTAX(SPACE)FEHLER(
42 :SPACE)IN(SPACE)NOTE"AN:END
43 :HB=INT(WE/256)
44 :LB=WE-HB*256
45 :POKE463+PA(1)+ZE,HB:ZE=ZE+1
46 :READTT
47 :POKE463+PA(1)+ZE,TT:ZE=ZE+1
48 :GOTO35
49 :
50 :
51 REM *** KLANGPARAMETER SETZEN ***
52 NO(1)=463+PA(1):B=0:EA=462+PA(1)+ZE
53 :FORA=1TO3
54 :HB=INT(NO(A)/256):LB=NO(A)-256*HB
55 :POKEPA(1)+434+B,LB:POKEPA(1)+435+B,HB
56 :B=B+2
57 :NEXTA
58 :
59 :
60 :FORA=2TO7
61 :POKEPA(1)+438+A,PA(A)
62 :NEXT
63 :
64 :AN(1)=AN(1)*3+PA(1)+463
65 :HB=INT(AN(1)/256):LB=AN(1)-256*HB
66 :POKEPA(1)+446,LB:POKEPA(1)+447,HB
67 :
68 :HB=INT((PA(1)+191)/256):LB=(PA(1)+191
69 :)-256*HB
70 :POKEPA(1)+448,LB:POKEPA(1)+449,HB
71 :
72 :POKEPA(1)+450,PA(8)
73 :
74 :FORA=1TO12
75 :POKEPA(1)+450+A,PA(A+8)
76 :NEXTA
77 :HB=INT((PA(1)+434)/256):LB=(PA(1)+434
78 :)-256*HB
79 :POKEPA(1)+1,LB:POKEPA(1)+5,HB
80 :
81 :PRINTCHR$(147);"STARTADRESSE:"PA(1)
82 :PRINT"ENDADRESSE(SPACE2):"EA
83 :
84 REM *** PARAMETERDATAS ***
85 :
86 DATA 49152 : REM STARTADRESSE
87 DATA 33,33,0 : REM WELLENFORMEN FUER
88 :STIMMEN 1-3
89 DATA 1,1,0 : REM STIMMEN EINSCHALTEN:
90 :1=EIN 0=AUS
91 DATA 13 : REM TEMPO (1-255)
92 DATA 0,0 : REM TASTENVERHAELTNI STIM
93 :ME 1
94 DATA 10,10 : REM HUELLKURVE STIMME 1
95 :16*A+D, 16*S+R
96 DATA 0,0 : REM TASTENVERHAELTNIS STIM
97 :ME 2
98 DATA 10,10 : REM HUELLKURVE STIMME 2
99 :16*A+D, 16*S+R
100 DATA 0,0 : REM TASTENVERHAELTNIS STIM
101 :ME 3
102 DATA 0,0 : REM HUELLKURVE STIMME 3 :
103 :16*A+D, 16*S+R
104 :
105 :
106 :
107 :
108 :
109 :
110 :
111 :
112 :
113 :
114 :
115 :
116 :
117 :
118 :
119 :
120 :
121 :
122 :
123 :
124 :
125 :
126 :
127 :
128 :
129 :
130 :
131 :
132 :
133 :
134 :
135 :
136 :
137 :
138 :
139 :
140 :
141 :
142 :
143 :
144 :
145 :
146 :
147 :
148 :
149 :
150 :
151 :
152 :
153 :
154 :
155 :
156 :
157 :
158 :
159 :
160 :
161 :
162 :
163 :
164 :
165 :
166 :
167 :
168 :
169 :
170 :
171 :
172 :
173 :
174 :
175 :
176 :
177 :
178 :
179 :
180 :
181 :
182 :
183 :
184 :
185 :
186 :
187 :
188 :
189 :
190 :
191 :
192 :
193 :
194 :
195 :
196 :
197 :
198 :
199 :
200 :
201 :
202 :
203 :
204 :
205 :
206 :
207 :
208 :
209 :
210 :
211 :
212 :
213 :
214 :
215 :
216 :
217 :
218 :
219 :
220 :
221 :
222 :
223 :
224 :
225 :
226 :
227 :
228 :
229 :
230 :
231 :
232 :
233 :
234 :
235 :
236 :
237 :
238 :
239 :
240 :
241 :
242 :
243 :
244 :
245 :
246 :
247 :
248 :
249 :
250 :
251 :
252 :
253 :
254 :
255 :
256 :
257 :
258 :
259 :
260 :
261 :
262 :
263 :
264 :
265 :
266 :
267 :
268 :
269 :
270 :
271 :
272 :
273 :
274 :
275 :
276 :
277 :
278 :
279 :
280 :
281 :
282 :
283 :
284 :
285 :
286 :
287 :
288 :
289 :
290 :
291 :
292 :
293 :
294 :
295 :
296 :
297 :
298 :
299 :
300 :
301 :
302 :
303 :
304 :
305 :
306 :
307 :
308 :
309 :
310 :
311 :
312 :
313 :
314 :
315 :
316 :
317 :
318 :
319 :
320 :
321 :
322 :
323 :
324 :
325 :
326 :
327 :
328 :
329 :
330 :
331 :
332 :
333 :
334 :
335 :
336 :
337 :
338 :
339 :
340 :
341 :
342 :
343 :
344 :
345 :
346 :
347 :
348 :
349 :
350 :
351 :
352 :
353 :
354 :
355 :
356 :
357 :
358 :
359 :
360 :
361 :
362 :
363 :
364 :
365 :
366 :
367 :
368 :
369 :
370 :
371 :
372 :
373 :
374 :
375 :
376 :
377 :
378 :
379 :
380 :
381 :
382 :
383 :
384 :
385 :
386 :
387 :
388 :
389 :
390 :
391 :
392 :
393 :
394 :
395 :
396 :
397 :
398 :
399 :
400 :
401 :
402 :
403 :
404 :
405 :
406 :
407 :
408 :
409 :
410 :
411 :
412 :
413 :
414 :
415 :
416 :
417 :
418 :
419 :
420 :
421 :
422 :
423 :
424 :
425 :
426 :
427 :
428 :
429 :
430 :
431 :
432 :
433 :
434 :
435 :
436 :
437 :
438 :
439 :
440 :
441 :
442 :
443 :
444 :
445 :
446 :
447 :
448 :
449 :
450 :
451 :
452 :
453 :
454 :
455 :
456 :
457 :
458 :
459 :
460 :
461 :
462 :
463 :
464 :
465 :
466 :
467 :
468 :
469 :
470 :
471 :
472 :
473 :
474 :
475 :
476 :
477 :
478 :
479 :
480 :
481 :
482 :
483 :
484 :
485 :
486 :
487 :
488 :
489 :
490 :
491 :
492 :
493 :
494 :
495 :
496 :
497 :
498 :
499 :
500 :
501 :
502 :
503 :
504 :
505 :
506 :
507 :
508 :
509 :
510 :
511 :
512 :
513 :
514 :
515 :
516 :
517 :
518 :
519 :
520 :
521 :
522 :
523 :
524 :
525 :
526 :
527 :
528 :
529 :
530 :
531 :
532 :
533 :
534 :
535 :
536 :
537 :
538 :
539 :
540 :
541 :
542 :
543 :
544 :
545 :
546 :
547 :
548 :
549 :
550 :
551 :
552 :
553 :
554 :
555 :
556 :
557 :
558 :
559 :
560 :
561 :
562 :
563 :
564 :
565 :
566 :
567 :
568 :
569 :
570 :
571 :
572 :
573 :
574 :
575 :
576 :
577 :
578 :
579 :
580 :
581 :
582 :
583 :
584 :
585 :
586 :
587 :
588 :
589 :
590 :
591 :
592 :
593 :
594 :
595 :
596 :
597 :
598 :
599 :
600 :
601 :
602 :
603 :
604 :
605 :
606 :
607 :
608 :
609 :
610 :
611 :
612 :
613 :
614 :
615 :
616 :
617 :
618 :
619 :
620 :
621 :
622 :
623 :
624 :
625 :
626 :
627 :
628 :
629 :
630 :
631 :
632 :
633 :
634 :
635 :
636 :
637 :
638 :
639 :
640 :
641 :
642 :
643 :
644 :
645 :
646 :
647 :
648 :
649 :
650 :
651 :
652 :
653 :
654 :
655 :
656 :
657 :
658 :
659 :
660 :
661 :
662 :
663 :
664 :
665 :
666 :
667 :
668 :
669 :
670 :
671 :
672 :
673 :
674 :
675 :
676 :
677 :
678 :
679 :
680 :
681 :
682 :
683 :
684 :
685 :
686 :
687 :
688 :
689 :
690 :
691 :
692 :
693 :
694 :
695 :
696 :
697 :
698 :
699 :
700 :
701 :
702 :
703 :
704 :
705 :
706 :
707 :
708 :
709 :
710 :
711 :
712 :
713 :
714 :
715 :
716 :
717 :
718 :
719 :
720 :
721 :
722 :
723 :
724 :
725 :
726 :
727 :
728 :
729 :
730 :
731 :
732 :
733 :
734 :
735 :
736 :
737 :
738 :
739 :
740 :
741 :
742 :
743 :
744 :
745 :
746 :
747 :
748 :
749 :
750 :
751 :
752 :
753 :
754 :
755 :
756 :
757 :
758 :
759 :
760 :
761 :
762 :
763 :
764 :
765 :
766 :
767 :
768 :
769 :
770 :
771 :
772 :
773 :
774 :
775 :
776 :
777 :
778 :
779 :
780 :
781 :
782 :
783 :
784 :
785 :
786 :
787 :
788 :
789 :
790 :
791 :
792 :
793 :
794 :
795 :
796 :
797 :
798 :
799 :
800 :
801 :
802 :
803 :
804 :
805 :
806 :
807 :
808 :
809 :
810 :
811 :
812 :
813 :
814 :
815 :
816 :
817 :
818 :
819 :
820 :
821 :
822 :
823 :
824 :
825 :
826 :
827 :
828 :
829 :
830 :
831 :
832 :
833 :
834 :
835 :
836 :
837 :
838 :
839 :
840 :
841 :
842 :
843 :
844 :
845 :
846 :
847 :
848 :
849 :
850 :
851 :
852 :
853 :
854 :
855 :
856 :
857 :
858 :
859 :
860 :
861 :
862 :
863 :
864 :
865 :
866 :
867 :
868 :
869 :
870 :
871 :
872 :
873 :
874 :
875 :
876 :
877 :
878 :
879 :
880 :
881 :
882 :
883 :
884 :
885 :
886 :
887 :
888 :
889 :
890 :
891 :
892 :
893 :
894 :
895 :
896 :
897 :
898 :
899 :
900 :
901 :
902 :
903 :
904 :
905 :
906 :
907 :
908 :
909 :
910 :
911 :
912 :
913 :
914 :
915 :
916 :
917 :
918 :
919 :
920 :
921 :
922 :
923 :
924 :
925 :
926 :
927 :
928 :
929 :
930 :
931 :
932 :
933 :
934 :
935 :
936 :
937 :
938 :
939 :
940 :
941 :
942 :
943 :
944 :
945 :
946 :
947 :
948 :
949 :
950 :
951 :
952 :
953 :
954 :
955 :
956 :
957 :
958 :
959 :
960 :
961 :
962 :
963 :
964 :
965 :
966 :
967 :
968 :
969 :
970 :
971 :
972 :
973 :
974 :
975 :
976 :
977 :
978 :
979 :
980 :
981 :
982 :
983 :
984 :
985 :
986 :
987 :
988 :
989 :
990 :
991 :
992 :
993 :
994 :
995 :
996 :
997 :
998 :
999 :
1000 :

```

```

45 POKE463+PA(1)+ZE,HB:ZE=ZE+1
46 READTT
47 POKE463+PA(1)+ZE,TT:ZE=ZE+1
48 GOTO35
49 :
50 :
51 REM *** KLANGPARAMETER SETZEN ***
52 NO(1)=463+PA(1):B=0:EA=462+PA(1)+ZE
53 :FORA=1TO3
54 :HB=INT(NO(A)/256):LB=NO(A)-256*HB
55 :POKEPA(1)+434+B,LB:POKEPA(1)+435+B,HB
56 :B=B+2
57 :NEXTA
58 :
59 :
60 :FORA=2TO7
61 :POKEPA(1)+438+A,PA(A)
62 :NEXT
63 :
64 :AN(1)=AN(1)*3+PA(1)+463
65 :HB=INT(AN(1)/256):LB=AN(1)-256*HB
66 :POKEPA(1)+446,LB:POKEPA(1)+447,HB
67 :
68 :HB=INT((PA(1)+191)/256):LB=(PA(1)+191
69 :)-256*HB
70 :POKEPA(1)+448,LB:POKEPA(1)+449,HB
71 :
72 :POKEPA(1)+450,PA(8)
73 :
74 :FORA=1TO12
75 :POKEPA(1)+450+A,PA(A+8)
76 :NEXTA
77 :HB=INT((PA(1)+434)/256):LB=(PA(1)+434
78 :)-256*HB
79 :POKEPA(1)+1,LB:POKEPA(1)+5,HB
80 :
81 :PRINTCHR$(147);"STARTADRESSE:"PA(1)
82 :PRINT"ENDADRESSE(SPACE2):"EA
83 :
84 REM *** PARAMETERDATAS ***
85 :
86 DATA 49152 : REM STARTADRESSE
87 DATA 33,33,0 : REM WELLENFORMEN FUER
88 :STIMMEN 1-3
89 DATA 1,1,0 : REM STIMMEN EINSCHALTEN:
90 :1=EIN 0=AUS
91 DATA 13 : REM TEMPO (1-255)
92 DATA 0,0 : REM TASTENVERHAELTNI STIM
93 :ME 1
94 DATA 10,10 : REM HUELLKURVE STIMME 1
95 :16*A+D, 16*S+R
96 DATA 0,0 : REM TASTENVERHAELTNIS STIM
97 :ME 2
98 DATA 10,10 : REM HUELLKURVE STIMME 2
99 :16*A+D, 16*S+R
100 DATA 0,0 : REM TASTENVERHAELTNIS STIM
101 :ME 3
102 DATA 0,0 : REM HUELLKURVE STIMME 3 :
103 :16*A+D, 16*S+R
104 :
105 :
106 :
107 :
108 :
109 :
110 :
111 :
112 :
113 :
114 :
115 :
116 :
117 :
118 :
119 :
120 :
121 :
122 :
123 :
124 :
125 :
126 :
127 :
128 :
129 :
130 :
131 :
132 :
133 :
134 :
135 :
136 :
137 :
138 :
139 :
140 :
141 :
142 :
143 :
144 :
145 :
146 :
147 :
148 :
149 :
150 :
151 :
152 :
153 :
154 :
155 :
156 :
157 :
158 :
159 :
160 :
161 :
162 :
163 :
164 :
165 :
166 :
167 :
168 :
169 :
170 :
171 :
172 :
173 :
174 :
175 :
176 :
177 :
178 :
179 :
180 :
181 :
182 :
183 :
184 :
185 :
186 :
187 :
188 :
189 :
190 :
191 :
192 :
193 :
194 :
195 :
196 :
197 :
198 :
199 :
200 :
201 :
202 :
203 :
204 :
205 :
206 :
207 :
208 :
209 :
210 :
211 :
212 :
213 :
214 :
215 :
216 :
217 :
218 :
219 :
220 :
221 :
222 :
223 :
224 :
225 :
226 :
227 :
228 :
229 :
230 :
231 :
232 :
233 :
234 :
235 :
236 :
237 :
238 :
239 :
240 :
241 :
242 :
243 :
244 :
245 :
246 :
247 :
248 :
249 :
250 :
251 :
252 :
253 :
254 :
255 :
256 :
257 :
258 :
259 :
260 :
261 :
262 :
263 :
264 :
265 :
266 :
267 :
268 :
269 :
270 :
271 :
272 :
273 :
274 :
275 :
276 :
277 :
278 :
279 :
280 :
281 :
282 :
283 :
284 :
285 :
286 :
287 :
288 :
289 :
290 :
291 :
292 :
293 :
294 :
295 :
296 :
297 :
298 :
299 :
300 :
301 :
302 :
303 :
304 :
305 :
306 :
307 :
308 :
309 :
310 :
311 :
312 :
313 :
314 :
315 :
316 :
317 :
318 :
319 :
320 :
321 :
322 :
323 :
324 :
325 :
326 :
327 :
328 :
329 :
330 :
331 :
332 :
333 :
334 :
335 :
336 :
337 :
338 :
339 :
340 :
341 :
342 :
343 :
344 :
345 :
346 :
347 :
348 :
349 :
350 :
351 :
352 :
353 :
354 :
355 :
356 :
357 :
358 :
359 :
360 :
361 :
362 :
363 :
364 :
365 :
366 :
367 :
368 :
369 :
370 :
371 :
372 :
373 :
374 :
375 :
376 :
377 :
378 :
379 :
380 :
381 :
382 :
383 :
384 :
385 :
386 :
387 :
388 :
389 :
390 :
391 :
392 :
393 :
394 :
395 :
396 :
397 :
398 :
399 :
400 :
401 :
402 :
403 :
404 :
405 :
406 :
407 :
408 :
409 :
410 :
411 :
412 :
413 :
414 :
415 :
416 :
417 :
418 :
419 :
420 :
421 :
422 :
423 :
424 :
425 :
426 :
427 :
428 :
429 :
430 :
431 :
432 :
433 :
434 :
435 :
436 :
437 :
438 :
439 :
440 :
441 :
442 :
443 :
444 :
445 :
446 :
447 :
448 :
449 :
450 :
451 :
452 :
453 :
454 :
455 :
456 :
457 :
458 :
459 :
460 :
461 :
462 :
463 :
464 :
465 :
466 :
467 :
468 :
469 :
470 :
471 :
472 :
473 :
474 :
475 :
476 :
477 :
478 :
479 :
480 :
481 :
482 :
483 :
484 :
485 :
486 :
487 :
488 :
489 :
490 :
491 :
492 :
493 :
494 :
495 :
496 :
497 :
498 :
499 :
500 :
501 :
502 :
503 :
504 :
505 :
506 :
507 :
508 :
509 :
510 :
511 :
512 :
513 :
514 :
515 :
516 :
517 :
518 :
519 :
520 :
521 :
522 :
523 :
524 :
525 :
526 :
527 :
528 :
529 :
530 :
531 :
532 :
533 :
534 :
535 :
536 :
537 :
538 :
539 :
540 :
541 :
542 :
543 :
544 :
545 :
546 :
547 :
548 :
549 :
550 :
551 :
552 :
553 :
554 :
555 :
556 :
557 :
558 :
559 :
560 :
561 :
562 :
563 :
564 :
565 :
566 :
567 :
568 :
569 :
570 :
571 :
572 :
573 :
574 :
575 :
576 :
577 :
578 :
579 :
580 :
581 :
582 :
583 :
584 :
585 :
586 :
587 :
588 :
589 :
590 :
591 :
592 :
593 :
594 :
595 :
596 :
597 :
598 :
599 :
600 :
601 :
602 :
603 :
604 :
605 :
606 :
607 :
608 :
609 :
610 :
611 :
612 :
613 :
614 :
615 :
616 :
617 :
618 :
619 :
620 :
621 :
622 :
623 :
624 :
625 :
626 :
627 :
628 :
629 :
630 :
631 :
632 :
633 :
634 :
635 :
636 :
637 :
638 :
639 :
640 :
641 :
642 :
643 :
644 :
645 :
646 :
647 :
648 :
649 :
650 :
651 :
652 :
653 :
654 :
655 :
656 :
657 :
658 :
659 :
660 :
661 :
662 :
663 :
664 :
665 :
666 :
667 :
668 :
669 :
670 :
671 :
672 :
673 :
674 :
675 :
676 :
677 :
678 :
679 :
680 :
681 :
682 :
683 :
684 :
685 :
686 :
687 :
688 :
689 :
690 :
691 :
692 :
693 :
694 :
695 :
696 :
697 :
698 :
699 :
700 :
701 :
702 :
703 :
704 :
705 :
706 :
707 :
708 :
709 :
710 :
711 :
712 :
713 :
714 :
715 :
716 :
717 :
718 :
719 :
720 :
721 :
722 :
723 :
724 :
725 :
726 :
727 :
728 :
729 :
730 :
731 :
732 :
733 :
734 :
735 :
736 :
737 :
738 :
739 :
740 :
741 :
742 :
743 :
744 :
745 :
746 :
747 :
748 :
749 :
750 :
751 :
752 :
753 :
754 :
755 :
756 :
757 :
758 :
759 :
760 :
761 :
762 :
763 :
764 :
765 :
766 :
767 :
768 :
769 :
770 :
771 :
772 :
773 :
774 :
775 :
776 :
777 :
778 :
779 :
780 :
781 :
782 :
783 :
784 :
785 :
786 :
787 :
788 :
789 :
790 :
791 :
792 :
793 :
794 :
795 :
796 :
797 :
798 :
799 :
800 :
801 :
802 :
803 :
804 :
805 :
806 :
807 :
808 :
809 :
810 :
811 :
812 :
813 :
814 :
815 :
816 :
817 :
818 :
819 :
820 :
821 :
822 :
823 :
824 :
825 :
826 :
827 :
828 :
829 :
830 :
831 :
832 :
833 :
834 :
835 :
836 :
837 :
838 :
839 :
840 :
841 :
842 :
843 :
844 :
845 :
846 :
847 :
848 :
849 :
850 :
851 :
852 :
853 :
854 :
855 :
856 :
857 :
858 :
859 :
860 :
861 :
862 :
863 :
864 :
865 :
866 :
867 :
868 :
869 :
870 :
871 :
872 :
873 :
874 :
875 :
87
```



```

145 DATA 3,208,37,160,0,177,176,200,141,1102 <2>
146 DATA 68,3,177,176,200,141,69,3,177,1014 <188>
147 DATA 176,141,60,3,165,176,24,105,3,853 <24>
148 DATA 133,176,165,177,105,0,133,177,169,1235 <219>
149 DATA 0,141,4,212,174,84,3,240,46,904 <23>
150 DATA 174,61,3,202,142,61,3,208,37,891 <215>
151 DATA 160,0,177,178,200,141,70,3,177,1106 <36>
152 DATA 178,200,141,71,3,177,178,141,61,1150 <104>
153 DATA 3,165,178,24,105,3,133,178,165,954 <108>
154 DATA 179,105,0,133,179,169,0,141,11,917 <211>
155 DATA 212,173,85,3,240,46,174,62,3,998 <67>
156 DATA 202,142,62,3,208,37,160,0,177,991 <221>
157 DATA 180,200,141,72,3,177,180,200,141,1294 <150>
158 DATA 73,3,177,180,141,62,3,165,180,984 <72>
159 DATA 24,105,3,133,180,165,181,105,0,896 <86>
160 DATA 133,181,169,0,141,18,212,234,173,1261 <173>
161 DATA 80,3,141,4,212,173,68,3,141,825 <31>
162 DATA 0,212,173,69,3,141,1,212,173,84 <3>
163 DATA 81,3,141,11,212,173,70,3,141,835 <219>
164 DATA 7,212,173,71,3,141,8,212,173,1000 <46>
165 DATA 82,3,141,18,212,173,72,3,141,845 <130>
166 DATA 14,212,173,73,3,141,15,212,76,919 <201>
167 DATA 49,234,0,0,0,0,0,0,0,283 <142>
168 : <226>
169 REM *** DEMO-STUECK *** <144>
170 REM ***** <131>
171 : <229>
172 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,H4,4,H4,4,C#5,1,C#5,1 <65>
173 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2,E5,1,A4,3 <201>
174 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,H4,4,H4,4,C#5,1,C#5,1 <67>
175 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2,E5,1,A4,3 <203>
176 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1,F#5,2,E5,5 <43>
177 DATA F#5,3,E5,1,D5,2,D5,1,H4,1,E5,1,F#5,2,E5,3,D5,1,E5,1 <96>
178 DATA F#5,2,A5,2,G5,2,F#5,1,D5,1,E5,1,C#5,1,H4,1,A4,5 <157>
179 DATA D5,3,C#5,1,H4,1,C#5,1,D5,2,E5,1,0 <208>
180 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1,F#5,2,A5,2,G5,2,F#5,1,D5,3,F#5,2,E5,6 <28>
181 DATA F#5,2,E5,6,F#5,2,E5,2,D5,1,E5,1,F#5,2,A5,2,G5,2,F#5,1,D5,1 <87>
182 DATA E5,1,C#5,1,H4,1,A4,3,H4,1,C#5,1,183 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,H4,4,H4,4,C#5,1,C#5,1 <76>
184 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2,E5,1,A4,3 <212>

```

```

185 DATA D5,3,D5,1,D5,1,E5,1,D5,1,C#5,1,H4,4,H4,4,C#5,1,C#5,1 <78>
186 DATA D5,1,E5,3,D5,1,E5,1,F#5,2,F#5,2,E5,1,D5,7,P,1 <201>
187 DATA ENDE <27>
188 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A3,4,D3,4,A3,2,E4,1,A3,1 <100>
189 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A3,4,D3,4,A3,2,E4,1,A3,1 <101>
190 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A3,4,A3,4 <68>
191 DATA D3,4,D3,4,A3,4,A3,4,H3,4,G3,4,A3,4,A3,3 <29>
192 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A3,4,A3,3 <140>
193 DATA A3,1,A3,7,A3,1,A3,8,D3,4,D3,4,A3,4,A3,4 <181>
194 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A3,4,D3,4,A3,2,E4,1,A3,1 <106>
195 DATA D3,4,D3,4,G3,4,H3,2,D3,2,A3,4,A3,4,D3,4,D3,8 <80>
196 DATA ENDE,ENDE <149>
ENDE DES LISTINGS

```

Datengenerator

```

90 GOTO190 <176>
120 FORB=0TO15:READDA:POKEBA+B,DA:SU=SU+DA:NEXTB <163>
130 READSD:IFSD<>SUTHENPRINT"DATAFEHLER(SPACE)IN(SPACE)ZEILE"ZE:END <222>
140 ZE=ZE+1:BA=BA+16 <220>
150 NEXTA <154>
160 SYSBB <136>
170 : <228>
180 : <238>
190 PRINT"{CLEAR}MUSIC-MASTER(SPACE)DATAGENERATOR":PRINT:PRINT <77>
200 INPUT"STARTADRESSE";SA:IFSA=0THENPRINT"UP2":GOTO200 <219>
210 INPUT"ENDADRESSE(SPACE2)";EA:IFEA<SA THENPRINT"UP2":GOTO210 <142>
220 PRINT"{CLEAR}260(SPACE)SA="SA":EA="EA <158>
230 PRINT"GOTO(SPACE)250" <245>
240 POKE631,19:POKE632,13:POKE633,13:POKE634,13:POKE635,13:POKE636,13:POKE637,13:POKE638,13:POKE639,13:POKE640,13:POKE641,13:POKE642,13:POKE643,13:POKE644,13:POKE645,13:POKE646,13:POKE647,13:POKE648,13:POKE649,13:POKE650,13:POKE651,13:POKE652,13:POKE653,13:POKE654,13:POKE655,13:POKE656,13:POKE657,13:POKE658,13:POKE659,13:POKE660,13:POKE661,13:POKE662,13:POKE663,13:POKE664,13:POKE665,13:POKE666,13:POKE667,13:POKE668,13:POKE669,13:POKE670,13:POKE671,13:POKE672,13:POKE673,13:POKE674,13:POKE675,13:POKE676,13:POKE677,13:POKE678,13:POKE679,13:POKE680,13:POKE681,13:POKE682,13:POKE683,13:POKE684,13:POKE685,13:POKE686,13:POKE687,13:POKE688,13:POKE689,13:POKE690,13:POKE691,13:POKE692,13:POKE693,13:POKE694,13:POKE695,13:POKE696,13:POKE697,13:POKE698,13:POKE699,13:POKE700,13:POKE701,13:POKE702,13:POKE703,13:POKE704,13:POKE705,13:POKE706,13:POKE707,13:POKE708,13:POKE709,13:POKE710,13:POKE711,13:POKE712,13:POKE713,13:POKE714,13:POKE715,13:POKE716,13:POKE717,13:POKE718,13:POKE719,13:POKE720,13:POKE721,13:POKE722,13:POKE723,13:POKE724,13:POKE725,13:POKE726,13:POKE727,13:POKE728,13:POKE729,13:POKE730,13:POKE731,13:POKE732,13:POKE733,13:POKE734,13:POKE735,13:POKE736,13:POKE737,13:POKE738,13:POKE739,13:POKE740,13:POKE741,13:POKE742,13:POKE743,13:POKE744,13:POKE745,13:POKE746,13:POKE747,13:POKE748,13:POKE749,13:POKE750,13:POKE751,13:POKE752,13:POKE753,13:POKE754,13:POKE755,13:POKE756,13:POKE757,13:POKE758,13:POKE759,13:POKE760,13:POKE761,13:POKE762,13:POKE763,13:POKE764,13:POKE765,13:POKE766,13:POKE767,13:POKE768,13:POKE769,13:POKE770,13:POKE771,13:POKE772,13:POKE773,13:POKE774,13:POKE775,13:POKE776,13:POKE777,13:POKE778,13:POKE779,13:POKE780,13:POKE781,13:POKE782,13:POKE783,13:POKE784,13:POKE785,13:POKE786,13:POKE787,13:POKE788,13:POKE789,13:POKE790,13:POKE791,13:POKE792,13:POKE793,13:POKE794,13:POKE795,13:POKE796,13:POKE797,13:POKE798,13:POKE799,13:POKE800,13:POKE801,13:POKE802,13:POKE803,13:POKE804,13:POKE805,13:POKE806,13:POKE807,13:POKE808,13:POKE809,13:POKE810,13:POKE811,13:POKE812,13:POKE813,13:POKE814,13:POKE815,13:POKE816,13:POKE817,13:POKE818,13:POKE819,13:POKE820,13:POKE821,13:POKE822,13:POKE823,13:POKE824,13:POKE825,13:POKE826,13:POKE827,13:POKE828,13:POKE829,13:POKE830,13:POKE831,13:POKE832,13:POKE833,13:POKE834,13:POKE835,13:POKE836,13:POKE837,13:POKE838,13:POKE839,13:POKE840,13:POKE841,13:POKE842,13:POKE843,13:POKE844,13:POKE845,13:POKE846,13:POKE847,13:POKE848,13:POKE849,13:POKE850,13:POKE851,13:POKE852,13:POKE853,13:POKE854,13:POKE855,13:POKE856,13:POKE857,13:POKE858,13:POKE859,13:POKE860,13:POKE861,13:POKE862,13:POKE863,13:POKE864,13:POKE865,13:POKE866,13:POKE867,13:POKE868,13:POKE869,13:POKE870,13:POKE871,13:POKE872,13:POKE873,13:POKE874,13:POKE875,13:POKE876,13:POKE877,13:POKE878,13:POKE879,13:POKE880,13:POKE881,13:POKE882,13:POKE883,13:POKE884,13:POKE885,13:POKE886,13:POKE887,13:POKE888,13:POKE889,13:POKE890,13:POKE891,13:POKE892,13:POKE893,13:POKE894,13:POKE895,13:POKE896,13:POKE897,13:POKE898,13:POKE899,13:POKE900,13:POKE901,13:POKE902,13:POKE903,13:POKE904,13:POKE905,13:POKE906,13:POKE907,13:POKE908,13:POKE909,13:POKE910,13:POKE911,13:POKE912,13:POKE913,13:POKE914,13:POKE915,13:POKE916,13:POKE917,13:POKE918,13:POKE919,13:POKE920,13:POKE921,13:POKE922,13:POKE923,13:POKE924,13:POKE925,13:POKE926,13:POKE927,13:POKE928,13:POKE929,13:POKE930,13:POKE931,13:POKE932,13:POKE933,13:POKE934,13:POKE935,13:POKE936,13:POKE937,13:POKE938,13:POKE939,13:POKE940,13:POKE941,13:POKE942,13:POKE943,13:POKE944,13:POKE945,13:POKE946,13:POKE947,13:POKE948,13:POKE949,13:POKE950,13:POKE951,13:POKE952,13:POKE953,13:POKE954,13:POKE955,13:POKE956,13:POKE957,13:POKE958,13:POKE959,13:POKE960,13:POKE961,13:POKE962,13:POKE963,13:POKE964,13:POKE965,13:POKE966,13:POKE967,13:POKE968,13:POKE969,13:POKE970,13:POKE971,13:POKE972,13:POKE973,13:POKE974,13:POKE975,13:POKE976,13:POKE977,13:POKE978,13:POKE979,13:POKE980,13:POKE981,13:POKE982,13:POKE983,13:POKE984,13:POKE985,13:POKE986,13:POKE987,13:POKE988,13:POKE989,13:POKE990,13:POKE991,13:POKE992,13:POKE993,13:POKE994,13:POKE995,13:POKE996,13:POKE997,13:POKE998,13:POKE999,13:POKE1000,13:POKE1001,13:POKE1002,13:POKE1003,13:POKE1004,13:POKE1005,13:POKE1006,13:POKE1007,13:POKE1008,13:POKE1009,13:POKE1010,13:POKE1011,13:POKE1012,13:POKE1013,13:POKE1014,13:POKE1015,13:POKE1016,13:POKE1017,13:POKE1018,13:POKE1019,13:POKE1020,13:POKE1021,13:POKE1022,13:POKE1023,13:POKE1024,13:POKE1025,13:POKE1026,13:POKE1027,13:POKE1028,13:POKE1029,13:POKE1030,13:POKE1031,13:POKE1032,13:POKE1033,13:POKE1034,13:POKE1035,13:POKE1036,13:POKE1037,13:POKE1038,13:POKE1039,13:POKE1040,13:POKE1041,13:POKE1042,13:POKE1043,13:POKE1044,13:POKE1045,13:POKE1046,13:POKE1047,13:POKE1048,13:POKE1049,13:POKE1050,13:POKE1051,13:POKE1052,13:POKE1053,13:POKE1054,13:POKE1055,13:POKE1056,13:POKE1057,13:POKE1058,13:POKE1059,13:POKE1060,13:POKE1061,13:POKE1062,13:POKE1063,13:POKE1064,13:POKE1065,13:POKE1066,13:POKE1067,13:POKE1068,13:POKE1069,13:POKE1070,13:POKE1071,13:POKE1072,13:POKE1073,13:POKE1074,13:POKE1075,13:POKE1076,13:POKE1077,13:POKE1078,13:POKE1079,13:POKE1080,13:POKE1081,13:POKE1082,13:POKE1083,13:POKE1084,13:POKE1085,13:POKE1086,13:POKE1087,13:POKE1088,13:POKE1089,13:POKE1090,13:POKE1091,13:POKE1092,13:POKE1093,13:POKE1094,13:POKE1095,13:POKE1096,13:POKE1097,13:POKE1098,13:POKE1099,13:POKE1100,13:POKE1101,13:POKE1102,13:POKE1103,13:POKE1104,13:POKE1105,13:POKE1106,13:POKE1107,13:POKE1108,13:POKE1109,13:POKE1110,13:POKE1111,13:POKE1112,13:POKE1113,13:POKE1114,13:POKE1115,13:POKE1116,13:POKE1117,13:POKE1118,13:POKE1119,13:POKE1120,13:POKE1121,13:POKE1122,13:POKE1123,13:POKE1124,13:POKE1125,13:POKE1126,13:POKE1127,13:POKE1128,13:POKE1129,13:POKE1130,13:POKE1131,13:POKE1132,13:POKE1133,13:POKE1134,13:POKE1135,13:POKE1136,13:POKE1137,13:POKE1138,13:POKE1139,13:POKE1140,13:POKE1141,13:POKE1142,13:POKE1143,13:POKE1144,13:POKE1145,13:POKE1146,13:POKE1147,13:POKE1148,13:POKE1149,13:POKE1150,13:POKE1151,13:POKE1152,13:POKE1153,13:POKE1154,13:POKE1155,13:POKE1156,13:POKE1157,13:POKE1158,13:POKE1159,13:POKE1160,13:POKE1161,13:POKE1162,13:POKE1163,13:POKE1164,13:POKE1165,13:POKE1166,13:POKE1167,13:POKE1168,13:POKE1169,13:POKE1170,13:POKE1171,13:POKE1172,13:POKE1173,13:POKE1174,13:POKE1175,13:POKE1176,13:POKE1177,13:POKE1178,13:POKE1179,13:POKE1180,13:POKE1181,13:POKE1182,13:POKE1183,13:POKE1184,13:POKE1185,13:POKE1186,13:POKE1187,13:POKE1188,13:POKE1189,13:POKE1190,13:POKE1191,13:POKE1192,13:POKE1193,13:POKE1194,13:POKE1195,13:POKE1196,13:POKE1197,13:POKE1198,13:POKE1199,13:POKE1200,13:POKE1201,13:POKE1202,13:POKE1203,13:POKE1204,13:POKE1205,13:POKE1206,13:POKE1207,13:POKE1208,13:POKE1209,13:POKE1210,13:POKE1211,13:POKE1212,13:POKE1213,13:POKE1214,13:POKE1215,13:POKE1216,13:POKE1217,13:POKE1218,13:POKE1219,13:POKE1220,13:POKE1221,13:POKE1222,13:POKE1223,13:POKE1224,13:POKE1225,13:POKE1226,13:POKE1227,13:POKE1228,13:POKE1229,13:POKE1230,13:POKE1231,13:POKE1232,13:POKE1233,13:POKE1234,13:POKE1235,13:POKE1236,13:POKE1237,13:POKE1238,13:POKE1239,13:POKE1240,13:POKE1241,13:POKE1242,13:POKE1243,13:POKE1244,13:POKE1245,13:POKE1246,13:POKE1247,13:POKE1248,13:POKE1249,13:POKE1250,13:POKE1251,13:POKE1252,13:POKE1253,13:POKE1254,13:POKE1255,13:POKE1256,13:POKE1257,13:POKE1258,13:POKE1259,13:POKE1260,13:POKE1261,13:POKE1262,13:POKE1263,13:POKE1264,13:POKE1265,13:POKE1266,13:POKE1267,13:POKE1268,13:POKE1269,13:POKE1270,13:POKE1271,13:POKE1272,13:POKE1273,13:POKE1274,13:POKE1275,13:POKE1276,13:POKE1277,13:POKE1278,13:POKE1279,13:POKE1280,13:POKE1281,13:POKE1282,13:POKE1283,13:POKE1284,13:POKE1285,13:POKE1286,13:POKE1287,13:POKE1288,13:POKE1289,13:POKE1290,13:POKE1291,13:POKE1292,13:POKE1293,13:POKE1294,13:POKE1295,13:POKE1296,13:POKE1297,13:POKE1298,13:POKE1299,13:POKE1300,13:POKE1301,13:POKE1302,13:POKE1303,13:POKE1304,13:POKE1305,13:POKE1306,13:POKE1307,13:POKE1308,13:POKE1309,13:POKE1310,13:POKE1311,13:POKE1312,13:POKE1313,13:POKE1314,13:POKE1315,13:POKE1316,13:POKE1317,13:POKE1318,13:POKE1319,13:POKE1320,13:POKE1321,13:POKE1322,13:POKE1323,13:POKE1324,13:POKE1325,13:POKE1326,13:POKE1327,13:POKE1328,13:POKE1329,13:POKE1330,13:POKE1331,13:POKE1332,13:POKE1333,13:POKE1334,13:POKE1335,13:POKE1336,13:POKE1337,13:POKE1338,13:POKE1339,13:POKE1340,13:POKE1341,13:POKE1342,13:POKE1343,13:POKE1344,13:POKE1345,13:POKE1346,13:POKE1347,13:POKE1348,13:POKE1349,13:POKE1350,13:POKE1351,13:POKE1352,13:POKE1353,13:POKE1354,13:POKE1355,13:POKE1356,13:POKE1357,13:POKE1358,13:POKE1359,13:POKE1360,13:POKE1361,13:POKE1362,13:POKE1363,13:POKE1364,13:POKE1365,13:POKE1366,13:POKE1367,13:POKE1368,13:POKE1369,13:POKE1370,13:POKE1371,13:POKE1372,13:POKE1373,13:POKE1374,13:POKE1375,13:POKE1376,13:POKE1377,13:POKE1378,13:POKE1379,13:POKE1380,13:POKE1381,13:POKE1382,13:POKE1383,13:POKE1384,13:POKE1385,13:POKE1386,13:POKE1387,13:POKE1388,13:POKE1389,13:POKE1390,13:POKE1391,13:POKE1392,13:POKE1393,13:POKE1394,13:POKE1395,13:POKE1396,13:POKE1397,13:POKE1398,13:POKE1399,13:POKE1400,13:POKE1401,13:POKE1402,13:POKE1403,13:POKE1404,13:POKE1405,13:POKE1406,13:POKE1407,13:POKE1408,13:POKE1409,13:POKE1410,13:POKE1411,13:POKE1412,13:POKE1413,13:POKE1414,13:POKE1415,13:POKE1416,13:POKE1417,13:POKE1418,13:POKE1419,13:POKE1420,13:POKE1421,13:POKE1422,13:POKE1423,13:POKE1424,13:POKE1425,13:POKE1426,13:POKE1427,13:POKE1428,13:POKE1429,13:POKE1430,13:POKE1431,13:POKE1432,13:POKE1433,13:POKE1434,13:POKE1435,13:POKE1436,13:POKE1437,13:POKE1438,13:POKE1439,13:POKE1440,13:POKE1441,13:POKE1442,13:POKE1443,13:POKE1444,13:POKE1445,13:POKE1446,13:POKE1447,13:POKE1448,13:POKE1449,13:POKE1450,13:POKE1451,13:POKE1452,13:POKE1453,13:POKE1454,13:POKE1455,13:POKE1456,13:POKE1457,13:POKE1458,13:POKE1459,13:POKE1460,13:POKE1461,13:POKE1462,13:POKE1463,13:POKE1464,13:POKE1465,13:POKE1466,13:POKE1467,13:POKE1468,13:POKE1469,13:POKE1470,13:POKE1471,13:POKE1472,13:POKE1473,13:POKE1474,13:POKE1475,13:POKE1476,13:POKE1477,13:POKE1478,13:POKE1479,13:POKE1480,13:POKE1481,13:POKE1482,13:POKE1483,13:POKE1484,13:POKE1485,13:POKE1486,13:POKE1487,13:POKE1488,13:POKE1489,13:POKE1490,13:POKE1491,13:POKE1492,13:POKE1493,13:POKE1494,13:POKE1495,13:POKE1496,13:POKE1497,13:POKE1498,13:POKE1499,13:POKE1500,13:POKE1501,13:POKE1502,13:POKE1503,13:POKE1504,13:POKE1505,13:POKE1506,13:POKE1507,13:POKE1508,13:POKE1509,13:POKE1510,13:POKE1511,13:POKE1512,13:POKE1513,13:POKE1514,13:POKE1515,13:POKE1516,13:POKE1517,13:POKE1518,13:POKE1519,13:POKE1520,13:POKE1521,13:POKE1522,13:POKE1523,13:POKE1524,13:POKE1525,13:POKE1526,13:POKE1527,13:POKE1528,13:POKE1529,13:POKE1530,13:POKE1531,13:POKE1532,13:POKE1533,13:POKE1534,13:POKE1535,13:POKE1536,13:POKE1537,13:POKE1538,13:POKE1539,13:POKE1540,13:POKE1541,13:POKE1542,13:POKE1543,13:POKE1544,13:POKE1545,13:POKE1546,13:POKE1547,13:POKE1548,13:POKE1549,13:POKE1550,13:POKE1551,13:POKE1552,13:POKE1553,13:POKE1554,13:POKE1555,13:POKE1556,13:POKE1557,13:POKE1558,13:POKE1559,13:POKE1560,13:POKE1561,13:POKE1562,13:POKE1563,13:POKE1564,13:POKE1565,13:POKE1566,13:POKE1567,13:POKE1568,13:POKE1569,13:POKE1570,13:POKE1571,13:POKE1572,13:POKE1573,13:POKE1574,13:POKE1575,13:POKE1576,13:POKE1577,13:POKE1578,13:POKE1579,13:POKE1580,13:POKE1581,13:POKE1582,13:POKE1583,13:POKE1584,13:POKE1585,13:POKE1586,13:POKE1587,13:POKE1588,13:POKE1589,13:POKE1590,13:POKE1591,13:POKE1592,13:POKE1593,13:POKE1594,13:POKE1595,13:POKE1596,13:POKE1597,13:POKE1598,13:POKE1599,13:POKE1600,13:POKE1601,13:POKE1602,13:POKE1603,13:POKE1604,13:POKE1605,13:POKE1606,13:POKE1607,13:POKE1608,13:POKE1609,13:POKE1610,13:POKE1611,13:POKE1612,13:POKE1613,13:POKE1614,13:POKE1615,13:POKE1616,13:POKE1617,13:POKE1618,13:POKE1619,13:POKE1620,13:POKE162
```


Neu: Sonderzeichen werden durch Klartext ersetzt!

Eingabe-Hinweise für Commodore C-16/116-Benutzer!

Um Ihnen die Eingabe unserer Listings zu erleichtern, werden zukünftig all C-16/116-Listings ohne Steuerzeichen abgedruckt. Alle Steuerzeichen werden in unseren Listings durch Klartext, welcher die Taste kennzeichnet, ersetzt. Eine Tabelle der Tastenfunktionen finden Sie in jedem Tronic-Magazin.

Tastenbezeichnungen für Steuercodes

DOWN	CURSOR UNTEN	TASTE NEBEN RECHTEM SHIFT	CYAN	TUERKIS	CONTROL-TASTE & 4
UP	CURSOR HOCH	SHIFT-TASTE & TASTE NEBEN RECHTEM SHIFT	PURPLE	PURPUR	CONTROL-TASTE & 5
CLR	CLEAR SCHIRM	SHIFT-TASTE & 2. TASTE GANZ RECHTS OBEN	GREEN	GRUEN	CONTROL-TASTE & 6
INST	EINFUEGEN	SHIFT-TASTE & TASTE GANZ RECHTS OBEN	BLUE	BLAU	CONTROL-TASTE & 7
HOME	CURSOR IN ECKE	2. TASTE VON GANZ RECHTS OBEN	YELLOW	GELB	CONTROL-TASTE & 8
DEL	DELETE	TASTE GANZ RECHTS OBEN	RVSON	INVERSE EIN	CONTROL-TASTE & 9
RIGHT	CURSOR RECHTS	TASTE GANZ RECHTS UNTEN	RVSOFF	INVERSE AUS	CONTROL-TASTE & 0
LEFT	CURSOR LINKS	SHIFT-TASTE & TASTE UNTEN RECHTS	ORANGE	ORANGE	COMMODORE-TASTE & 1
SPACE	LEERZEICHEN	LEERTASTE (GROESSTE TASTE)	BROWN	BRAUN	COMMODORE-TASTE & 2
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1	LIG.RED	HELLROT	COMMODORE-TASTE & 3
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3	DGREY	DUNKELGRAU	COMMODORE-TASTE & 4
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5	MGREY	MITTELGRAU	COMMODORE-TASTE & 5
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7	LIG.GREEN	HELLGRUEN	COMMODORE-TASTE & 6
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2	LIG.BLUE	HELLBLAU	COMMODORE-TASTE & 7
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4	HGREY	HELLGRAU	COMMODORE-TASTE & 8
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6	CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN MIT DEM NACHFOLGENDEN ZEICHEN. (Z.B. CTRLA ENTSPRICHT CTRL & A)
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8			CONTROL-TASTE &
BLACK	SCHWARZ	CONTROL-TASTE & 1	FLASH ON	3BINKEN EIN	CONTROL-TASTE &
WHITE	WEISS	CONTROL-TASTE & 2	FLASH OFF	3BINKEN AUS	CONTROL-TASTE &
RED	ROT	CONTROL-TASTE & 3			

Achtung: Einige Listings enthalten das Zeichen "-", dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben)
das Zeichen "^" steht für den Pfeil nach oben,
das Zeichen "~" steht für das (engl.) Pfund-Symbol (Taste neben Restore)

Folgendes Beispiel demonstriert die Arbeitsweise unseres Systems:

STANDARD AUSDRUCK:

```
10 A$="TEST":PRINT"HALLO LEUTE !":PRINT"TRONIC VERLAG"
20 PRINT"ENDE"
```

UNSER NEUER AUSDRUCK DER GLEICHEN ZEILEN:

```
10 A$="TEST":PRINT"⟨CLEAR RIGHT3 DOWN3 R
ED⟩HALLO⟨RIGHT2 BLUE⟩LEUTE⟨SPACE⟩!":PRIN
T"TRONIC⟨SPACE⟩VERLAG"
20 PRINT"⟨CYAN PURPLE GREEN⟩ENDE"
⟨34⟩
⟨117⟩
```

Wie in dem Beispiel zu erkennen ist, wird das Herz-Symbol durch das Wort CLEAR ersetzt. Dies bedeutet, Sie müssen die Taste CLEAR drücken, um das Herz-Symbol zu erzeugen. Sollten Sie einmal nicht wissen, welche Taste gemeint ist, hilft Ihnen ein Blick in unsere Tastentabelle bestimmt weiter.

Um im Listing Tastenbezeichnungen von Grafik oder normalen Texten zu unterscheiden, werden alle Tastenkennzeichnungen in geschweifte Klammern gesetzt. Diese dürfen selbstverständlich nicht eingegeben werden. Auch Leerzeichen innerhalb geschweiften Klammern

dienen nur zur Trennung einzelner Tastenfunktionen und dürfen ebenfalls nicht eingegeben werden. Um die Tastenfunktionen noch besser hervorzuheben, werden diese in unterstrichener Kursivschrift (Schrägschrift) abgedruckt.

Steht hinter einer Tastenfunktion eine Zahl, welche ebenfalls unterstrichen ist, bedeutet dies, daß die letzte Tastenfunktion mehrmals betätigt werden muß. Die in unserem Beispiel abgedruckte Funktion RIGHT3 bedeutet, daß die Taste RIGHT (Cursor rechts) 3 mal hintereinander betätigt werden muß. Auch einzelne oder mehrere Leer-

zeichen innerhalb von Anführungszeichen werden auf diese Art gekennzeichnet. Das bislang übliche Abzählen einzelner Zeichen entfällt somit völlig.

Alle Zeichen außerhalb von den geschweiften Klammern werden normal abgedruckt und auch eingegeben.

Auf den ersten Blick hört sich das sicher etwas kompliziert an, ist jedoch in der Praxis ganz einfach. Wenn man sich erst an die in Klartext geschriebenen Steuerzeichen gewöhnt hat, wird man den großen Vorteil dieser Schreibweise erkennen.

Checksummer OC Version 1.0 für C-16/116

Die Arbeitsweise des C-16/116 Checksummers entspricht weitgehend der VC-20- und C-64-Version und braucht deshalb nicht näher erklärt zu werden. Da auch der C-16/116 nicht den größten Speicher besitzt, wurde hier die Methode der VC-20-Version gewählt, und das Maschinenprogramm in den Kassettenpuffer gelegt. Dies hat den Vorteil, daß kein Basic-Speicher verlorengeht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchgeführt werden dürfen, da diese unweigerlich zum Systemabsturz führen. Um die Kassettenoperationen wieder zuzulassen, muß der Checksummer durch die beiden folgenden SYS-Aufrufe abgeschaltet werden:

SYS 62158 : SYS 33047

Aktiviert wird der Checksummer mit **SYS 818**. Die Funktion und Arbeitsweise entnehmen Sie bitte der C64-Checksummer-Version. Eine Ausnahme bei C-16/116 sind

die zusätzlichen Steuerzeichen 'FLASHON' (CTRL-) und 'FLASHOFF' (CTRL-) welche ebenfalls durch Klartext ersetzt werden (s. Tabelle Tastenbezeichnungen).

Arbeitsweise und Aufbau unseres Checksummers:

Unser Checksummer besteht aus einem kleinen Maschinenprogramm welches als Basic-Loader abgedruckt ist.

Tippen Sie diesen Loader ein und speichern ihn auf Kassette oder Diskette, denn Sie können ihn zukünftig immer wieder benutzen.

Der Start erfolgt durch den Befehl „RUN“. Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC ...“. Der Checksummer ist nun aktiv und man kann ein beliebiges Tronic-Listing eingeben. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem

Schirm eine Prüfzahl. Vergleichen Sie diese mit der Zahl im Heft hinter der Zeile. Stimmt die Zahl überein, ist die Zeile richtig eingegeben.

Auf diese Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben.

Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern. Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben.

Für alle, die nicht gerne abtippen, ist der Checksummer unter folgender Bestellnummer zu beziehen:

OV 10 K Kassette 10 DM
OV 10 D Diskette 15 DM

```
1 REM *****
2 REM *
3 REM * C16-CHECKSUMMER
4 REM *
5 REM * COPYRIGHT BY FRANK BRALL
6 REM *
7 REM * (C) 18.04.86
8 REM *
9 REM *****
10 PRINT"⟨CLEAR DOWN SPACE2⟩♦♦♦⟨SPACE2⟩
CHECKSUMMER⟨SPACE⟩OC⟨SPACE⟩1.0⟨SPACE2⟩♦♦
♦♦⟨DOWN⟩"
11 PRINT"⟨SPACE2⟩♦♦⟨SPACE2⟩COPYRIGHT⟨SPA
CE⟩FRANK⟨SPACE⟩BRALL⟨SPACE3⟩♦♦⟨DOWN⟩"
12 PRINT"⟨SPACE3⟩FUER⟨SPACE⟩TRONIC-VERLA
G"
60000 DIMH(75):FORI=0TO7
60010 H(48+I)=I:H(65+I)=I+10:NEXT
60020 FORI= 818 TO 1010 :READA$
60030 H=ASC(LEFT$(A$,1))
60040 L=ASC(RIGHT$(A$,1))
60050 D=H(H)*16+H(L):S=S+D:POKEI,D
60060 A=A+1:IFA<20THENNEXT:A=-1
60070 READV:Z=Z+1:IFV=STHEN60085
60080 PRINT"DATAFEHLER⟨SPACE⟩IN⟨SPACE⟩ZE
ILE⟨SPACE⟩":;60200+Z:END
60085 IFA<0THEN60100
60090 S=0:A=0:NEXT
60100 PRINT"⟨DOWN2⟩"
60110 PRINT"CHECKSUMMER⟨SPACE⟩EIN⟨SPACE⟩
=⟨SPACE⟩SYS⟨SPACE⟩818⟨DOWN⟩"
60120 PRINT"CHECKSUMMER⟨SPACE⟩AUS⟨SPACE⟩
=⟨SPACE⟩SYS⟨SPACE⟩62158+SYS⟨SPACE⟩33047⟨
DOWN⟩"
60130 PRINT"⟨DOWN⟩SAVE/LOAD⟨SPACE⟩IST⟨SP
ACE⟩WAEREND⟨SPACE⟩DER⟨SPACE⟩CHECK-"
60140 PRINT"SUMMER⟨SPACE⟩AKTIV⟨SPACE⟩IST
,⟨SPACE⟩NICHT⟨SPACE⟩MOEGLICH⟨SPACE⟩!"
60150 PRINT"⟨DOWN⟩BEACHTEN⟨SPACE⟩SIE⟨SPA
CE⟩DIE⟨SPACE⟩HINWEISE⟨SPACE⟩IN⟨SPACE⟩DEN
"
60170 PRINT"HEFTEN⟨SPACE⟩COMPUTE⟨SPACE⟩M
IT⟨SPACE⟩UND⟨SPACE⟩COMPUTRONIC"
60180 PRINT"⟨DOWN⟩VIEL⟨SPACE⟩SPASS⟨SPACE
⟩!⟨SPACE⟩(AUTOR:F.BRALL/6443SONTRA)
60190 POKE 814,147:POKE815,251:POKE 816,
147:POKE 817,251
60195 SYS 818:END
60201 DATA A9,3D,8D,02,03,A9,03,8D,03,03
,60,A2,FF,86,3A,20,5A,8B,86,3B, 1851
60202 DATA B4,3C,20,73,04,AA,F0,EF,90,09
,20,53,89,20,79,04,4C,D9,8B,20, 2018
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D
,E4,03,8D,E5,03,8D,E6,03,18,A5, 2076
60204 DATA 14,65,15,8D,E6,03,A0,FF,CB,B1
,3B,F0,2C,C9,22,D0,0A,AD,E4,03, 2508
60205 DATA 49,01,8D,E4,03,B1,3B,AE,E4,03
,D0,04,C9,20,F0,E4,EE,E5,03,AE, 2644
60206 DATA E5,03,18,B1,3B,6D,E6,03,8D,E6
,03,CA,D0,F4,4C,76,03,3B,20,39, 2204
60207 DATA DB,8E,E2,03,8C,E3,03,A2,00,A0
,00,1B,20,39,DB,A9,5B,20,D2,FF, 2365
60208 DATA A9,00,AE,E6,03,20,5F,A4,A9,5D
,20,D2,FF,20,0B,FB,20,0B,FB,AC, 2380
60209 DATA E3,03,AE,E2,03,18,20,39,DB,A2
,00,86,FF,4C,36,87,00,00,00,00, 1778
60210 DATA 00,00,00,00,00,00,00,00,00,00
,00,00,00, 0
```


19


```

1780 CHAR1,C,23,"_":CHAR1,C,22,"{SPACE}" <45>
1790 FORA=1023TO850STEP-10 <181>
1800 SOUND3,A,1:NEXTA <200>
1810 FORA=1TO200:NEXTA:CHAR1,C,23,"{SPACE" <218>
E}" <89>
1820 LI=LI-1:E=1 <50>
1830 PRINT "{WHITE HOME DOWN RIGHT 23}";LI

```

```

1840 IFLI=0THENGOTO1850:ELSERETURN <133>
1850 IFPU>RETHENRE=PU <85>
1860 CHAR1,12,19,"{YELLOW RVSON FLASHON}"
G{SPACE}A{SPACE}M{SPACE}E{SPACE2}D{SPACE" <186>
3V{SPACE}E{SPACE}R{RVSOFF FLASHOFF}"
1870 GETA$:IFA$="{SPACE}"THEN550:ELSE187 <83>
0

```

Tips & Tricks für den Commodore 16/116

Cursor setzen

Ein altes Problem der Commodore Rechner ist der fehlende Befehl zur Positionierung des Cursors. Zwar gibt es beim C-16/116 den Char-Befehl, jedoch läßt sich hiermit nur ein Zeichen ausgeben. Der eigentliche Cursor bleibt an der alten Stelle. Es fehlt ein Befehl, der es erlaubt, ohne folgende Ausgaben den Cursor direkt auf eine beliebige Bildschirmposition zu setzen (zum Beispiel, um anschließend aus der zugehörigen Position im Bildspeicher mit „PEEK“ das Zeichen an der momentanen Bildschirmposition zu lesen).

In der sogenannten „Zeropage“ des C-16/116 existieren zwei Speicherstellen, in denen die Spalte und Zeile der aktuellen Cursorposition festgehalten ist (Spalte in 202, Zeile in 205).

Durch Veränderung der Inhalte dieser Speicherstellen kann der Cursor auf eine beliebige Position gesetzt werden. Anschließend muß jedoch noch eine Betriebssystemroutine mit **SYS 55464** aufgerufen werden. Erst nach diesem Aufruf ist dem Betriebssystem die neue Situation bekannt.

(Speicherplatz-) Not macht erfinderisch

Auf der Suche nach einer Möglichkeit dem begrenzten Speicherplatz des C-16 ein Schnippchen zu schlagen, bin ich auf die folgende interessante Idee zum Einsparen von Speicherplatz gestoßen: Daß Integervariablen (z. B. A%) in nur 2 Bytes (im Gegensatz zu 7 Bytes bei normalen Variablen) abgelegt werden können, ist den meisten Basicprogrammierern bekannt. Dieser Vorteil kommt allerdings erst richtig zum Tragen, wenn man im Programm größere Felder verwendet. Probieren Sie mal

```
CLR: DIM A(100): PRINT FRE(X)
```

und vergleichen Sie dann mit

```
CLR: DIM A%(100): PRINT FRE(X)!
```

Bleibt das Problem, daß aus Genauigkeitsgründen meist einige Nachkommastellen nötig sind, Integervariable aber nur ganzzahlige Werte aufnehmen können. Kennt man jedoch in etwa den Wertebereich der Zahlen, die das Feld aufnehmen soll, kann man dieses Problem umgehen, indem man nur die reine Ziffernfolge in dem Integerfeld ablegt und diese erst bei der Weiterverarbeitung wieder in die ursprüngliche Zahl zurückverwandelt. Dabei muß

gewährleistet sein, daß der Wertebereich für Integerzahlen (-32768, +32767) nicht verlassen wird.

Folgendes Beispiel soll die Vorgehensweise verdeutlichen:

1. Schritt

Der im Integerfeld abzulegende Wert wird mit allen Nachkommastellen berechnet und einer Variablen zugewiesen.

```
A = .... (Ergebnis z. B. A = 4.926538)
```

2. Schritt

Der Wert wird auf 3 Nachkommastellen gerundet und mit 1000 multipliziert; die entstandene Ziffernfolge wird dem Integerfeld zugewiesen.

```
A%(I) = INT( A * 1000 + 0.5)
```

```
(z. B. A%(45) = 4927)
```

3. Schritt

Vor der Weiterverarbeitung wird die Ziffernfolge in den ursprünglichen Wert zurückverwandelt.

```
B = A%(I)/1000 (z. B. B = 4.927)
```

Mit Hilfe dieses kleinen Tricks ist es gelungen, ein Programm zum Zeichnen von dreidimensionalen Funktionsflächen in dem 2k „großen“ Basic-Speicher des C-16 (bei Verwendung der hochauflösenden Grafik) unterzubringen.

Werkstatt:

INVERS-UTILITY für den C-16/116

Das Programm INVERS-UTILITY erlaubt es, beliebige Bildschirmausschnitte zu invertieren, d. h. alle in diesem Ausschnitt

befindlichen normalen Zeichen erscheinen revers und alle reversen Zeichen erscheinen wieder normal. Besonders in

Verbindung mit einer FILL-UTILITY stellt dieses Programm darum eine nützliche Hilfe dar. Nach dem Eingeben und Starten

des BASIC-Loaders (das Listing) wird zunächst das Maschinenprogramm eingelesen. Danach erscheint auf dem Bildschirm die Frage, ob das reine Maschinenprogramm „INVERS-UTILITY“ auf Diskette oder auf Kassette gespeichert werden soll (der Loader wird danach nicht mehr benötigt und kann gelöscht werden). Wenn Sie das Maschinenprogramm dann später brauchen, können Sie es mit **LOAD „INVERS-UTILITY“,8,1** (von Kassette mit **.1,1**) einladen (die Sekundäradresse darf nicht fehlen) und mit **POKE 55,0: POKE 56,62** vor Überschieben durch BASIC schützen. Die Routine wird gestartet mit dem Befehl **SYS 15872,X1,Y1,X2,Y2**, wobei X1/Y1 und X2/Y2 die Koordinaten von zwei sich diagonal gegenüberliegenden Eckpunkten des rechteckigen Bildschirmausschnittes, der invertiert werden soll sind (X muß zwischen 0 und 39 liegen, Y zwischen 0 und 24).

Beispiel: **SYS 15872,0,0,39,24** invertiert den gesamten Bildschirm.

Noch eine wichtige Anmerkung zum BASIC-Loader: Die ersten drei REM-Zeilen dürfen auf keinen Fall fehlen, da sie den Filenamen für die SAVE-Routine enthalten!

Daniel Durstewitz

```

10 REM ***** <121>
15 REM * <112>
20 REM * INVERS-UTILITY * <59>
25 REM * <122>
30 REM * EINGABEFORM: * <236>
35 REM * <132>
40 REM * SYS15872,X1,Y1,X2,Y2 * <20>
45 REM * <142>
50 REM * SCHUETZEN VOR UEBERSCHREIBEN * <196>
55 REM * DURCH BASIC: * <122>
60 REM * <157>
65 REM * POKE 55,0 : POKE 56,62 * <172>
70 REM * <167>
75 REM * 1986 BY DANIEL DURSTEWITZ * <167>
80 REM * REICHENBERGER STR.26 * <144>
85 REM * 4460 NORDHORN * <166>
90 REM * TEL.05921/79200 * <20>
95 REM * <192>
99 REM ***** <210>
105 : <163>
110 : <168>
115 : <173>
120 : <178>
125 FORT=15872TO16042:READA:POKET,A:P=P+ <204>
A:NEXT <73>
130 IFP=24174GOTO145
135 PRINT "{DOWN} FEHLER {SPACE} IN{SPACE} DE <221>
N{SPACE} DATAS! {SPACE} BITTE {SPACE} UEBERPR
UEFEN"
140 PRINT "SIE {SPACE} DIE {SPACE} ZEILEN {SPA <172>
CE} 195-265 {SPACE} NOCH {SPACE} EINMAL! ":END
145 PRINT "{DOWN} SOLL {SPACE} DAS {SPACE} PRO <23>
GRAMM {SPACE} AUF {SPACE} RVSON} D{RVSOFF} ISK
ETTE {SPACE} ODER"
150 INPUT "AUF {SPACE} RVSON} K{RVSOFF} ASSET <112>
TE {SPACE} ABGESPEICHERT {SPACE} WERDEN":A%
155 IFA$="{D" THEN POKE16011,B:GOTO170 <169>
160 IFA$="{K" THEN POKE16011,1:GOTO170 <104>
165 GOTO145 <0>

```

```

170 SYS16010:END <183>
175 : <233>
180 : <238>
185 : <243>
190 : <248>
195 DATA32,216,157,224,40,144,3,76,28,15 <226>
3,134,216,32,216,157,224,25,176,244
200 DATA134,217,32,216,157,224,40,176,23 <102>
5,134,218,32,216,157,224,25,176,226
205 DATA134,219,165,219,24,229,217,16,13 <203>
,165,217,72,165,219,133,217,104,133
210 DATA219,76,39,62,133,219,165,218,24, <65>
229,216,16,13,165,216,72,165,218
215 DATA133,216,104,133,218,76,61,62,133 <164>
,218,160,12,166,217,165,216,132,217
220 DATA202,48,9,24,105,40,144,248,230,2 <155>
17,208,244,133,216,166,219,232,164
225 DATA218,200,177,216,24,105,128,145,2 <22>
16,136,16,246,165,216,24,105,40,133
230 DATA216,144,2,230,217,202,16,229,76, <183>
220,139 <37>
235 : <42>
240 : <243>
245 REM SAVE-ROUTINE <52>
250 : <57>
255 :
260 DATA162,1,160,1,32,49,244,169,14,162 <71>
,94,160,16,32,12,244,169,0,133,216
265 DATA169,62,133,217,169,216,162,138,1 <131>
60,62,76,148,241

```

SOFTWARE:

Commodore 16/116 CASSETTE

ACE (Air Combat Emulator)	34.00 DM
Beach Head	29.00 DM
Bongo Construction Set	25.00 DM
Commando	25.00 DM
Winter Olympics	32.00 DM
Bandits at Zero	14.95 DM
Spiderman	14.95 DM
Grenlins	25.00 DM
Thai Boxing	25.00 DM
Kung Fu Kid	29.00 DM
Space Pilot	25.00 DM
u.v.a....	

Commodore 64

"V"	33.00 DM
Law of the West	39.00 DM
The last V8	14.95 DM
Master of Magic	14.95 DM
Uridium	29.95 DM
Rebel Planet	39.00 DM
Rock'n Wrestle	39.00 DM
Five-a-side SOCCER	14.95 DM
Psi 5 Trading Company	39.00 DM
Hero of the Golden Talisman	14.95 DM

Legen Sie bei Bestellungen das Geld entweder in bar oder als Scheck bei. Bei Nachnahme-Versand wird ein Porto-Preis von 4.00 DM angerechnet. Telefonische Bestellungen sind **SOFTWARE-VERSAND** ab 18.30 Uhr möglich. Fordern Sie unsere kostenlosen Informationen an!

INH. ANDREAS BACHLER
BLUCHERSTRASSE 24
D-4260 BOCHOLT
☎ 02871/183088
POSTFACH 428

Haben Sie schon Ihren Lohnsteuerjahresausgleich bei Ihrem Finanzamt eingereicht? Wenn nicht, dann können Sie mit dem folgenden Anwendungsprogramm ausrechnen, welche Rückzahlung Sie erhalten!

Lohnsteuerjahresausgleich

ist ein Anwendungsprogramm, das die ganze Vielfalt des Steuerrechts berücksichtigt. Mit seiner Hilfe können Sie das zu versteuernde Einkommen aus nichtselbständiger Arbeit, Kapitalvermögen, Vermietung, Renten (Versorgungsbezüge)

berechnen. Dabei sind die Werbungskosten, die Sonderausgaben und evtl. Freibeträge berücksichtigt. Mit ein bißchen Programmiererfahrung können Sie dies Programm ständig aktualisieren und somit immer die neuesten Daten berücksichtigen.

Bei Eingabe Ihrer persönlichen Daten können Sie sehr schnell und einfach Ihren Erstattungsbetrag errechnen. Auf diese Weise wissen Sie schon zu Beginn des Jahres, was Ihnen Ihr Finanzamt noch schuldig ist.

```

1000 REM"LOHNSTEUERJAHRESAUSGLEICH(SPACE
>85" <255>
1001 REM* BERECHNUNG DES ZU VERSTEU- * <126>
1002 REM* ERNDEN EINKOMMENS FUER 1985* <31>
1003 REM* FUER C16/116 BY ***** <17>
1004 REM* PETER MUENZER ***** <238>
1005 REM* DIESSENHOFERSTR.13 ***** <251>
1006 REM* 7715 BRAEUNLINGEN ***** <87>
1007 REM***** <183>
1008 REM***** C 1985 ***** <63>
1009 REM***** <185>
1010 SONCLR <221>
1020 PRINT"*****" <52>
***** <205>
1030 PRINTTAB(4)"BERECHNUNG(SPACE)DES(SP
ACE)ZU(SPACE)VERSTEUERNDEN" <120>
1040 PRINTTAB(4)"(SPACE11)EINKOMMENS" <83>
1050 PRINT"*****" <185>
***** <113>
1052 PRINT <242>
1053 PRINT"LEDIG(SPACE):(SPACE9)L(SPACE)
EINGEBEN!" <150>
1054 PRINT"VERHEIRATET:(SPACE4)V(SPACE)E
INGEBEN!" <34>
1055 INPUT A$ <226>
1056 IF A$="(SPACE)" THEN GOTO 65535 <43>
1057 IF A$="L" THEN LET Q=1 <192>
1058 IF A$="V" THEN LET Q=2 <193>
1059 PRINT
1060 PRINT
1070 INPUT "BRUTTOLOHN(EHEMANN):(SPACE5)
";L <31>
1080 GOSUB 3605 <154>
1090 PRINT <223>
1100 PRINT"EINKUENFT(SPACE)DER(SPACE)EHE
FRAU(SPACE)AUS(SPACE)NICHTSELBST." <176>
1110 PRINT"ARBEIT?(SPACE7)(J/N)" <26>
1120 INPUT A$: IF A$="(SPACE)" THEN GOTO <22>
1130 IF A$="N" THEN GOTO 1160 <33>
1140 INPUT "BRUTTOLOHN(EHEFRAU):(SPACE5)
";H <61>
1150 IF A$="J" THEN GOSUB 3850 <150>
1160 LET N=INT(L-W-1080) <218>
1170 LET M=INT(H-WK-1080) <10>
1180 IF N<0 THEN LET N=0 <117>
1190 IF M<0 THEN LET M=0 <118>
1200 PRINT "EINKOMMEN(SPACE)ABZUEGLICH(S
PACE)WERBUNGSKOSTEN:" <28>
1210 PRINT <87>
1220 PRINT"EHEMANN:(SPACE3);N;(SPACE)D
M" <225>
1230 PRINT <107>

```

```

1240 PRINT"EHEFRAU:(SPACE3);M;(SPACE)D
M" <9>
1250 PRINT <127>
1260 PRINT" <51>
" <235>
1270 PRINT"GEMEINSAM:(SPACE);N+M;(SPACE
E)DM" <72>
1280 PRINT" <168>
" <224>
1290 PRINT <188>
1300 PRINT"OHNE(SPACE)VERSORGUNGSFREIBET
RAG!" <160>
1310 PRINT <23>
1320 PRINT"EINKUENFTE(SPACE)AUS(SPACE)KA
PITALVERMOEGEN?(SPACE)(J/N)" <93>
1330 INPUT A$: IF A$="(SPACE)" THEN GOTO <106>
1340 IF A$="N" THEN LET ZI=0 <7>
1350 IF A$="J" THEN GOSUB 3950 <248>
1360 PRINT"EINKUENFTE(SPACE)KAPITALVERMO
EGEN:";ZI <167>
1370 PRINT <113>
1375 PRINT" <135>
" <187>
1380 PRINT"EINKUENFTE(NICHTSELBST.ARBEIT
(SPACE)UND" <22>
1390 PRINT"KAPITALVERMOEGEN:(SPACE2);N+
M+ZI;(SPACE)DM" <170>
1395 PRINT" <189>
" <41>
1400 PRINT <85>
1405 PRINT"EINKOMMEN(SPACE)AUS(SPACE)VER
MIETUNG(SPACE)UND(SPACE)VERPACH-" <38>
1410 INPUT"TUNG?(SPACE4)(J/N)";B$ <49>
1412 IF B$="(SPACE)" THEN GOTO 1412 <92>
1414 IF B$="N" THEN LET VE=0 <70>
1416 PRINT <247>
1418 IF B$="J" THEN GOSUB 4200 <198>
1400 PRINT"SIND(SPACE)BEI(SPACE)DEN(SPAC
E)EINKUENFTEN(SPACE)VERSORGUNGS-" <254>
1405 INPUT"BEZUEGE(SPACE)ENTHALTEN(SPACE
2)(J/N)(SPACE)";A$ <12>
1410 IF A$="(SPACE)" THEN GOTO 3410 <237>
1415 IF A$="N" THEN LET VS=0 <22>
1420 IF A$="J" THEN GOSUB 4100 <249>
1430 PRINT <119>
1435 PRINT"VERSORGUNGSFREIBETRAG:(SPACE)
";VS;(SPACE)DM" <38>
1440 PRINT <22>
1450 PRINT"EINKOMMEN(SPACE)NACH(SPACE)AB
ZUG(SPACE)DES(SPACE)VERSORGUNGS-" <22>
1455 PRINT"FREIBETRAGS:(SPACE)";INT(N+M+
ZI-VS) <119>
1456 PRINT <38>

```

```

3457 PRINT"ZU/ABZUEGLICH(SPACE)EINKUENFT
E(SPACE)AUS(SPACE)V+V(SPACE)";VE;"(SPACE
)DM" <187>
3460 LET GG=INT(N+M+ZI+VE-VS) <122>
3465 PRINT"ZWISCHENERGEBNIS(SPACE)MIT(SPACE)
V+V";GG;"(SPACE)DM" <34>
3500 PRINT <82>
3505 PRINT"ALTERSFREIBETRAEGE(SPACE)ODER
(SPACE)ALTERSENTLAS-" <85>
3506 INPUT"BETRAG?(SPACE7)(J/N)";C$ <91>
3508 IF C$="(SPACE)" THEN GOTO 3508 <196>
3509 IF C$="J" THEN GOSUB 4300 <113>
3510 IF C$="N" THEN LET AH=0 <222>
3520 PRINT <102>
3525 PRINT" <21>
" <66>
3530 PRINT"NACH(SPACE)ABZUG(SPACE)DES(SP
ACE)ALTERSFREIBETRAGS(SPACE)UND(SPACE)" <117>
3535 PRINT <96>
3540 LET EK=INT(N+M+ZI-AH-VS+VE) <186>
3545 PRINT"DES(SPACE)ALTERSENTLASTUNGSBE
TRAGS:";EK;"(SPACE)DM" <46>
3550 PRINT" <35>
" <142>
3555 PRINT"EINGABE(SPACE)DER(SPACE)KINDE
RFREIBETRAEGE" <200>
3560 PRINT <152>
3565 INPUT"ANZAHL(SPACE)DER(SPACE)ZU(SPA
CE)BERUECKSICHTIGENDEN(SPACE)KINDER";KZ <98>
3570 PRINT <241>
3575 LET KF=KZ*432 <168>
3580 PRINTTAB(10)"KINDERFREIBETRAG:(SPAC
E)";KF;"(SPACE)DM" <84>
3585 PRINT <215>
3587 PRINT" <189>
" <94>
3590 PRINT"EINKOMMEN(SPACE)NACH(SPACE)ABZ
UG(SPACE)DER(SPACE)KINDER" <82>
3595 PRINT"FREIBETRAEGE:(SPACE)";EK-KF;
"(SPACE)DM" <158>
3597 PRINT" <78>
" <193>
3599 GOTO 6000
3600 END
3605 REM "WEKO"
3610 PRINT
3615 PRINT "UNTERPROGRAMM(SPACE)WERBUNGS
KOSTEN" <54>
3620 LET R=0:LET U=0:LET Z=0 <23>
3625 PRINT <208>
3630 INPUT "WIEVIEL(SPACE)ARBEITSTAGE";T <116>
3635 INPUT "KILOMETER(SPACE)(EINFACH)";K <184>
3640 PRINT <223>
3645 LET R=INT(T*K*.36) <211>
3650 PRINT"FAHRTEN(SPACE)ZUR(SPACE)ARBEI
T(SPACE)I.H.V.";R;"DM" <155>
3655 PRINT <238>
3660 PRINT"SOLLEN(SPACE)DIE(SPACE)KOSTEN
(SPACE)FUER(SPACE)EIN(SPACE)ARBEITS-" <113>
3665 PRINT"ZIMMER(SPACE)BERECHNET(SPACE)
WERDEN?(SPACE)(J/N)" <132>
3670 PRINT <253>
3675 INPUT Z$ <14>
3680 IF Z$="(SPACE)" THEN GOTO 3680 <162>
3685 IF Z$="J" THEN GOSUB 3775 <7>
3690 IF Z$="N" THEN PRINT"GIBT(SPACE)ES(
SPACE)WEITERE(SPACE)WERBUNGSKOSTEN?(SPAC
E)(J/N)" <3>
3695 INPUT Y$ <32>
3700 IF Y$="(SPACE)" THEN GOTO 3700 <95>
3705 IF Y$="N" THEN GOTO 3735 <221>
3710 PRINT "BEI(SPACE)DIESEM(SPACE)EINFA
CHEN(SPACE)PROGRAMM(SPACE)MUESSEN"; <84>
3715 PRINT"(SPACE3)DIE(SPACE)WEITEREN(SP

```

```

ACE)WERBUNGSKOSTEN(SPACE)ALS(SPACE)SUMME
(SPACE)"; <159>
3720 PRINT"(SPACE2)EINGEGEBEN(SPACE)WERD
EN!" <38>
3725 PRINT <52>
3730 INPUT "WEITERE(SPACE)WERBUNGSKOSTEN
(SPACE)";U <142>
3735 LET W=INT(Z+U+R) <232>
3740 IF W<564 THEN LET W=564 <93>
3745 IF L<W THEN W=L <210>
3750 PRINT <77>
3755 PRINT"DIE(SPACE)ABSETZBAREN(SPACE)W
ERBUNGSKOSTEN(SPACE)"; <134>
3760 PRINT"BETRAGEN(SPACE)";W;"(SPACE)DM
" <18>
3765 RETURN <81>
3770 END <72>
3775 REM***UNTERPROGRAMM ZUR BERECHNUNG*
* <238>
3780 REM***DES ARBEITSZIMMERS***** <152>
* <78>
3785 INPUT"HOEHE(SPACE)DER(SPACE)MIETE:(
SPACE9)";H <151>
3790 INPUT"HOEHE(SPACE)DER(SPACE)NEBENKO
STEN(SPACE4)";N <143>
3795 INPUT"STROM(SPACE)UND(SPACE)WEITERE
(SPACE)KOSTEN(SPACE)";W <249>
3800 INPUT"QUADRATMETER(SPACE)WOHNFLAECHE
(SPACE)";QW <25>
3805 INPUT"QUADRATMETER(SPACE)ARBEITSZIM
M.";B <111>
3810 LET Z=INT(((H+N+W)/QW)*B*12) <142>
3815 PRINT <6>
3820 PRINT"DIE(SPACE)ABSCHREIBBAREN(SPAC
E)KOSTEN(SPACE)FUER(SPACE)DAS(SPACE5)" <137>
3825 PRINT"ARBEITSZIMMER(SPACE)BETRAGEN(
SPACE)";Z;"DM." <157>
3830 PRINT <179>
3835 PRINT"GIBT(SPACE)ES(SPACE)WEITERE(S
PACE)WERBUNGSKOSTEN?" <157>
3840 RETURN <148>
3845 END <178>
3850 PRINT <3>
3855 LET R=0:LET U=0:LET Z=0 <211>
3860 PRINT"WERBUNGSKOSTEN(SPACE)EHEFRAU:
" <96>
3865 INPUT "WIEVIEL(SPACE)ARBEITSTAGE";T <164>
3870 INPUT "KILOMETER(SPACE)(EINFACH)";K <203>
3875 PRINT <191>
3880 LET R=INT(T*K*.36) <135>
3885 PRINT"FAHRTEN(SPACE)ZUR(SPACE)ARBEI
T(SPACE)I.H.V.";R;"DM" <218>
3890 PRINT <240>
3895 PRINT"GIBT(SPACE)ES(SPACE)WEITERE(S
PACE)WERBUNGSKOSTEN?" <113>
3900 INPUT A$: IF A$="(SPACE)" THEN GOTO <63>
3905 IF A$="N" THEN GOTO 3920
3910 PRINT"WEITERE(SPACE)WERBUNGSKOSTEN(
SPACE)ALS(SPACE)SUMME(SPACE)" <148>
3915 PRINT"EINGEBEN!";INPUT U <252>
3920 LET WK=INT(R+U) <220>
3925 IF WK<564 THEN LET WK=564 <171>
3930 IF H<WK THEN LET WK=H <121>
3935 PRINT"WERBUNGSKOSTEN(SPACE)DER(SPAC
E)EHEFRAU:";WK <45>
3940 RETURN <1>
3945 END <248>
3950 REM***EINKUENFTE KAPITALVERMOEGEN**
<5>
3955 LET ZI=0:E=0:WB=0:F=0 <139>
3960 PRINT <32>
3965 PRINT"EINKUENFTE(SPACE)KAPITALVERMO
EGEN" <164>
3970 PRINT <42>

```



```

3975 INPUT "EINNAHMEN: {SPACE2}"; E <207>
3980 PRINT <52>
4015 INPUT "WERBUNGSKOSTEN: {SPACE3}"; WB <196>
4020 IF WB<Q*100 THEN LET WB=Q*100 <234>
4025 LET F=Q*300 <148>
4030 LET ZI=INT(E-WB-F) <185>
4035 IF ZI<0 THEN LET ZI=0 <175>
4040 RETURN <101>
4045 END <92>
4100 PRINT <173>
4105 REM***GOSUB FUER VERSORGUNGS-**** <64>
4106 REM***FREIBETRAG***** <101>
4110 PRINT "GEBEN {SPACE} SIE {SPACE} DIE {SPA <96>
CE} HOEHE {SPACE} DER {SPACE} VERSORGUNGS-"
4115 PRINT "BEZUEGE {SPACE} EIN {SPACE} (EHM <251>
ANN)."
4120 INPUT VZ <103>
4125 LET VB=INT(VZ*40/100) <79>
4130 IF VB>4800 THEN LET VB=4800 <106>
4135 PRINT <208>
4140 INPUT "VERSORGUNGSBEZUEGE {SPACE} (EHE <179>
FRAU)"; WV
4142 PRINT <215>
4145 LET VC=INT(WV*40/100) <77>
4150 IF VC>4800 THEN LET VC=4800 <141>
4155 LET VS=VB+VC <43>
4160 PRINT "VERSORGUNGSFREIBETRAG {SPACE} ( <205>
EHEMANN)"; VB
4162 PRINT <235>
4165 PRINT "VERSORGUNGSFREIBETRAG {SPACE} ( <120>
EHFRAU)"; VC
4170 PRINT <243>
4175 PRINT "GEMEINSAM: "; VS; " {SPACE} DM" <153>
4185 RETURN <247>
4190 END <238>
4200 REM***EINKUENFTE AUS VERMIETUNG*** <64>
4201 PRINT <18>
4205 PRINT "ACHTUNG: DIE {SPACE} BERECHNUNG { <186>
SPACE} ERFOLGT {SPACE} VORERST"
4207 PRINT "NOCH {SPACE} NACH {SPACE} DER {SPA <51>
CE} SEHR {SPACE} EINFACHEN {SPACE} GLEICHUNG:
"
4210 PRINT " {SPACE4} SUMME {SPACE} DER {SPACE <85>
3} EINNAHMEN"
4212 PRINT " {SPACE3} - SUMME {SPACE} DER {SPA <144>
CE} ABSCHREIBUNGEN"
4214 PRINT " {SPACE3} - SUMME {SPACE} DER {SPA <0>
CE} SONSTIGEN {SPACE} UNKOSTEN"
4215 PRINT " {SPACE3} " <82>
4217 PRINT " {SPACE4} EINKUENFTE {SPACE} AUS { <30>
SPACE} VERM. {SPACE} U. {SPACE} VERPACHTUNG"
4220 PRINT <37>
4225 INPUT "EINNAHMEN {SPACE} (AUCH {SPACE} E <11>
IGENNUTZUNG)"; EN
4227 PRINT <44>
4230 INPUT " {SPACE17} ABSCHREIBUNG"; AB <244>
4235 PRINT <52>
4240 INPUT " {SPACE9} SUMME {SPACE} ALLER {SPA <113>
CE} UNKOSTEN"; UK
4242 LET VE=INT(EN-AB-UK) <114>
4245 PRINT <62>
4250 PRINT "DIE {SPACE} EINKUENFTE {SPACE} AU <159>
S {SPACE} V+V {SPACE} BETRAGEN {SPACE}"; VE; "D
M"
4270 PRINT <87>
4275 RETURN <81>
4280 END <72>
4300 REM***ALTERSFREIBETRAEGE***** <123>
4301 REM***ALTERSENTLASTUNGSBETRAEGE*** <189>
4302 PRINT "ANZAHL {SPACE} DER {SPACE} ALTERS <184>
FREIBETRAEGE? {SPACE} (0/1/2)"
4305 INPUT "STICHTAG {SPACE} FUER {SPACE} 198 <215>
5: {SPACE} 1.1.1920 {SPACE}"; D

```

```

4310 LET AF=D*720 <167>
4315 PRINT <132>
4316 PRINT "ALTERSFREIBETRAG: {SPACE2}"; AF <221>
; " {SPACE} DM"
4320 PRINT <137>
4322 LET VE=EV; IF EV <0 THEN LET EV=0 <74>
4323 LET ZI=IZ; IF IZ <0 THEN LET IZ=0 <83>
4325 LET AS=INT((N+EV+IZ-VZ)*40/100) <92>
4326 LET AT=INT((M-WV)*40/100) <93>
4327 IF AT>3000 THEN LET AT=3000 <113>
4330 IF AS>3000 THEN LET AS=3000 <101>
4335 PRINT <152>
4340 PRINT "A-ENTLASTUNG {SPACE} EHEMANN: {S <55>
PACE8}"; AS; " {SPACE} DM"
4341 PRINT "A-ENTLASTUNG {SPACE} EHFRAU: {S <179>
PACE8}"; AT; " {SPACE} DM"
4342 PRINT <159>
4343 LET AH=AF+AS+AT <146>
4345 PRINT "A-FREI {SPACE} + {SPACE} A-ENTL. B <139>
ETRAG {SPACE}"; AH; " {SPACE} DM"
4350 RETURN <156>
4360 END <153>
6000 REM***BERECHNUNG DER SONDERAUSGABEN* <155>
6010 PRINT
6015 PRINT "BERECHNUNG {SPACE} DER {SPACE} SO <84>
NDERAUSGABEN"
6020 PRINT <52>
6022 PRINT "A) {SPACE} UNBESCHRAENKT {SPACE} <147>
ABZUGSFAEHIGE:"
6023 PRINT <55>
6025 INPUT "KIRCHENSTEUER: {SPACE2}"; KS <251>
6030 INPUT "ERSTATTUNG {SPACE2} KI-STEUER: { <58>
SPACE}"; KI
6035 INPUT "STEUERBERATUNGSKOSTEN: {SPACE} <201>
"; SB
6040 INPUT "ABSETZBARE {SPACE} SPENDEN: {SPA <228>
CE4}"; SP
6042 PRINT "HOECHSTBETRAEGE {SPACE} BEACHT <41>
E N!"
6045 INPUT "SONSTIGE {SPACE} ABZ. SO-AUSGAB: <37>
"; SS
6050 LET UB=INT(KS-KI+SB+SP+SS) <226>
6052 IF UB<Q*270 THEN LET UB=Q*270 <148>
6055 PRINT <87>
6060 PRINT "UNBESCHRAENKT {SPACE} ABZUGSFAE <170>
HIGE {SPACE} SONDERAUS-"
6063 PRINT "GABEN: {SPACE13}"; UB; " {SPACE} D <69>
M"
6070 PRINT <102>
6073 PRINT "B) {SPACE} BESCHRAENKT {SPACE} AB <243>
ZUGSFAEHIGE:"
6075 PRINT <107>
6100 PRINT "ARBEITNEHMERANTEIL {SPACE} GESE <21>
TL. SOZIALVERS."
6105 INPUT "EHEMANN: {SPACE15}"; SM <66>
6110 INPUT "EHFRAU: {SPACE15}"; SF <160>
6115 INPUT "FREIWILLIGE {SPACE} HOEHERVERS: <142>
"; FH
6120 INPUT "ARBEITGEBERZUSCHUSS: {SPACE3} " <242>
; AZ
6125 INPUT "KRANKENVERSICHERUNGEN: {SPACE} <129>
"; KV
6130 INPUT "EVTL. ERSTATTUNGEN/ZUSCH"; ES <106>
6135 INPUT "UNFALLVERSICHERUNGEN: {SPACE2} <155>
"; UV
6140 INPUT "LEBENSVERSICHERUNGEN: {SPACE2} <175>
"; LV
6150 INPUT "HAFTPFLICHTVERS. O. KASKO"; HP <37>
6160 LET BA=SM+SF+FH+KV+UV+LV+HP-AZ-ES <55>
6200 PRINT <233>
6205 INPUT "SIND {SPACE} SIE {SPACE} BEAMTER? <192>
{SPACE2} (J/N)"; D#
6210 IF D#=" {SPACE} " THEN GOTO 6210 <0>
6213 IF D#="N" THEN GOTO 6250 <71>

```

```

6220 PRINT "EINNAHMEN {SPACE} ALS {SPACE} BEAM <25>
TER (EHEMANN {SPACE} UND {SPACE} EHEFR. "
6222 PRINT <255>
6225 INPUT "EHEMANN: {SPACE15}"; BB <160>
6230 PRINT "EHFRAU: {SPACE15}"; CC <22>
6235 LET B1=INT(BB/100*9) <193>
6238 LET C1=INT(CC/100*9) <213>
6250 PRINT <27>
6255 PRINT "ARBEITGEBERANTEIL {SPACE} GESET <206>
ZL. {SPACE} SOZIALVERS."
6260 INPUT "EHEMANN: {SPACE15}"; AG <56>
6265 INPUT "EHFRAU: {SPACE15}"; EF <180>
6275 LET AA=AG+EF <99>
6276 IF AA>3000*Q THEN LET AA=3000*Q <144>
6280 LET ZZ=Q*3000-AA-B1-C1 <114>
6300 LET YV=BA-ZZ <56>
6305 IF YV<0 THEN LET YV=0 <76>
6310 PRINT <87>
6315 PRINT "BEITRAEGE {SPACE} AN {SPACE} BAUS <131>
PARKASSEN {SPACE} (OHNE {SPACE} VER-"
6318 INPUT "MOEGENSWIRKSAME {SPACE} LEISTUN <230>
GEN)"; BS
6320 LET ZX=YY+BS <72>
6325 LET HH=Q*2340+KZ*600 <242>
6330 LET FF=ZX-HH <199>
6335 IF FF<0 THEN LET FF=0 <200>
6340 LET DD=FF/100*50 <165>
6345 LET MM=HH/100*50 <237>
6400 IF BA>ZZ THEN LET S1=ZZ <163>
6405 IF ZZ>BA THEN LET S1=BA <176>
6410 IF ZX>HH THEN LET S2=HH <164>
6415 IF HH>ZX THEN LET S2=ZX <183>
6420 IF DD>MM THEN LET S3=MM <4>
6425 IF MM>DD THEN LET S3=DD <242>
6430 LET SA=INT(S1+S2+S3) <244>
6432 IF SA<Q*300 THEN LET SA=Q*300 <72>
6435 PRINT <213>
6440 PRINT "BESCHRAENKT {SPACE} ABZUGSFAEHI <173>
GE {SPACE} SONDERAUS-"
6445 PRINT "GABEN: {SPACE6}"; SA; " {SPACE} DM <213>
"
6500 LET SO=SA+UB <37>
6600 PRINT <122>
6605 INPUT "AUSSERGEWOEHNLICHE {SPACE} BELA <117>
STUNGEN"; FX
6607 PRINT <129>
6610 PRINT "WIRD {SPACE} EINHAUSHALTSFREIBE <103>
TRAG {SPACE} GELTEND {SPACE} GE-"
6615 INPUT "MACHT {SPACE7} (J/N)"; E# <214>
6617 PRINT " {SPACE3} ALLEINSTEHENDE {SPACE} M <83>
IT {SPACE} KIND {SPACE} BEI {SPACE} GRUNDTAB. ! <213>
{SPACEOFF} "
6620 IF E#=" {SPACE} " THEN GOTO 6620 <213>
6625 IF E#="N" THEN LET FZ=0:GOTO 9000 <244>
6630 IF E#="J" THEN LET FZ=4212 <237>
6888 PRINT <115>
9000 PRINT "CONT {SPACE} & {SPACE} RETURN {SPA <197>
CE} EINGEBEN!"; STOP
9003 SCNLRL <54>
9005 PRINT <233>
9010 PRINT "BERECHNUNG {SPACE} DES {SPACE} VE <2>
RSTEUERNEN {SPACE} EINKOMMENS"
9020 PRINT " " <33>
9030 PRINT " {SPACE11} EHEMANN: {SPACE8} EHEF <4>
RAU: "
9035 PRINT "BRUTTOLOHN: {SPACE}"; L; " {SPACE <170>
Z}"; H
9040 PRINT "WERBUNGSK.: {SPACE}"; W; " {SPACE <18>
Z}"; WK
9045 PRINT "VERMIETUNG: {SPACE}"; VE <222>
9050 PRINT "KAP.-VERM.: {SPACE}"; ZI <98>
9052 PRINT "SONDERAUSG: {SPACE}"; SO <19>
9055 PRINT "VERS.FREI.: {SPACE}"; VB; " {SPAC

```

```

E7}"; VC <173>
9060 PRINT "ALTERSFREI: {SPACE}"; AF <25>
9065 PRINT "ALTERSENTL: {SPACE}"; AS; " {SPAC <218>
E7}"; AT
9070 PRINT "KINDERFREI: {SPACE}"; KF <88>
9072 PRINT "AUSSERG.BE: {SPACE}"; FX <30>
9073 PRINT "HAUSHALTSFREIBETRAG: {SPACE3} " <9>
; FZ
9075 PRINT " " <217>
" <50>
9078 PRINT <51>
9080 LET ZE=N+M+ZI+VE-VS-KF-AH-SO-FX-FZ <82>
9090 IF ZE<0 THEN LET ZE=0
9880 PRINT "ZU {SPACE} VERSTUEHNDES {SPACE} <120>
EINKOMMEN: "; ZE; " {SPACE} DM"
9890 PRINT <97>
9899 PRINT " " <20>
9910 PRINT " {SPACE} 'CONT' {SPACE} & {SPACE} R <41>
ETURN {SPACE} EINGEBEN!"; STOP
9915 SCNLRL <201>
9920 PRINT <127>
9925 PRINT "BERECHNUNG {SPACE} DER {SPACE} ZU <90>
{SPACE} ERSTATTENDEN {SPACE} STEUER"
9927 PRINT " " <48>
9930 PRINT "BEZAHLTE {SPACE} STEUERN: " <106>
9935 INPUT "EHEMANN: {SPACE11}"; R1 <191>
9940 INPUT "EHFRAU: {SPACE11}"; R2 <1>
9945 INPUT "KI-STEUER (EHEMANN)"; R3 <230>
9950 INPUT "KI-STEUER (EHFRAU)"; R4 <80>
9955 PRINT <162>
9960 PRINT "STEUER {SPACE} NACH"; <18>
10900 IF Q=2 THEN PRINT " {SPACE2} SPLITTIN <176>
GTABELLE"
10910 IF Q=1 THEN PRINT " {SPACE2} GRUNDTAB <13>
ELLE"
10915 IF FZ>0 THEN PRINT " {SPACE2} GRUNDT <33>
ABELLE"
10920 PRINT <107>
10925 PRINT "TABELLE {SPACE} ODER {SPACE} FIN <121>
ANZAMT {SPACE} FRAGEN!"; INPUT "STEUER {SPACE
} NACH {SPACE} TABELLE: "; T5
10930 LET T6=INT(T5/100*8) <101>
10935 LET Z7=R1+R2-T5 <8>
10940 LET Z8=R3+R4-T6 <58>
10945 PRINT <132>
10950 PRINT "STEUER {SPACE} NACH {SPACE} TABE <216>
LLE: {SPACE4}"; T5
10955 PRINT "KIRCHENSTEUER {SPACE} (8%) {SPA <37>
CE}: {SPACE4}"; T6
10957 PRINT <144>
10960 PRINT "EINBEHALTENE {SPACE} STEUERN: { <99>
SPACE3}"; R1+R2
10962 PRINT "EINBEHALTENE {SPACE} KI-STEUER <86>
N: "; R3+R4
10970 PRINT " " <71>
10972 PRINT "ZU {SPACE} ERSTATTENDE {SPACE} S <19>
TEUERN: {SPACE}"; Z7; " {SPACE} DM"
10975 PRINT "ZU {SPACE} ERSTATTENDE {SPACE} K <109>
I-ST.: {SPACE2}"; Z8; " {SPACE} DM"
10980 PRINT " " <81>
11000 END <162>
12000 REM*** ALLE BETRAEGE WERDEN AUF * <177>
12001 REM*** VOLLE DM GERUNDET UM DIE * <44>
12002 REM*** UEBERSICHTLICHKEIT ZU * <214>
12003 REM*** WAHREN * <34>
12004 REM*** DIE HOECHSTBETRAEGE WUR- * <26>
12005 REM*** DEN BERUECKSICHTIGT * <246>
12010 REM*** DIE RESTLICHEN 1848 BITS * <70>
12011 REM*** WERDEN FUER DIE EINGE- * <127>
12012 REM*** GEBENEN DATEN BENDETIGT * <166>

```


Was gibt's Neues auf dem Software-Markt

Banditen der Lüfte:

Rasant & waagemutig!

Wir sind der Meinung, noch kein derartiges Spiel für den Commodore 16 vorher gesehen zu haben. MASTERTRONIC'S BANDITS AT ZERO ist ein rasantes, spannendes, grafisch voll ausgereiftes Programm für den C-16! Es erinnert stark an den einstigen Spielhallen-Hit „Defender“, hat jedoch einiges an „Schwungvollem“ auf den „kleinen Commodore“ übertragen. Sie sind allein über den Wolken. Nachdem Sie Ihren Bomber in der Luft aufgetankt haben, sind Sie nur noch auf sich gestellt. Der Feind schläft nicht. Er entsendet eine ganze Luftflotte, um Sie zu vernichten. Doch: Mit geschickten Ausweichmanövern und gezieltem Feuer werden Sie nicht so leicht zu bezwingen sein!

Die Grafik ist schon eine Augenweide. Das Scrolling sehr gut. Die Steuerung (Tastatur oder Joystick) optimal. Herrliche Landschaftsbilder (Wolken, Bergketten und das Meer) wechseln sich ab.

Wenn Sie alle feindlichen Flugzeuge „erwischt“ haben, beginnt das nächste Level: der Nachtflug und -kampf. An jedem neuen Tag, den Sie überleben, wird ein Tankflugzeug zu Ihnen geschickt, von dem Sie auftanken können. Insgesamt gesehen: ein hervorragendes Programm für den C-16/116.

Preis: 15 Mark, Hersteller: Mastertronic, Kaiser-Otto-Weg 18, 4770 Soest.



Mit einem tollen Programm in die Abenteuer der 20'er Jahre

Mugsy's Revenge für Schneider

Dieses MELBOURNE HOUSE-Programm versetzt Sie in die Unterwelt und die Geschäftspraktiken der Alkohol-Schieber der 20er Jahre. Das von uns getestete Programm verfügt über eine Grafik, die man getrost zum Allerfeinsten zählen kann. Der Sound steht den hervorragenden Bildern in keinsten Weise nach, und herrlich sind die dazu ertönenden Klänge mit Melodien aus den Goldenen Zwanziger Jahren. Die Handlung basiert in äußerst unterhaltsamer Weise auf strategischen Überle-

gungen beim Ankauf von Alkohol, bei der Eröffnung von Nacht-Clubs, dem Abschluß von „Verträgen“ mit anderen Gangsterbossen usw. Zudem werden alle Action-Fans voll auf ihre Kosten kommen. Eine tolle Arcade-Sequenz gibt genügend Möglichkeit, sich auszutoben. Wenn auch Sie ein wenig aus dem Unterwelt-Milieu erfahren möchten, dann sollte dieses Programm wegen seiner genannten Vorzüge nicht in Ihrer Programmsammlung fehlen.



Hersteller: Melbourne House
Vertrieb: T.S. Datensysteme, Denisstr. 45,
8500 Nürnberg

Mr. Puniverse

für die neuen Publikumsrenner C-16/116

Mastertronic stellt ein Programm für den C-16/116 vor, daß in unserer Programmierabteilung auf sehr positive Meinung gestoßen ist. Es handelt sich hierbei um ein Labyrinthspiel mit sehr gutem grafischem Aufbau. Das zu steuernde Männchen bewegt sich durch eine Vielzahl von Bildschirmseiten, in denen unterschiedliche Aufgaben wahrgenommen werden müssen.

Schnelligkeit und gutes Reaktionsvermögen sind bei diesem Spiel gefragt. Zusätzlich

muß man auch noch genau zum richtigen Zeitpunkt entscheiden, welchen Weg man geht, um das Ziel zu erreichen. Da das Programm bei den 10,- DM Produkten einzureihen ist, hat man für wenig Geld ein relativ gutes Programm für den C-16/116. Die Steuerung erfolgt entweder über die Tastatur oder mit dem Joystick.

Anbieter: Mastertronic GmbH
Kaiser-Otto-Weg 18
4770 Soest



Den Computer auch mal als sinnvolle Lernhilfe nutzen:

BRUCHRECHNEN leicht gemacht

Commodore 64/128 Diskette

Die Rechentafel - Bruchrechnen ist ein universelles Bruchrechnungsprogramm, geeignet für Schüler ab dem 6. Schuljahr, Jugendliche, Auszubildende und Erwachsene, die ihre Kenntnisse auffrischen oder die Bruchrechnung neu erlernen wollen.

Lerngegenstand: Vorführung und Einübung der mathematischen Operationen der Bruchrechnung: Addition, Subtraktion, Multiplikation, Division, Erweitern und Kürzen, unechte und gemischte Brüche und Kettenaufgaben.

Im Übungsteil können vom Computer gestellte Aufgaben am Bildschirm gelöst werden. Dabei gibt der Benutzer den Rechenweg (bei beliebig vielen Zwischenrechnungen) und das Ergebnis analog der Schreibweise auf dem Papier ein. Der Lösungsweg wird vom Programm nicht vor-

gegeben, sondern obliegt dem Benutzer. Bei falschen Eingaben (Zwischenrechnungen oder Ergebnis) erfolgt ein Hinweis darauf, daß ein Fehler gemacht wurde. Es besteht die Möglichkeit, den Fehler zu korrigieren und sich eine gezielte Hilfestellung (Helptaste) geben zu lassen. Auch besteht jederzeit die Möglichkeit, von der zu lösenden Aufgabe aus den Regelteil anzuwählen, ohne daß die zu lösende Aufgabe gelöscht oder verändert wird. Im Anschluß an die Vorführung der angewählten Regel kann die Aufgabe weitergerechnet werden. Bei Überschreiten der Anzahl der Lösungsversuche wird das Ergebnis der Aufgabe samt ausführlichem Lösungsweg gezeigt.

Des weiteren können eigene Aufgaben eingegeben werden. Eine Aufgabe kann eine Bildschirmzeile lang sein und beliebig viele (unterschiedliche) Rechenoperationen beinhalten (auch negative Zahlen).

Nach dem Überschreiten der (veränderbaren) Anzahl der Lösungsversuche wird das Ergebnis samt ausführlichem Rechenweg vorgeführt. Je nach Schwierigkeitsgrad kann der Lösungsweg mehrere Bildschirmseiten lang sein.

Es besteht die Möglichkeit, den Schwierigkeitsgrad der Aufgaben zu wählen. Bei den Schwierigkeitsgraden 1 und 2 treten in den Aufgaben keine gemischten Brüche auf, ab Schwierigkeitsgrad 3 können gemischte Brüche und auf Schwierigkeitsgrad 4 auch negative Zahlen auftreten. Außerdem werden mit zunehmendem Schwierigkeitsgrad die Zahlen für Nenner und Zähler größer.

Diskette 5 1/4" für Commodore 64/128, empfohlener Verkaufspreis 79,- DM.

© Unterrichtsmedien Hoppius,
Bannstr. 27, Wetzlar.

Karate-Kid

für VC-20 Grundversion

Die Karate-Schule sucht

ihren neuen Meister



Teil 1:

Karate, eine alte und weitverbreitete asiatische Kampfsportart, können Sie nun auch auf dem Commodore VC-20 erleben. Sie sind dabei in einer Karate-Schule, die ihren Meister sucht. Schauplatz des Kampfgeschehens ist eine dreidimensionale Arena in deren Mittelpunkt der Karateschüler zum Kampf bereit steht. Die Prüfung besteht darin, herumfliegende Ge-

genstände wie Steine, Flaschen und Kristalle zu zerstören. Für diese Aufgabe steht Ihnen ein Schlagrepertoire, das aus drei Varianten besteht, zur Verfügung. Sie können es mit Hilfe des Joysticks abrufen. Für jeden zerstörten Gegenstand erhalten Sie 50 Punkte gutgeschrieben, die auf einer Anzeigentafel zu sehen sind. Gelingt es Ihnen nicht, das betreffende Objekt beim ersten Versuch zu zerstören, oder sind Sie mit dem Gegenstand kollidiert, zerfallen Sie in tausend Stücke, und das Spiel wird automatisch beendet.

Lassen Sie den Kampfschrei „Viehhh“ ertönen und zerschmettern Sie alle Gegenstände.

```
1 PRINT "CLEAR"
2 POKE36869,255
3 POKE36878,15
10 POKE52,28:POKE56,28
15 CLR:FORR=0T0511:POKE7168+R,PEEK(32768+R):NEXT
20 FORI=1T0255:READA:POKE7167+I,A:NEXTI
30 REM***ARMSCHLAG***
40 DATA60,60,118,127,88,126,124,56
50 DATA62,66,91,91,216,223,207,225
60 DATA0,0,254,248,0,192,224,0
70 DATA255,255,127,127,126,126,231,199
80 DATA1,1,3,3,7,7,14,14
90 DATA199,195,131,7,15,15,28,28
100 DATA12,12,15,15,0,0,0,0
110 DATA24,24,30,30,0,0,0,0
120 DATA0,0,128,128,0,0,0,0
125 REM*GRUNDSTELLUNG*
130 DATA62,66,90,154,219,219,153,221
140 DATA205,101,117,113,126,126,231,199
150 REM**FUSSSCHLAG-U**
160 DATA62,64,91,153,191,191,128,255
170 DATA0,0,128,128,128,128,0,0
180 DATA127,127,127,115,49,57,24,28
190 DATA0,0,0,128,128,192,192,224
200 DATA24,56,48,112,224,192,192,224
210 DATA224,112,48,56,29,15,14,4
220 REM**FUSSSCHLAG-O**
230 DATA255,255,119,54,48,48,112,112
240 DATA96,96,96,224,224,192,192,224
250 DATA14,12,24,56,48,112,224,224
260 DATA12,128,0,0,0,0,0,0
265 DATA0,0,0,0,0,0,0,0
267 DATA63,67,133,249,249,138,252
270 DATA1,2,28,60,124,248,112,32
280 DATA0,42,84,110,221,110,186,108
290 DATA24,90,36,203,211,36,90,24
300 DATA247,132,132,132,244,20,20,247
```

```
<31>
<152>
<126>
<246>
<213>
<136>
<131>
<56>
<16>
<217>
<197>
<32>
<226>
<88>
<56>
<187>
<70>
<11>
<89>
<171>
<228>
<41>
<84>
<41>
<115>
<199>
<157>
<48>
<10>
<4>
<205>
<225>
<207>
<167>
<98>
<214>
<113>
310 DATA189,37,37,37,37,37,189
320 DATA239,40,40,46,232,136,72,47
330 DATA8,129,36,129,40,130,40,129
340 DATA17,128,36,0,72,0,34,0
350 DATA0,0,32,0,2,128,16,0
355 W=31
360 POKE36879,8
400 PRINT "DOWN2 RIGHT4 RVSON"
410 PRINT "RIGHT4 RVSON BLUE" | "WHITE" KAR
ATE-KID "BLUE" | "
420 PRINT "RIGHT4 RVSON" | "RVSON"
450 POKE36875,150
500 PRINT "DOWN3 RIGHT10 RVSON" | "DOWN LEFT"
510 FORL=1T0300:NEXTL:POKE36875,0
520 FORK=200T0360:POKE7680+K,32:NEXTK
525 POKE36875,150
530 PRINT "UP RIGHT8 RVSON" | "DOWN LEFT"
540 FORL=1T0300:NEXTL:POKE36875,0
550 FORK=200T0360:POKE7680+K,32:NEXTK
555 POKE36875,150
560 PRINT "UP7 RIGHT10 RVSON" | "DOWN LEFT"
570 FORL=1T0300:NEXTL:POKE36875,0
580 FORK=200T0360:POKE7680+K,32:NEXTK
585 POKE36875,150
590 PRINT "UP2 RIGHT8 RVSON" | "DOWN LEFT"
595 FORL=1T0300:NEXTL:POKE36875,0
600 FORK=200T0360:POKE7680+K,32:NEXTK
605 POKE36875,150
610 PRINT "UP1 RIGHT8 RVSON" | "DOWN LEFT"
615 FORL=1T0300:NEXTL:POKE36875,0
620 FORK=200T0360:POKE7680+K,32:NEXTK
625 POKE36875,150
630 PRINT "UP0 RIGHT8 RVSON" | "DOWN LEFT"
635 FORL=1T0300:NEXTL:POKE36875,0
640 FORK=200T0360:POKE7680+K,32:NEXTK
645 POKE36875,150
650 PRINT "UP-1 RIGHT8 RVSON" | "DOWN LEFT"
655 FORL=1T0300:NEXTL:POKE36875,0
660 FORK=200T0360:POKE7680+K,32:NEXTK
665 POKE36875,150
670 PRINT "UP-2 RIGHT8 RVSON" | "DOWN LEFT"
675 FORL=1T0300:NEXTL:POKE36875,0
680 FORK=200T0360:POKE7680+K,32:NEXTK
685 POKE36875,150
690 PRINT "UP-3 RIGHT8 RVSON" | "DOWN LEFT"
695 FORL=1T0300:NEXTL:POKE36875,0
700 FORK=200T0360:POKE7680+K,32:NEXTK
705 POKE36875,150
710 PRINT "UP-4 RIGHT8 RVSON" | "DOWN LEFT"
715 FORL=1T0300:NEXTL:POKE36875,0
720 FORK=200T0360:POKE7680+K,32:NEXTK
725 POKE36875,150
730 PRINT "UP-5 RIGHT8 RVSON" | "DOWN LEFT"
735 FORL=1T0300:NEXTL:POKE36875,0
740 FORK=200T0360:POKE7680+K,32:NEXTK
745 POKE36875,150
750 PRINT "UP-6 RIGHT8 RVSON" | "DOWN LEFT"
755 FORL=1T0300:NEXTL:POKE36875,0
760 FORK=200T0360:POKE7680+K,32:NEXTK
765 POKE36875,150
770 PRINT "UP-7 RIGHT8 RVSON" | "DOWN LEFT"
775 FORL=1T0300:NEXTL:POKE36875,0
780 FORK=200T0360:POKE7680+K,32:NEXTK
785 POKE36875,150
790 PRINT "UP-8 RIGHT8 RVSON" | "DOWN LEFT"
795 FORL=1T0300:NEXTL:POKE36875,0
800 FORK=200T0360:POKE7680+K,32:NEXTK
805 POKE36875,150
810 PRINT "UP-9 RIGHT8 RVSON" | "DOWN LEFT"
815 FORL=1T0300:NEXTL:POKE36875,0
820 FORK=200T0360:POKE7680+K,32:NEXTK
825 POKE36875,150
830 PRINT "UP-10 RIGHT8 RVSON" | "DOWN LEFT"
835 FORL=1T0300:NEXTL:POKE36875,0
840 FORK=200T0360:POKE7680+K,32:NEXTK
845 POKE36875,150
850 PRINT "UP-11 RIGHT8 RVSON" | "DOWN LEFT"
855 FORL=1T0300:NEXTL:POKE36875,0
860 FORK=200T0360:POKE7680+K,32:NEXTK
865 POKE36875,150
870 PRINT "UP-12 RIGHT8 RVSON" | "DOWN LEFT"
875 FORL=1T0300:NEXTL:POKE36875,0
880 FORK=200T0360:POKE7680+K,32:NEXTK
885 POKE36875,150
890 PRINT "UP-13 RIGHT8 RVSON" | "DOWN LEFT"
895 FORL=1T0300:NEXTL:POKE36875,0
900 FORK=200T0360:POKE7680+K,32:NEXTK
905 POKE36875,150
910 PRINT "UP-14 RIGHT8 RVSON" | "DOWN LEFT"
915 FORL=1T0300:NEXTL:POKE36875,0
920 FORK=200T0360:POKE7680+K,32:NEXTK
925 POKE36875,150
930 PRINT "UP-15 RIGHT8 RVSON" | "DOWN LEFT"
935 FORL=1T0300:NEXTL:POKE36875,0
940 FORK=200T0360:POKE7680+K,32:NEXTK
945 POKE36875,150
950 PRINT "UP-16 RIGHT8 RVSON" | "DOWN LEFT"
955 FORL=1T0300:NEXTL:POKE36875,0
960 FORK=200T0360:POKE7680+K,32:NEXTK
965 POKE36875,150
970 PRINT "UP-17 RIGHT8 RVSON" | "DOWN LEFT"
975 FORL=1T0300:NEXTL:POKE36875,0
980 FORK=200T0360:POKE7680+K,32:NEXTK
985 POKE36875,150
990 PRINT "UP-18 RIGHT8 RVSON" | "DOWN LEFT"
995 FORL=1T0300:NEXTL:POKE36875,0
1000 FORK=200T0360:POKE7680+K,32:NEXTK
1005 POKE36875,150
1010 PRINT "UP-19 RIGHT8 RVSON" | "DOWN LEFT"
1015 FORL=1T0300:NEXTL:POKE36875,0
1020 FORK=200T0360:POKE7680+K,32:NEXTK
1025 POKE36875,150
1030 PRINT "UP-20 RIGHT8 RVSON" | "DOWN LEFT"
1035 FORL=1T0300:NEXTL:POKE36875,0
1040 FORK=200T0360:POKE7680+K,32:NEXTK
1045 POKE36875,150
1050 PRINT "UP-21 RIGHT8 RVSON" | "DOWN LEFT"
1055 FORL=1T0300:NEXTL:POKE36875,0
1060 FORK=200T0360:POKE7680+K,32:NEXTK
1065 POKE36875,150
1070 PRINT "UP-22 RIGHT8 RVSON" | "DOWN LEFT"
1075 FORL=1T0300:NEXTL:POKE36875,0
1080 FORK=200T0360:POKE7680+K,32:NEXTK
1085 POKE36875,150
1090 PRINT "UP-23 RIGHT8 RVSON" | "DOWN LEFT"
1095 FORL=1T0300:NEXTL:POKE36875,0
1100 FORK=200T0360:POKE7680+K,32:NEXTK
1105 POKE36875,150
1110 PRINT "UP-24 RIGHT8 RVSON" | "DOWN LEFT"
1115 FORL=1T0300:NEXTL:POKE36875,0
1120 FORK=200T0360:POKE7680+K,32:NEXTK
1125 POKE36875,150
1130 PRINT "UP-25 RIGHT8 RVSON" | "DOWN LEFT"
1135 FORL=1T0300:NEXTL:POKE36875,0
1140 FORK=200T0360:POKE7680+K,32:NEXTK
1145 POKE36875,150
1150 PRINT "UP-26 RIGHT8 RVSON" | "DOWN LEFT"
1155 FORL=1T0300:NEXTL:POKE36875,0
1160 FORK=200T0360:POKE7680+K,32:NEXTK
1165 POKE36875,150
1170 PRINT "UP-27 RIGHT8 RVSON" | "DOWN LEFT"
1175 FORL=1T0300:NEXTL:POKE36875,0
1180 FORK=200T0360:POKE7680+K,32:NEXTK
1185 POKE36875,150
1190 PRINT "UP-28 RIGHT8 RVSON" | "DOWN LEFT"
1195 FORL=1T0300:NEXTL:POKE36875,0
1200 FORK=200T0360:POKE7680+K,32:NEXTK
1205 POKE36875,150
1210 PRINT "UP-29 RIGHT8 RVSON" | "DOWN LEFT"
1215 FORL=1T0300:NEXTL:POKE36875,0
1220 FORK=200T0360:POKE7680+K,32:NEXTK
1225 POKE36875,150
1230 PRINT "UP-30 RIGHT8 RVSON" | "DOWN LEFT"
1235 FORL=1T0300:NEXTL:POKE36875,0
1240 FORK=200T0360:POKE7680+K,32:NEXTK
1245 POKE36875,150
1250 PRINT "UP-31 RIGHT8 RVSON" | "DOWN LEFT"
1255 FORL=1T0300:NEXTL:POKE36875,0
1260 FORK=200T0360:POKE7680+K,32:NEXTK
1265 POKE36875,150
1270 PRINT "UP-32 RIGHT8 RVSON" | "DOWN LEFT"
1275 FORL=1T0300:NEXTL:POKE36875,0
1280 FORK=200T0360:POKE7680+K,32:NEXTK
1285 POKE36875,150
1290 PRINT "UP-33 RIGHT8 RVSON" | "DOWN LEFT"
1295 FORL=1T0300:NEXTL:POKE36875,0
1300 FORK=200T0360:POKE7680+K,32:NEXTK
1305 POKE36875,150
1310 PRINT "UP-34 RIGHT8 RVSON" | "DOWN LEFT"
1315 FORL=1T0300:NEXTL:POKE36875,0
1320 FORK=200T0360:POKE7680+K,32:NEXTK
1325 POKE36875,150
1330 PRINT "UP-35 RIGHT8 RVSON" | "DOWN LEFT"
1335 FORL=1T0300:NEXTL:POKE36875,0
1340 FORK=200T0360:POKE7680+K,32:NEXTK
1345 POKE36875,150
1350 PRINT "UP-36 RIGHT8 RVSON" | "DOWN LEFT"
1355 FORL=1T0300:NEXTL:POKE36875,0
1360 FORK=200T0360:POKE7680+K,32:NEXTK
1365 POKE36875,150
1370 PRINT "UP-37 RIGHT8 RVSON" | "DOWN LEFT"
1375 FORL=1T0300:NEXTL:POKE36875,0
1380 FORK=200T0360:POKE7680+K,32:NEXTK
1385 POKE36875,150
1390 PRINT "UP-38 RIGHT8 RVSON" | "DOWN LEFT"
1395 FORL=1T0300:NEXTL:POKE36875,0
1400 FORK=200T0360:POKE7680+K,32:NEXTK
1405 POKE36875,150
1410 PRINT "UP-39 RIGHT8 RVSON" | "DOWN LEFT"
1415 FORL=1T0300:NEXTL:POKE36875,0
1420 FORK=200T0360:POKE7680+K,32:NEXTK
1425 POKE36875,150
1430 PRINT "UP-40 RIGHT8 RVSON" | "DOWN LEFT"
1435 FORL=1T0300:NEXTL:POKE36875,0
1440 FORK=200T0360:POKE7680+K,32:NEXTK
1445 POKE36875,150
1450 PRINT "UP-41 RIGHT8 RVSON" | "DOWN LEFT"
1455 FORL=1T0300:NEXTL:POKE36875,0
1460 FORK=200T0360:POKE7680+K,32:NEXTK
1465 POKE36875,150
1470 PRINT "UP-42 RIGHT8 RVSON" | "DOWN LEFT"
1475 FORL=1T0300:NEXTL:POKE36875,0
1480 FORK=200T0360:POKE7680+K,32:NEXTK
1485 POKE36875,150
1490 PRINT "UP-43 RIGHT8 RVSON" | "DOWN LEFT"
1495 FORL=1T0300:NEXTL:POKE36875,0
1500 FORK=200T0360:POKE7680+K,32:NEXTK
1505 POKE36875,150
1510 PRINT "UP-44 RIGHT8 RVSON" | "DOWN LEFT"
1515 FORL=1T0300:NEXTL:POKE36875,0
1520 FORK=200T0360:POKE7680+K,32:NEXTK
1525 POKE36875,150
1530 PRINT "UP-45 RIGHT8 RVSON" | "DOWN LEFT"
1535 FORL=1T0300:NEXTL:POKE36875,0
1540 FORK=200T0360:POKE7680+K,32:NEXTK
1545 POKE36875,150
1550 PRINT "UP-46 RIGHT8 RVSON" | "DOWN LEFT"
1555 FORL=1T0300:NEXTL:POKE36875,0
1560 FORK=200T0360:POKE7680+K,32:NEXTK
1565 POKE36875,150
1570 PRINT "UP-47 RIGHT8 RVSON" | "DOWN LEFT"
1575 FORL=1T0300:NEXTL:POKE36875,0
1580 FORK=200T0360:POKE7680+K,32:NEXTK
1585 POKE36875,150
1590 PRINT "UP-48 RIGHT8 RVSON" | "DOWN LEFT"
1595 FORL=1T0300:NEXTL:POKE36875,0
1600 FORK=200T0360:POKE7680+K,32:NEXTK
1605 POKE36875,150
1610 PRINT "UP-49 RIGHT8 RVSON" | "DOWN LEFT"
1615 FORL=1T0300:NEXTL:POKE36875,0
1620 FORK=200T0360:POKE7680+K,32:NEXTK
1625 POKE36875,150
1630 PRINT "UP-50 RIGHT8 RVSON" | "DOWN LEFT"
1635 FORL=1T0300:NEXTL:POKE36875,0
1640 FORK=200T0360:POKE7680+K,32:NEXTK
1645 POKE36875,150
1650 PRINT "UP-51 RIGHT8 RVSON" | "DOWN LEFT"
1655 FORL=1T0300:NEXTL:POKE36875,0
1660 FORK=200T0360:POKE7680+K,32:NEXTK
1665 POKE36875,150
1670 PRINT "UP-52 RIGHT8 RVSON" | "DOWN LEFT"
1675 FORL=1T0300:NEXTL:POKE36875,0
1680 FORK=200T0360:POKE7680+K,32:NEXTK
1685 POKE36875,150
1690 PRINT "UP-53 RIGHT8 RVSON" | "DOWN LEFT"
1695 FORL=1T0300:NEXTL:POKE36875,0
1700 FORK=200T0360:POKE7680+K,32:NEXTK
1705 POKE36875,150
1710 PRINT "UP-54 RIGHT8 RVSON" | "DOWN LEFT"
1715 FORL=1T0300:NEXTL:POKE36875,0
1720 FORK=200T0360:POKE7680+K,32:NEXTK
1725 POKE36875,150
1730 PRINT "UP-55 RIGHT8 RVSON" | "DOWN LEFT"
1735 FORL=1T0300:NEXTL:POKE36875,0
1740 FORK=200T0360:POKE7680+K,32:NEXTK
1745 POKE36875,150
1750 PRINT "UP-56 RIGHT8 RVSON" | "DOWN LEFT"
1755 FORL=1T0300:NEXTL:POKE36875,0
1760 FORK=200T0360:POKE7680+K,32:NEXTK
1765 POKE36875,150
1770 PRINT "UP-57 RIGHT8 RVSON" | "DOWN LEFT"
1775 FORL=1T0300:NEXTL:POKE36875,0
1780 FORK=200T0360:POKE7680+K,32:NEXTK
1785 POKE36875,150
1790 PRINT "UP-58 RIGHT8 RVSON" | "DOWN LEFT"
1795 FORL=1T0300:NEXTL:POKE36875,0
1800 FORK=200T0360:POKE7680+K,32:NEXTK
1805 POKE36875,150
1810 PRINT "UP-59 RIGHT8 RVSON" | "DOWN LEFT"
1815 FORL=1T0300:NEXTL:POKE36875,0
1820 FORK=200T0360:POKE7680+K,32:NEXTK
1825 POKE36875,150
1830 PRINT "UP-60 RIGHT8 RVSON" | "DOWN LEFT"
1835 FORL=1T0300:NEXTL:POKE36875,0
1840 FORK=200T0360:POKE7680+K,32:NEXTK
1845 POKE36875,150
1850 PRINT "UP-61 RIGHT8 RVSON" | "DOWN LEFT"
1855 FORL=1T0300:NEXTL:POKE36875,0
1860 FORK=200T0360:POKE7680+K,32:NEXTK
1865 POKE36875,150
1870 PRINT "UP-62 RIGHT8 RVSON" | "DOWN LEFT"
1875 FORL=1T0300:NEXTL:POKE36875,0
1880 FORK=200T0360:POKE7680+K,32:NEXTK
1885 POKE36875,150
1890 PRINT "UP-63 RIGHT8 RVSON" | "DOWN LEFT"
1895 FORL=1T0300:NEXTL:POKE36875,0
1900 FORK=200T0360:POKE7680+K,32:NEXTK
1905 POKE36875,150
1910 PRINT "UP-64 RIGHT8 RVSON" | "DOWN LEFT"
1915 FORL=1T0300:NEXTL:POKE36875,0
1920 FORK=200T0360:POKE7680+K,32:NEXTK
1925 POKE36875,150
1930 PRINT "UP-65 RIGHT8 RVSON" | "DOWN LEFT"
1935 FORL=1T0300:NEXTL:POKE36875,0
1940 FORK=200T0360:POKE7680+K,32:NEXTK
1945 POKE36875,150
1950 PRINT "UP-66 RIGHT8 RVSON" | "DOWN LEFT"
1955 FORL=1T0300:NEXTL:POKE36875,0
1960 FORK=200T0360:POKE7680+K,32:NEXTK
1965 POKE36875,150
1970 PRINT "UP-67 RIGHT8 RVSON" | "DOWN LEFT"
1975 FORL=1T0300:NEXTL:POKE36875,0
1980 FORK=200T0360:POKE7680+K,32:NEXTK
1985 POKE36875,150
1990 PRINT "UP-68 RIGHT8 RVSON" | "DOWN LEFT"
1995 FORL=1T0300:NEXTL:POKE36875,0
2000 FORK=200T0360:POKE7680+K,32:NEXTK
2005 POKE36875,150
2010 PRINT "UP-69 RIGHT8 RVSON" | "DOWN LEFT"
2015 FORL=1T0300:NEXTL:POKE36875,0
2020 FORK=200T0360:POKE7680+K,32:NEXTK
2025 POKE36875,150
2030 PRINT "UP-70 RIGHT8 RVSON" | "DOWN LEFT"
2035 FORL=1T0300:NEXTL:POKE36875,0
2040 FORK=200T0360:POKE7680+K,32:NEXTK
2045 POKE36875,150
2050 PRINT "UP-71 RIGHT8 RVSON" | "DOWN LEFT"
2055 FORL=1T0300:NEXTL:POKE36875,0
2060 FORK=200T0360:POKE7680+K,32:NEXTK
2065 POKE36875,150
2070 PRINT "UP-72 RIGHT8 RVSON" | "DOWN LEFT"
2075 FORL=1T0300:NEXTL:POKE36875,0
2080 FORK=200T0360:POKE7680+K,32:NEXTK
2085 POKE36875,150
2090 PRINT "UP-73 RIGHT8 RVSON" | "DOWN LEFT"
2095 FORL=1T0300:NEXTL:POKE36875,0
2100 FORK=200T0360:POKE7680+K,32:NEXTK
2105 POKE36875,150
2110 PRINT "UP-74 RIGHT8 RVSON" | "DOWN LEFT"
2115 FORL=1T0300:NEXTL:POKE36875,0
2120 FORK=200T0360:POKE7680+K,32:NEXTK
2125 POKE36875,150
2130 PRINT "UP-75 RIGHT8 RVSON" | "DOWN LEFT"
2135 FORL=1T0300:NEXTL:POKE36875,0
2140 FORK=200T0360:POKE7680+K,32:NEXTK
2145 POKE36875,150
2150 PRINT "UP-76 RIGHT8 RVSON" | "DOWN LEFT"
2155 FORL=1T0300:NEXTL:POKE36875,0
2160 FORK=200T0360:POKE7680+K,32:NEXTK
2165 POKE36875,150
2170 PRINT "UP-77 RIGHT8 RVSON" | "DOWN LEFT"
2175 FORL=1T0300:NEXTL:POKE36875,0
2180 FORK=200T0360:POKE7680+K,32:NEXTK
2185 POKE36875,150
2190 PRINT "UP-78 RIGHT8 RVSON" | "DOWN LEFT"
2195 FORL=1T0300:NEXTL:POKE36875,0
2200 FORK=200T0360:POKE7680+K,32:NEXTK
2205 POKE36875,150
2210 PRINT "UP-79 RIGHT8 RVSON" | "DOWN LEFT"
2215 FORL=1T0300:NEXTL:POKE36875,0
2220 FORK=200T0360:POKE7680+K,32:NEXTK
2225 POKE36875,150
2230 PRINT "UP-80 RIGHT8 RVSON" | "DOWN LEFT"
2235 FORL=1T0300:NEXTL:POKE36875,0
2240 FORK=200T0360:POKE7680+K,32:NEXTK
2245 POKE36875,150
2250 PRINT "UP-81 RIGHT8 RVSON" | "DOWN LEFT"
2255 FORL=1T0300:NEXTL:POKE36875,0
2260 FORK=200T0360:POKE7680+K,32:NEXTK
2265 POKE36875,150
2270 PRINT "UP-82 RIGHT8 RVSON" | "DOWN LEFT"
2275 FORL=1T0300:NEXTL:POKE36875,0
2280 FORK=200T0360:POKE7680+K,32:NEXTK
2285 POKE36875,150
2290 PRINT "UP-83 RIGHT8 RVSON" | "DOWN LEFT"
2295 FORL=1T0300:NEXTL:POKE36875,0
2300 FORK=200T0360:POKE7680+K,32:NEXTK
2305 POKE36875,150
2310 PRINT "UP-84 RIGHT8 RVSON" | "DOWN LEFT"
2315 FORL=1T0300:NEXTL:POKE36875,0
2320 FORK=200T0360:POKE7680+K,32:NEXTK
2325 POKE36875,150
2330 PRINT "UP-85 RIGHT8 RVSON" | "DOWN LEFT"
2335 FORL=1T0300:NEXTL:POKE36875,0
2340 FORK=200T0360:POKE7680+K,32:NEXTK
2345 POKE36875,150
2350 PRINT "UP-86 RIGHT8 RVSON" | "DOWN LEFT"
2355 FORL=1T0300:NEXTL:POKE36875,0
2360 FORK=200T0360:POKE7680+K,32:NEXTK
2365 POKE36875,150
2370 PRINT "UP-87 RIGHT8 RVSON" | "DOWN LEFT"
2375 FORL=1T0300:NEXTL:POKE36875,0
2380 FORK=200T0360:POKE7680+K,32:NEXTK
2385 POKE36875,150
2390 PRINT "UP-88 RIGHT8 RVSON" | "DOWN LEFT"
2395 FORL=1T0300:NEXTL:POKE36875,0
2400 FORK=200T0360:POKE7680+K,32:NEXTK
2405 POKE36875,150
2410 PRINT "UP-89 RIGHT8 RVSON" | "DOWN LEFT"
2415 FORL=1T0300:NEXTL:POKE36875,0
2420 FORK=200T0360:POKE7680+K,32:NEXTK
2425 POKE36875,150
2430 PRINT "UP-90 RIGHT8 RVSON" | "DOWN LEFT"
2435 FORL=1T0300:NEXTL:POKE36875,0
2440 FORK=200T0360:POKE7680+K,32:NEXTK
2445 POKE36875,150
2450 PRINT "UP-91 RIGHT8 RVSON" | "DOWN LEFT"
2455 FORL=1T0300:NEXTL:POKE36875,0
2460 FORK=200T0360:POKE7680+K,32:NEXTK
2465 POKE36875,150
2470 PRINT "UP-92 RIGHT8 RVSON" | "DOWN LEFT"
2475 FORL=1T0300:NEXTL:POKE36875,0
2480 FORK=200T0360:POKE7680+K,32:NEXTK
2485 POKE36875,150
2490 PRINT "UP-93 RIGHT8 RVSON" | "DOWN LEFT"
2495 FORL=1T0300:NEXTL:POKE36875,0
2500 FORK=200T0360:POKE7680+K,32:NEXTK
2505 POKE36875,150
2510 PRINT "UP-94 RIGHT8 RVSON" | "DOWN LEFT"
2515 FORL=1T0300:NEXTL:POKE36875,0
2520 FORK=200T0360:POKE7680+K,32:NEXTK
2525 POKE36875,150
2530 PRINT "UP-95 RIGHT8 RVSON" | "DOWN LEFT"
2535 FORL=1T0300:NEXTL:POKE36875,0
2540 FORK=200T0360:POKE7680+K,32:NEXTK
2545 POKE36875,150
2550 PRINT "UP-96 RIGHT8 RVSON" | "DOWN LEFT"
2555 FORL=1T0300:NEXTL:POKE36875,0
2560 FORK=200T0360:POKE7680+K,32:NEXTK
2565 POKE36875,150
2570 PRINT "UP-97 RIGHT8 RVSON" | "DOWN LEFT"
2575 FORL=1T0300:NEXTL:POKE36875,0
2580 FORK=200T0360:POKE7680+K,32:NEXTK
2585 POKE36875,150
2590 PRINT "UP-98 RIGHT8 RVSON" | "DOWN LEFT"
2595 FORL=1T0300:NEXTL:POKE36875,0
2600 FORK=200T0360:POKE7680+K,32:NEXTK
2605 POKE36875,150
2610 PRINT "UP-99 RIGHT8 RVSON" | "DOWN LEFT"
2615 FORL=1T0300:NEXTL:POKE36875,0
2620 FORK=200T0360:POKE7680+K,32:NEXTK
2625 POKE36875,150
2630 PRINT "UP-100 RIGHT8 RVSON" | "DOWN LEFT"
2635 FORL=1T0300:NEXTL:POKE36875,0
2640 FORK=200T0360:POKE7680+K,32:NEXTK
2645 POKE36875,150
2650 PRINT "UP-101 RIGHT8 RVSON" | "DOWN LEFT"
2655 FORL=1T0300:NEXTL:POKE36875,0
2660 FORK=200T0360:POKE7680+K,32:NEXTK
2665 POKE36875,150
2670 PRINT "UP-102 RIGHT8 RVSON" | "DOWN LEFT"
2675 FORL=1T0300:NEXTL:POKE36875,0
2680 FORK=200T0360:POKE7680+K,32:NEXTK
2685 POKE36875,150
2690 PRINT "UP-103 RIGHT8 RVSON" | "DOWN LEFT"
2695 FORL=1T0300:NEXTL:POKE36875,0
2700 FORK=200T0360
```


Ein nützliches Anwenderprogramm für den VC-20 mit 3K-Erweiterung

DATA-MAKER

Will man Maschinenprogramme oder andere wichtige Speicherbereiche (zum Beispiel die Daten für einen neuen Zeichensatz) in Datazeilen ablegen, so ist dies von Hand sehr mühsam und extrem zeitaufwendig. Da liegt es doch nahe, unseren Computer für diese Aufgabe einzuspannen. Er müßte sich halt selbst programmieren. Dies ist mit der vorliegenden Routine auf sehr komfortable Art möglich und wird in durchaus akzeptabler Zeit erledigt, da das Programm vollständig in Maschinensprache geschrieben ist. Dennoch kann das Programm nicht zaubern und so muß man auf die Verarbeitung von z. B. 1000 Daten ca. 15-20 Sekunden warten. Um nun in den Genuß dieser Routine zu kommen, gehen Sie wie folgt vor:

- Geben Sie den abgedruckten Basic-Loader ein und speichern ihn zur Sicherheit ab. Die REM-Zeilen können dabei weggelassen werden, da sie für den Programmablauf völlig irrelevant sind. Im Listing tragen sie sehr zur Übersichtlichkeit des Programms bei, in eingegebener Form verwirren sie eher.

Da das Programm auf einem VC20 mit 16 kByte-Erweiterung entstand, ist es möglich, daß Anwender mit kleinerer oder gar keiner Ausbaustufe die REM-Anweisungen sogar weglassen müssen. Unabhängig davon ist die eigentliche Maschinenroutine, die relativ kurz ist, in jeder Ausbaustufe lauffähig.

- Nach dem Start mit RUN meldet das Programm (etwas Geduld bitte, die Datas müssen vorher eingelesen werden) Ihnen entweder einen Prüfsummenfehler in den Datas, Sie müssen dann die entsprechenden Zeilen noch einmal überprüfen, oder es erscheint die Abfrage „Abspeichern (J/N)“. Kassettenanwender müssen hier leider Nein(N) eingeben. Da die Routine im

Kassettenpuffer liegt, ist es ohne weiteres nicht möglich, sie auf Kassette abzuspeichern. Kassettenanwender müssen also jedesmal, wenn sie mit dem Utility arbeiten wollen, den Basic-Loader einladen und starten. Auch jede Kassettenoperation zerstört das Programm; es muß danach neu eingeladen werden. Bequemer haben es da die Diskettenanwender. Diskettenoperationen greifen nicht auf den Kassettenpuffer zu, folglich wird das Programm nicht beeinflusst und kann sogar als reines Maschinenprogramm auf Diskette abgespeichert werden. In der Form belegt es nur einen Block auf Diskette, und das lästige Einlesen der Datas entfällt dann bei späterem Gebrauch. Geben Sie dazu einfach Ja(J) ein, nachdem Sie eine Diskette eingelegt haben. Die Routine wird dann automatisch unter dem Namen "DATAGEN.CODE" abgespeichert. Eingeladen wird die Routine dann mit "LOAD"DATAGEN.CODE",8,1", wobei Sie das "8,1" am Ende nicht vergessen dürfen, da Sie sonst an eine falsche Stelle geladen wird. Nach dem Laden muß unbedingt ein NEW eingegeben werden, um alle Basic-Pointer wieder zu normalisieren.

- Haben Sie die Routine nun ordnungsgemäß im Speicher, wird sie mit folgender Syntax aufgerufen:

SYS 828,AW,EW,SZ

AW, EW, SZ sind dabei jeweils Zahlenwerte aus dem Bereich von 0 bis 65535 und haben folgende Bedeutung:

AW gibt den Anfangswert an, mit dem die Routine beginnen soll. Später also der erste DATA-Wert.

EW ist der entsprechende Endwert. Also später der letzte DATA-Wert.

SZ gibt die Startzeile an, also die Zeilennummer, mit der die erste Datazeile versehen wird. Passen nicht alle Datas in eine Zeile, wird SZ von der Rou-

tine automatisch um 1 erhöht und dieser Zahlenwert als nächste Zeilennummer benutzt. Computerspezifisch sind nur Zeilennummern bis 63999 erlaubt. Wird diese Höchstgrenze überschritten, gibt die Routine einen "ILLEGAL QUANTITY ERROR" aus, und der Programmablauf wird unterbrochen.

Am Ende liegt dann ein Basicprogramm vor, das nur aus DATA-Zeilen besteht und das Sie ganz normal abspeichern und an andere Programme anhängen können.

Programmautoren gesucht!

Für unsere Rubrik „WERKSTATT“ suchen wir weitere interessante und hilfreiche Anwenderprogramme. Das Honorar beträgt 120,- DM pro veröffentlichte Seite.

Die Redaktion

Für Interessierte habe ich in das Basicprogramm ein kommentiertes Assemblerlisting integriert. Die Idee besteht einfach darin, alle Werte auszulesen, die Hex-Zahlen in die ASCII-Darstellung zu wandeln, in den Eingabepuffer zu schreiben und schließlich in die Systemroutine zu verzweigen, die den Eingabepuffer auswertet. Für C64-Anwender dürfte es mit Hilfe einer REFERENZ-LISTE C-64-VC-20 kein Problem sein, die Routine auf dem C-64 lauffähig zu machen.

```

0 REM *****
1 REM *
2 REM *      DATA - MAKER      FUER DEN COMMODORE VC20
3 REM *
4 REM *      BELEGT WIRD DER SPEICHERBEREICH VOM 828 BIS 999
5 REM *      SYNTAX:  SYS828,AW,EW,SZ
6 REM *      ( AW-ANFANGSWERT / EW-ENDWERT / SZ-STARTZEILE )
7 REM *
8 REM *      WRITTEN 1985 BY  W A L D E M A R R A A Z
9 REM *****
10 :
11 :
10000 DATA 20,DA,03 :REM JSR $03DA ; STARTWERT HOLEN
10002 DATA 8D,7B,03 :REM STA $037B ; UND ZWISCHENSPEICHERN
10004 DATA 8C,7C,03 :REM STY $037C ;
10006 DATA 20,DA,03 :REM JSR $03DA ; ENDWERT HOLEN
10008 DATA 85,FB :REM STA $FB ; UND ZWISCHENSPEICHERN
10010 DATA 84,FC :REM STY $FC ;
10012 DATA 20,DA,03 :REM JSR $03DA ; STARTZEILE HOLEN
10014 DATA A0,00 :REM LDY #$00 ;
10016 DATA A9,83 :REM LDA #$83 ; TOKEN FUER 'DATA' AN DEN BEGINN
10018 DATA 99,00,02 :REM STA $0200,Y ; DES EINGABEPUFFERS SCHREIBEN
10020 DATA AD,7C,03 :REM LDA $037C ; ENDWERT SCHON ERREICHT ??
10022 DATA C5,FC :REM CMP $FC ;
10024 DATA 90,10 :REM BCC $036D ; NEIN, DANN WEITERMACHEN
10026 DATA D0,07 :REM BNE $0366 ;
10028 DATA A5,FB :REM LDA $FB ;
10030 DATA CD,7B,03 :REM CMP $037B ;
10032 DATA B0,07 :REM BCS $036D ;
10034 DATA C0,00 :REM CPY #$00 ; JA, TEST AUF ZEILE IM EINGABEPUFFER
10036 DATA D0,3F :REM BNE $03A9 ; POSITIV, DANN ZEILE NOCH EINBINDEN
10038 DATA 4C,74,C4 :REM JMP $C474 ; NEGATIV, DANN IN DEN READY-MODUS
10040 DATA A5,15 :REM LDA $15 ; ZEILENNUMMER NOCH IM ERLAUTBEN
10042 DATA C9,FA :REM CMP #$FA ; BEREICH ??
10044 DATA 90,03 :REM BCC $0376 ;
10046 DATA 4C,48,D2 :REM JMP $D248 ; NEIN, DANN 'ILLEGAL QUANTITY ERROR'
10048 DATA 84,FD :REM STY $FD ;
10050 DATA A9,00 :REM LDA #$00 ;
10052 DATA AE,00,00 :REM LDX $0000 ; AKTUELLES BYTE HOLEN
10054 DATA 85,62 :REM STA $62 ; UND ZWISCHENSPEICHERN
10056 DATA 86,63 :REM STX $63 ;
10058 DATA A2,90 :REM LDX #$90 ;
10060 DATA 38 :REM SEC ;
10062 DATA 20,49,DC :REM JSR $DC49 ; --> INTEGER IN DEN FAC SCHREIBEN
10064 DATA 20,DF,DD :REM JSR $DDDF ; --> FAC NACH ASCII WANDELN
10066 DATA A4,FD :REM LDY $FD ;
10068 DATA A2,00 :REM LDX #$00 ;
10070 DATA EB :REM INX ;
10072 DATA CB :REM INY ;
10074 DATA BD,FF,00 :REM LDA $00FF,X ;
10076 DATA 99,00,02 :REM STA $0200,Y ;
10078 DATA D0,F6 :REM BNE $03BE ;
10080 DATA A9,2C :REM LDA #$2C ; MIT KOMMA ABSCHLIESSEN
10082 DATA 99,00,02 :REM STA $0200,Y ;
10084 DATA EE,7B,03 :REM INC $037B ; AKTUELLE BYTEADRESSE ERHOEHEN
10086 DATA D0,03 :REM BNE $03A5 ;
10088 DATA EE,7C,03 :REM INC $037C ;
10090 DATA C0,41 :REM CPY #$41 ;
10092 DATA 90,AD :REM BCC $0356 ; EINGABEPUFFER SCHON VOLL ??
10094 DATA A9,00 :REM LDA #$00 ; NEIN, DANN NAECHSTES BYTE AUSWERTEN
; JA, DANN EINGABEPUFFER MIT

```

Fortsetzung S. 34

Zeitschriften und Programme von 1985

Heft 5/85		Heft 6/85		Heft 7/85		Heft 8/85	
Commodore		Commodore		Commodore		Commodore	
Bestell-Nr. COM CK 1/1	16,- DM	Bestell-Nr. COM CK 2/1	16,- DM	Bestell-Nr. COM CK 3/1	16,- DM	Bestell-Nr. COM CK 4/1	16,- DM
Bestell-Nr. COM CD 1/1	20,- DM	Bestell-Nr. COM CD 2/1	20,- DM	Bestell-Nr. COM CD 3/1	20,- DM	Bestell-Nr. COM CD 4/1	20,- DM
Sense (C-16/116), Jungle (C-64), Auto-start (C-64), Sequenzer (C-64), Sufo (C-64), Olympia (VC-20), Flowers (VC-20)		Mampman (C-64), Softlanding (C-64), Karate (C-64), Einbrecher (VC-20), Tanker (VC-20), Jupiter Rescue (VC-20), SOS-Schiffbruch (C-16)		Sea-Attack (C-64), Saco-Age (C-64), Micro-Tools (C-64), Disky 1.0 (C-64), Stuntman (VC-20), Sepp (VC-20), Balloon-Shoot (VC-20), Farbenfroh (C-16)		Drucker-List (VC-20), Flowers II (VC-20), Dino Eggs (VC-20), Piano (C-16/116), Tank-Duell (C-64), Vampir (C-64), Memory (C-64), Raster Register, Interrupt (C-64)	
Schneider		Schneider		Schneider		Schneider	
Bestell-Nr. COM SK 1/1	16,- DM	Bestell-Nr. COM SK 2/1	16,- DM	Bestell-Nr. COM SK 3/1	16,- DM	Bestell-Nr. COM SK 4/1	16,- DM
				Bestell-Nr. COM SD 3/1	39,- DM	Bestell-Nr. COM SD 4/1	39,- DM
Cobra, Krüml, Haunted Castle, Q-Bert, ROM-CALLS		Tastaturbelegung, Cave-Runner, Old-Shurehand		Secret Valley, Smileyman, Super-File		Copter Patrol, Suchrätsel, Tower	
Heft 9/85		Heft 10/85		Heft 11/85		Heft 12/85	
Commodore		Commodore		Commodore		Commodore	
Bestell-Nr. COM CK 5/1	16,- DM	Bestell-Nr. COM CK 6/1	16,- DM	Bestell-Nr. COM CK 7/1	16,- DM	Bestell-Nr. COM CK 4/1	16,- DM
Bestell-Nr. COM CD 5/1	20,- DM	Bestell-Nr. COM CD 6/1	20,- DM	Bestell-Nr. COM CD 7/1	20,- DM	Bestell-Nr. COM CD 4/1	20,- DM
Supermoni 64 (C-64), Galaxy (C-64), Atlantis (C-64), Shaolin (C-16), Hospital (VC-20), Flipper-Ball (VC-20), Alien (VC-20)		The Castle of Dr. Croop (VC-20), Catchgo (VC-20), Sub-Zone (VC-20), The Maze Graze Game (C-64), Super-Breakout (C-64), Planet of Death (C-64), Funktionstasten-Programmierung (C-64), Car Race (C-16/116), Key-Helper (C-16/116)		Interpreter Music-Language (C-64), Firelegion (C-64), Funktionstastenprogrammierung (C-64), Schatztaucher (C-16), Dangerous Planet (VC-20), Geo-Zaxxon (VC-20)		Froggies Nightmare (VC-20), Wächter des heiligen Tempels (VC-20), Jumping Joe (C-16/116), City-Runner (C-16/116), Mission Rescue (C-64), Turbo-Jet (C-64), Print-Generator (C-64), Datagenerator (C-64)	
Schneider		Schneider		Schneider		Schneider	
Bestell-Nr. COM SK 5/1	16,- DM	Bestell-Nr. COM SK 6/1	16,- DM	Bestell-Nr. COM SK 7/1	16,- DM	Bestell-Nr. COM SK 8/1	16,- DM
Bestell-Nr. COM SD 5/1	39,- DM	Bestell-Nr. COM SD 6/1	39,- DM	Bestell-Nr. COM SD 7/1	39,- DM	Bestell-Nr. COM SD 8/1	39,- DM
Duell, Crash, Citylander, Klondike, REM-Killer		Labyrinth, Catch, Mini Car Race, Inka-Sogra, Single Disk Backup		Etis, Allen-Hunter, Demon-Attack, Datenverwaltung, Discshow		Checksummer 1.0 CPC, Two Cherry Chaps, Erwin im Zauberschloß, Drucksteuerung NLQ 401	

Alle hier aufgeführten Kassetten und Disketten können jederzeit nachbestellt werden. Bei den Zeitschriften sind die Ausgaben 6/85, 7/85 und 2/86 leider vergriffen und können nicht mehr nachgeliefert werden. Bitte beachten Sie dies bei Ihrer Bestellung.

Leserservice: Programme von 1986

Heft 1/86		Heft 2/86		Heft 3/86	
Commodore		Commodore		Commodore	
Bestell-Nr. COM CK 1/2	16,- DM	Bestell-Nr. COM CK 2/2	16,- DM	Bestell-Nr. COM CK 3/2	16,- DM
Bestell-Nr. COM CD 1/2	20,- DM	Bestell-Nr. COM CD 2/2	20,- DM	Bestell-Nr. COM CD 3/2	20,- DM
Checksummer 1.0 C, Astronaut Battle (C-64), Hueycobra (C-16/116), Hardcopyroutine (C-16/116), Helmut und die Zeitmaschine (C-16/116), Locos (VC-20), Rockme (VC-20), Charly der Raketenmann (VC-20), Interrupt-Programmierung (VC-20)		Robo (C-64), Basic-Befehlsweiterung (C-64), Raketenstart (C-16/116), Dreikampf (C-16/116), Headline (C-16/116), Defender (VC-20), Formel 1 (VC-20), Memory-Display (VC-20)		Minor-Jumper (C-16/116), Hero (C-16/116), Data-Maker (C-16/116), Befehlsweiterung (C-64), Starchip (C-64), Mini-Car-Race (C-64), Text-Scroller (C-64), Text-Editor (C-64), Checksummer (VC-20), Robin at Ravenwreck (VC-20), Mini-Karate (VC-20), Autonumber-Routine (VC-20)	
Schneider		Schneider		Schneider	
Bestell-Nr. COM SK 1/2	16,- DM	Bestell-Nr. COM SK 2/2	16,- DM	Bestell-Nr. COM SK 3/2	16,- DM
Bestell-Nr. COM SD 1/2	39,- DM	Bestell-Nr. COM SD 2/2	39,- DM	Bestell-Nr. COM SD 3/2	39,- DM
Defender, Boulder Dash, Basic-Befehle auf den Funktionstasten		Allen 2001, Diamond Hunter, Directory		Smily, Screpi, Die verrückte Fabrik, Disc-Help	
Heft 4/86		Heft 5/86		Aus diesem Heft	
Commodore		Commodore		Commodore	
Bestell-Nr. COM CK 4/2	16,- DM	Bestell-Nr. COM CK 5/2	16,- DM	Bestell-Nr. COM CK 6/2	16,- DM
Bestell-Nr. COM CD 4/2	20,- DM	Bestell-Nr. COM CD 5/2	20,- DM	Bestell-Nr. COM CD 6/2	20,- DM
Shuttle (VC-20), Josef in den Höhlen (VC-20), Mergo-Routine (VC-20), Shoot It (VC-20), Fast Trust (C-64), Data-Text 64 (C-64), Gary, der schnelle Kabelverleger (C-16/116), SDI (C-16/116), Beep-Routine (C-16/116)		Space-Devil (C-64), Disk-Eddi (C-64), ABC-Legen (V-20), 3-D-Labyrinth (VC-20), Tron (VC-20), Invaders of Space (C-16/116), Submarine (C-16/116), Balkendiagramm (C-16/116)			
Schneider		Schneider		Schneider	
Bestell-Nr. COM SK 4/2	16,- DM	Bestell-Nr. COM SK 5/1	16,- DM	Bestell-Nr. COM SK 6/2	16,- DM
Bestell-Nr. COM SD 4/2	39,- DM	Bestell-Nr. COM SD 5/1	39,- DM	Bestell-Nr. COM SD 6/2	39,- DM
Space-Rescue, Texteditor, Starflight, Programm-Rettter		Startank, Grafik-Designer, Cross-Reference, Itext			

Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.	Programm	Preis/ Kassette	Preis/ Diskette	Bestell-Nr.
C=64				VIC-20			
Multi-Key/S-Tool 64/ Interrupt-Programme Spritehilfe/Diskloader/ Directory	16,- DM	21,- DM	UC 2/85	Bestellschein/Glücky	8,50 DM	15,- DM	V 61
Tape-Directory/Asmon/ Data-Generator	21,- DM	-	UC 3.1/85	Multigraph/All Rammer	11,- DM	15,50 DM	V 71
Fast Load	15,- DM	21,- DM	UC 3.2/85	Zyklo/Meteorit	11,- DM	19,50 DM	V 81
Diskmonitor/Zeichensatz	10,- DM	-	UC 4/84	Garten/Schloß Gruselstein	14,- DM	19,50 DM	V 91
Reassembler/Maskengenerator	20,- DM	-	UC 5/85	Frossman/Outlaw	14,- DM	19,50 DM	V 101
Del 64/Treiberprogramm für 1526/MPS 802 Decelerator	15,- DM	20,- DM	UC 6/85	Prost/Bufalo Bill	14,- DM	19,50 DM	V 121
Supertapedirectory/Renew/ Kontrollabfrage	14,- DM	19,- DM	UC 7/85	Joy Man/Powerpack	14,- DM	19,50 DM	V 22
Video-Utilities	15,- DM	-	UC 8/85	Der rasende Malocher/ Frankie goes to Pharoa	14,- DM	19,50 DM	V 32
Multi-Basic V3, Bildeditor	17,- DM	23,- DM	UC 9/85	Matron/Obst	14,- DM	19,50 DM	V 42
Better Basic/Autonumber/ Hardcopy/Terminalprogramm	17,- DM	23,- DM	UC 10/85	Race On/Cagy	14,- DM	19,50 DM	V 52
Speicher-Oszilloskop/ Makro-Assembler/Change Type	49,- DM	54,- DM	UC 1/86	Nürnberg 3D/Dösl	14,- DM	19,50 DM	V 62
Reactor/Concentration/Datenbank	17,50 DM	23,50 DM	HC/C-1	Breaker/Expulsion	14,- DM	19,50 DM	V 72
Warlords/Caverns of Death	14,50 DM	19,50 DM	HC/C-2	Brückenbau/Jango	16,- DM	21,50 DM	V 13
Alien-Destroyer/Duell	24,50 DM	29,50 DM	HC/C-3	Inka-Schatz/ Minor der Fassadenstreicher	16,- DM	21,50 DM	V 23
Ocean Game/Tennis	17,50 DM	23,50 DM	HC/C-4	Booby in Action/Bergshooting	16,- DM	21,50 DM	V 33
The Caves	17,50 DM	23,50 DM	HC/C-5	Zyklo/Geldsammel	11,50 DM	-	HC/V-1
Gardener	17,50 DM	23,50 DM	HC/C-6	Star-Wars/Punktfresser	14,50 DM	-	HC/V-2
Ufo/Skateboard Sam	24,50 DM	29,50 DM	HC/C-1-4	Catch the Fish/Mister Jump	14,50 DM	-	HC/V-3
Olympic Biathlon/ Struggle for Life	24,50 DM	29,50 DM	HC/C-2-4	Ghost's Eggs/Fox Hunt	14,50 DM	-	HC/V-4
Pole Position/Save	24,50 DM	29,50 DM	HC/C-3-4	Collector/Break Out	14,50 DM	-	HC/V-5
your teeth/Push	24,50 DM	29,50 DM	HC/C-4-4	Q-Bert/Salvage Crew	16,50 DM	-	HC/V-6
Explorer 2/Programmname?	8,- DM	15,- DM	C 41	Dog Fight/Schwitzbad	16,50 DM	-	HC/V-1-4
Mauern/Widerstand	15,- DM	23,50 DM	C 51	Earth Rescue	14,50 DM	-	HC/V-2-4
Space-Comets/Erdspalte/ Sprite-Data	16,50 DM	23,50 DM	C 61	Tank	14,50 DM	-	HC/V-3-4
Roadpainter	15,50 DM	19,50 DM	C 71	Terra 2/Pac-Ball	16,50 DM	21,50 DM	HC/V-4-4
Hardcopy/Space-Fighter/ Data-Generator	16,50 DM	23,50 DM	C 81	Sprites/Space-Battle	13,- DM	19,- DM	UV 2/85
Monster-Attack/Block- Painter/Epson-Drucker	16,50 DM	23,50 DM	C 91	Grafik-Printer	10,- DM	-	UV 5/85
Projekt/Datenbank	16,50 DM	23,50 DM	C 101	VIC-Clock	9,- DM	-	UV 6/85
Spiders/The Basic	17,50 DM	23,50 DM	C 121	Decelerator	9,- DM	-	UV 7/85
High Noon/Skeet/ Grafik-Designer	17,50 DM	23,50 DM	C 22	Joypaint	12,- DM	-	UV 8/85
Painter/Star-Baddle/Editor	19,50 DM	24,50 DM	C 32	Disassembler, Old-Programme	10,- DM	-	9/85
Wüstenrally/Jet-Pac/ Black Moore Castle	19,50 DM	24,50 DM	C 42	Directory/Farb-JRQ	10,- DM	-	UV 10/85
Brieftaube/Cadelon	19,50 DM	24,50 DM	C 52				
Ritter Erik/Grand Prix Spritehilfe	24,50 DM	29,50 DM	C 62				
Firebird/High Music	24,50 DM	29,50 DM	C 72				
Moonsweeper/Scotti	24,50 DM	29,50 DM	C 13				
Tron/Mercurious	24,50 DM	29,50 DM	C 23				
Interceptor-Basel/Schotter	24,50 DM	29,50 DM	C 33				
Jump-Man/Space-Taxi/The Way	24,50 DM	29,50 DM					
Neu in unserer Software-Bestell- liste der				Schneider			
C=16				CPC 464			
Fantasy-Country/ Horror-Castle	18,- DM	23,- DM	0 23	Autorennen	11,- DM	-	US 3/85
Moon Fighter/Goldrausch	18,- DM	23,- DM	0 33	Universal-Datei V1	9,- DM	-	US 4/85
Zeichendesigner	14,- DM	-	UO 8/85	Computerschrift/Symbol-Swap	10,- DM	-	US 5/85
OLD-Routine-Merge-Routine/ Change Type	14,-	19,-	UO 1/86	Keyboard Toolkit/Farbdecodierung	11,- DM	-	US 6.1/85
				Aladin	9,- DM	-	US 6.2/85
				Painter/Box-Befehl	14,- DM	24,- DM	US 7/85
				Maschinensprache-Monitor/ Disk-Hilfe	15,- DM	25,- DM	US 8/85
				Basic, Maschinen-Kit	12,- DM	22,- DM	US 9/85
				Backup/Directory/ Ellipse/Disk	-	27,- DM	US 10/85
				Data-Generator/Grafik Toolkit/ ASCII List (Turbo Pascal)	15,- DM	25,- DM	US 1/86
				Fallschirmspringer	9,- DM	-	HC/SR-3
				Geisterschloß	9,- DM	-	HC/SR-4
				Zeichendesigner	12,50 DM	-	HC/SR-5
				Mini Car Race/Interceptor 3 D	17,50 DM	26,50 DM	HC/SR-6
				Secret Agent	12,50	24,50	HC/SR-1-4
				Space Invaders/Mini Pac/ Otti's „Sound“	17,50 DM	27,50 DM	HC/SR-2-4
				Samson und die Kisten/ Die verflixte 15	17,50 DM	27,50 DM	HC/SR-3-4
				Moon-Landing/Panzer	17,50 DM	26,50 DM	HC/SR-4-4
				Super-Miner	14,- DM	24,- DM	SR 42
				CPC-Bert	14,- DM	24,- DM	SR 52
				Conan's Castle	15,50 DM	25,50 DM	SR 62
				Snider's Mace	16,50	26,50 DM	SR 72
				Schneider Panik/ Killership	18,50 DM	28,50 DM	SR 13
				Horror-Caves/ Midnight	18,50 DM	28,50 DM	SR 23
				Cosmis-Castle/Spider-Mace/ Traumland	18,50 DM	28,50 DM	SR 33


```

10106 DATA 99,00,02 :REM STA $0200,Y
10108 DATA C8 :REM INY
10110 DATA C8 :REM INY
10112 DATA C8 :REM INY
10114 DATA C8 :REM INY
10116 DATA C8 :REM INY
10118 DATA A9,C7 :REM LDA #$C7
10120 DATA A2,03 :REM LDX #$03
10122 DATA 8D,02,03 :REM STA $0302
10124 DATA 8E,03,03 :REM STX $0303
10126 DATA A2,00 :REM LDX #$00
10128 DATA 86,7A :REM STX $7A
10130 DATA CA :REM DEX
10132 DATA 86,3A :REM STX $3A
10134 DATA 4C,A2,C4 :REM JMP $C4A2
10136 DATA A9,83 :REM LDA #$83
10138 DATA A2,C4 :REM LDX #$C4
10140 DATA 8D,02,03 :REM STA $0302
10142 DATA 8E,03,03 :REM STX $0303
10144 DATA E6,14 :REM INC $14
10146 DATA D0,02 :REM BNE $03D7
10148 DATA E6,15 :REM INC $15
10150 DATA 4C,4F,03 :REM JMP $034F
10152 DATA 20,FD,CE :REM JSR $CEFD
10154 DATA 20,8A,CD :REM JSR $CDBA
10156 DATA 20,F7,D7 :REM JSR $D7F7
10158 DATA A5,14 :REM LDA $14
10160 DATA A4,15 :REM LDY $15
10162 DATA 60 :REM RTS
10200 :
10210 : RESTORE : P=0
10220 : FOR I = 828 TO 999
10230 : READ A$
10240 : GOSUB 10500
10250 : P = P + A
10260 : POKE I , A
10270 : NEXT I
10280 :
10290 : IF P = 20068 THEN 10400
10299 :
10300 PRINT"LIHNEN IST BEIM ABTIP-";
10305 PRINT"PEN DER DATAZEILEN EIN";
10310 PRINT"FEHLER UNTERLAUFEN!";
10315 PRINT"UEBERPRUEFEN SIE BITTE";
10320 PRINT"NOCH EINMAL DIE ENT-";
10325 PRINT"SPRECHENDEN PROGRAMM-";
10330 PRINT"ZEILEN!";END
10399 :
10400 PRINT"LSIE HABEN ALLE DATA-";
10405 PRINT"WERTE RICHTIG EINGEGE-";
10410 PRINT"BEN. WENN SIE MIT DIS-";
10415 PRINT"KETTE ARBEITEN,KOENNEN";
10420 PRINT"SIE DAS PROGRAMM ALS";
10425 PRINT"REINE MASCHINENROUTINE";
10430 PRINT"ABSPEICHERN. ES BELEGT";
10435 PRINT"IN DER FORM NUR EINEN";
10440 PRINT"BLOCK. FUER KASSETTEN-";
10445 PRINT"ANWENDER BESTEHT DIESE";
10450 PRINT"MOEGlichkeit NICHT!"

```

```

; EINER NULL ABSCHLIESSEN
; UND LAENGE DER ZEILE BERECHNEN
;
;
;
;
; BASIC-WARMSTART-VEKTOR UMBIEGEN
;
;
;
; ZEIGER INNERHALB DER CHRGET-ROUTINE
; AUF EINGABEPUFFER SETZEN
; DIREKTMODUS-FLAG SETZEN
;
; --> PROGRAMMEILE EINFUEGEN
; BASIC-WARMSTART-VEKTOR WIEDER NORMAL
;
;
; AKTUELLE ZEILENNUMMER ERHOEHEN
;
;
; NAECHSTES BYTE AUSWERTEN
; --> CHKCOM
; --> FRMNUM
; --> GETADR
;
;
;

```

```

10455 PRINT"ABSPEICHERN (J/N) ?"
10460 GETG$: IFG$="" THEN 10460
10465 IFG$<>"J" THEN 10485
10470 SYS57809 "DATAGEN.CODE",8,1
10475 POKE251,60:POKE252,3:POKE781,232
10480 POKE782,3:POKE780,251:SYS65496
10485 PRINT"GESTARTET WIRD DURCH:"
10490 PRINT"SYS828,AW,EW,SZ"
10495 END
10499 :
10500 A$=LEFT$(A$,2)
10510 A1=ASC(LEFT$(A$,1)):A2=ASC(RIGHT$(A$,1))
10520 A1=A1-48:IFA1>9 THEN A1=A1-7
10530 A2=A2-48:IFA2>9 THEN A2=A2-7
10540 A=16*A1+A2
10560 RETURN

```

**Auf der nächsten Seite
– präsentieren wir Ihnen ein –
weiteres VC-20-Programm!**

VC-20(3K) ★ VC-20(3K) ☆ VC-20(3K) ★ VC-20(3K)

Gehen Sie mit **Q-Bert** auf Punktejagd

Wer kennt es nicht, das bekannte „Q-Bert“-Spiel, das es inzwischen für alle Homecomputertypen gibt? Wir veröffentlichen hier eine Version, in der Sie die Aufgabe haben, alle 3-dimensionalen Felder des Quaders umzufärben. Diese Aufgabe ist natürlich wieder nicht so einfach wie es sich vielleicht anhören mag. Rutscht Q-Bert nämlich über den Rand eines Feldes, stürzt er sofort zu Tode. Außerdem wird er von „Helmut, dem Springer“ gejagt, der auch um Ecken springt, und somit ein Entkommen ziemlich schwierig ist. Daneben müssen

Sie noch auf die sogenannten „Schleimminen“ achten, die immer plötzlich vor Ihnen auftauchen und dann wieder verschwinden. Als weitere Schwierigkeit kommt der Faktor Zeit hinzu. Sie sehen also, es ist ein recht schwieriges Unterfangen, die gestellte Aufgabe zu lösen.

Wichtig: „Helmut der Springer“ und die „Schleimminen“ sind in der Lage, alle von Ihnen bereits gefärbten Felder bei Berührung wieder umzufärben. Aber auch Sie selbst können in die Situation kommen, bereits gefärbte Felder wieder umzufärben. Dann

nämlich, wenn sie diese Felder erneut berühren müssen. Für jedes gefärbte Feld erhalten sie 2 Punkte gutgeschrieben, für jedes umgefärbte Feld, werden diese Punkte wieder abgezogen. Nach jedem geschafften Level erhalten Sie Bonus-Punkte. Alle 1000 Punkte erhalten Sie einen Bonus-Q-Bert.

Gesteuert wird das Spiel mit dem Joystick, wobei zu beachten ist, daß keine Diagonalsteuerung möglich ist. Da das Hauptprogramm automatisch geladen wird, bitte nach dem Laden des ersten Teils nicht den Daten-Recorder stoppen!!!

Teil 1

```

5 PRINT" (CLEAR) ":GOTO90 <138>
10 ===== <180>
15 === PYRASOFT PROUDLY PRESENTS <23>
20 ===== <190>
25 ===== THE Q - B E R T ===== <86>
30 ===== <200>
35 ===== FOR VIC - 20 + 3K-RAM ===== <79>
40 ===== <210>
45 === (C) 6.4 1986 ANDREAS SPINDLER <93>
50 ===== <220>
55 ===== TEL. 06242 / 3390 ===== <95>
60 ===== <230>
65 ===== ALL RIGHTS RESERVED !! ===== <47>
70 ===== <240>
75 REM <218>
80 REMEMBER THE NEW PYRASOFT-GAMES TEL. <53>
06242/3390 <228>
85 REM
90 F=30720:C=36876:POKEC+2,15:POKEC-7,25 <38>
5:FORT=0T060:POKE7680+T,T <254>
95 POKE7680+T+F,0:NEXT
100 PRINT" (RVSON DOWN4 BLACK) DONT (SPACE) <166>
SWITCH (SPACE) OFF (SPACE) TAPE (SPACE) ! (DOWN <115>
SPACE3) PLEASE (SPACE) WAIT (SPACE) ... ":POK
E52,28:POKE56,28:GOSUB500
105 POKEC+3,255
110 PRINT" (CLEAR BLACK DOWN RIGHT) ,-. / : <166>
; < " :PRINT" (RVSON DOWN RIGHT) P (YELLOW) R <115>
RED) O (CYAN) U (PURPLE) D (GREEN) L (BLUE) Y (BLA

```

```

CK SPACE) PRESENTS" <44>
112 PRINT" (DOWN2 RIGHT) TUV (RIGHT8 DOWN <157>
RIGHT) W (RIGHT) X (RIGHT) YVZT"
114 PRINT" (DOWN2 RIGHT) (SPACE) 29 (RVSON <244>
) (RVSON) 3 (RIGHT RVSON) - (RVSON RIGHT) 6
(RVSON) (RVSON) 4 (SPACE) 1986 (RIGHT DOWN
RIGHT) ! (RIGHT) # % & (SPACE) ' ( ) * + (DOWN2) " <38>
125 T$=" (HOME RVSON DOWN) ? "
130 PRINT" INSTRUCTIONS (SPACE) ?? (SPACE) <8>
(J/N) "
140 IFPEEK(197)<>20ANDPEEK(197)<>28THEN1 <227>
30 <200>
145 IFPEEK(197)=28THEN170
150 FORT=8054T08075:POKEC,200:POKET+F,2: <172>
POKEC,0:POKET,189:NEXT:FORT=8163T081428T
EP-1
152 POKEC,200:POKET+F,2:POKET,189:POKEC, <253>
0:NEXT
155 PRINT" (RVSON UP4) F (RVSON SPACE) HE <213>
LMUT (SPACE2 RVSON) G (RVSON SPACE) SLIME-M
INE"
162 READTE$: IFTE$="" THEN 170 <17>
164 FORT=1T0LEN(TE$):PRINTT$TAB(21)MID$( <198>
TE$,T,1):PRINTT$ (SPACE) "CHR$(20)
166 FORT=1T05:POKEC-2,240:POKEC-2,0:POKE <48>
C,250:POKEC,0:NEXT
168 NEXT:GOTO162 <55>
170 FORT=8010T08163:POKEC,245:POKET,32:P <41>
OKEC,0:FORY=1T010:NEXTY,T
172 POKEC+3,25:PRINTT$" (WHITE RVSON UP5 <204>
) ":POKE198,1:POKE631,131:END
500 FORT=7168T07655:READA:POKET,A:POKEC, <26>
A:NEXT:FORT=7670T07679:READA:POKET,A:NEX
T:RETURN
501 DATA1,3,7,15,31,63,127,255,255,255,2 <203>
55,255,255,255,255,255,255,255,253,249,2
41,225
502 DATA193,129,1,3,7,15,31,63,127,255,2 <107>
55,255,255,255,255,255,255,255,255,255,2
53,249
504 DATA241,225,193,129,255,60,90,189,25 <248>
5,90,165,255,255,189,36,219,231,231,153,
255
506 DATA165,153,66,102,90,165,195,195,10
2,153,219,219,165,102,36,66,129,126,126,

```



```

126,126
508 DATA126,126,129,255,195,189,189,189,
189,195,255,255,231,219,189,189,219,231,
255,255
510 DATA255,231,219,219,231,255,255,255,
255,255,231,231,255,255,255,28,98,128,12
8,112
512 DATA14,226,28,28,98,128,128,128,66,7
6,48,56,68,68,132,130,130,98,28,24,36,66
,68,120
514 DATA144,136,132,56,70,64,64,224,128,
140,112,254,130,238,40,40,40,40,56,238,1
70,170
516 DATA186,130,186,170,238,254,130,142,
184,160,190,130,254,124,130,186,170,170,
190,137
518 DATA126,,124,130,124,,252,130,218
,100,90,194,130,252,124,130,186,170,186,
132,178
520 DATA238,255,255,24,24,24,24,24,24,12
6,126,24,24,24,24,126,126,198,238,254,25
4,214
522 DATA198,198,198,254,254,192,240,240,
192,254,254,60,66,153,161,161,153,66,60,
,111
524 DATA,,33,85,149,230,148,164,196,,
,49,85,87,85,229,169,169,,48,40,4
0,69,69
526 DATA73,114,,131,196,164,38,72,200,47
,,49,42,42,42,113,81,86,,12,18,16,8,4,34
,28
528 DATA,50,72,74,114,196,132,132,,67,82
,114,84,148,148,151,,4,132,136,72,80,144
,14,
530 DATA51,69,101,134,137,137,105,15,25,
50,100,248,192,192,192,98,98,100,105,115
,99,99
532 DATA99,60,100,200,145,227,67,35,35,6
0,100,200,144,231,38,39,32,60,126,102,10
2,102
534 DATA126,60,,28,60,124,28,28,126,126,
,60,126,78,28,48,126,126,,60,126,70,28,7
0,126
536 DATA60,,6,14,22,38,126,126,6,,126,12
6,64,124,6,126,60,,30,60
538 DATA120,124,102,126,60,,126,126,70,1
2,24,24,24,,24,60,36,126,102,126,60,,60,
126,102
540 DATA62,14,28,56,,24,48,96,125,15,27,
51,99,49,115,214,159,28,44,76,140,255,24
,48,96
541 DATA96,96,96,96,162,255,138,141,15,1
44,202,208,249,96
542 DATA" (SPACE8)HELLO (SPACE)FREAK (SPACE
)!! (SPACE)YOUR (SPACE)OBJECT (SPACE) IS (SPA
CE) IT (SPACE) TO (SPACE) SCORE (SPACE) AS (SPAC
E) MANY (SPACE) POINTS (SPACE) AS (SPACE) POSSI
BLE,"
544 DATA"BY (SPACE) CHANGING (SPACE) THE (SPA
CE) COLOR (SPACE) OF (SPACE) THE (SPACE) CUBES (
SPACE) WITH (SPACE) YOUR (SPACE) Q*BERT (SPACE
) FROM (SPACE) A (SPACE) STARTING (SPACE) COLOR
(SPACE)"
546 DATA"TO (SPACE) A (SPACE) DESTINATION (SP
ACE) COLOR. (SPACE) YOU'LL (SPACE) START (SPAC
E) THE (SPACE) GAME (SPACE) WITH (SPACE) THREE (
SPACE) Q*BERTS (SPACE)"
548 DATA"THE (SPACE) FIRST (SPACE) Q*BERT (SP
ACE) WILL (SPACE) APPEAR (SPACE) ON (SPACE) THE
(SPACE) TOPMOST (SPACE) CUBE (SPACE) WHEN (SPA
CE) THE (SPACE) GAME (SPACE) STARTS. (SPACE)"
550 DATA"BE (SPACE) CAREFULL (SPACE) NOT (SPA
CE) TO (SPACE) HOPE (SPACE) Q*BERT (SPACE) OUT (
SPACE) OF (SPACE) THE (SPACE) CUBES. (SPACE) IF
(SPACE) YOU (SPACE) DO, HE (SPACE) FALLS (SPACE
) AND (SPACE)"

```

<196>
<193>
<85>
<124>
<130>
<156>
<240>
<241>
<92>
<165>
<92>
<88>
<240>
<139>
<60>
<122>
<75>
<16>
<2>
<155>
<64>
<134>
<31>
<98>

```

552 DATA"YOU (SPACE) LOSE (SPACE) THAT (SPACE
) Q*BERT. BUT (SPACE) THERE (SPACE) ARE (SPACE)
MANY (SPACE) OTHER (SPACE) DANGERS (SPACE)! (S
PACE)"
554 DATA" (SPACE) BE (SPACE) CAREFULL (SPACE)
OF (SPACE) THE (SPACE) AWFULL (SPACE) MONSTER (
SPACE) HELMUT (SPACE)! (SPACE2) IF (SPACE) HE'
S (SPACE) CATCHING (SPACE) YOU, OR (SPACE)"
556 DATA"YOU'RE (SPACE) RUNNING (SPACE) IN (S
PACE) ONE (SPACE) OF (SPACE) THE (SPACE) SLIME-
MINES, YOU'LL (SPACE) LOSE (SPACE) THAT (SPACE
) Q*BERT (SPACE) TOO. (SPACE)"
560 DATA"HELMUT (SPACE) AND (SPACE) THE (SPAC
E) SLIME-MINES (SPACE) ARE (SPACE) CHANGING (S
PACE) THE (SPACE) CUBES' (SPACE) COLORS (SPACE
) IF (SPACE) THEY (SPACE) WANT (SPACE)! (SPACE)
"
562 DATA" (SPACE) THIS (SPACE) ALSO (SPACE) HA
PPENS, WHEN (SPACE) YOUR (SPACE) Q*BERT (SPACE
) IS (SPACE) JUMPING (SPACE) ON (SPACE) A (SPACE
) COLORED (SPACE) CUBE. (SPACE)"
564 DATA"AT (SPACE) THE (SPACE) END (SPACE) OF
(SPACE) EVERY (SPACE) LEVEL (SPACE) YOU'LL (SP
ACE) EARN (SPACE) BONUS (SPACE) - (SPACE) POINT
S. (SPACE)"
566 DATA"YOU'LL (SPACE) BECOME (SPACE) A (SPA
CE) BONUS (SPACE) - (SPACE) Q*BERT (SPACE) EVER
Y (SPACE) 1000 (SPACE) POINTS (SPACE)! (SPACE)
"
568 DATA" (SPACE) FOR (SPACE) PLAYING (SPACE)
USE (SPACE) JOYSTICK. (SPACE2)"
570 DATA" (SPACE) YOU (SPACE) MUSTN'T (SPACE)
MOVE (SPACE) THE (SPACE) STICK (SPACE) DIAGONA
LY (SPACE)!!"
572 DATA" (SPACE3) NOW (SPACE) PYRASOFT (SPAC
E) WISHES (SPACE) MUCH (SPACE) FUN (SPACE) BY (S
PACE) PLAYING (SPACE) THE (SPACE) GAME."
574 DATA" (SPACE2) 'THE (SPACE) Q*BERT' (SPAC
E) IS (SPACE) A (SPACE) TRADEMARK (SPACE) OF (SP
ACE) ANDREAS (SPACE) SPINDLER (SPACE) / (SPACE
) PYRASOFT, AND (SPACE) USED (SPACE) BY (SPACE2
) "
576 DATA" (SPACE) R (SPACE) O (SPACE) N (SPACE
) I (SPACE) C (SPACE2) - (SPACE2) VERLAG (SPACE)
GBH. (SPACE) UNDER (SPACE) AUTHORIZATION. (S
PACE)"
578 DATA" (SPACE2) COPYRIGHT (SPACE2) 6. 4 (SP
ACE) 1986 (SPACE) ANDREAS (SPACE) SPINDLER. (S
PACE2)" , "e"

```

Teil 2

```

10 REM =====<35>
15 REM ===== THE Q - BERT =====<141>
20 REM =====<45>
40 REM ===== VON ANDREAS SPINDLER =====<222>
45 REM ===== TEL. 06242/3390 =====<137>
50 REM<193>
60 POKE36869,242<171>
70 IFPEEK(37151)=62 THEN POKE646,RND(1)*17
:PRINT" (CLEAR DOHNZ RIGHT) - ATASSETTE (SPAC
E) STOPPEN (SPACE)!!":GOTO70
100 F=30720:A=7709:C=36876:DD=-1:LI=3:CS
=1000:BO=50
102 A$(0)=" (RIGHT6) TUV (RIGHT) WXYVZT":A$(
2)=" (RIGHT) (RIGHT) 4 (RVSON) " (RVSOFF) B6 (R
IGHT) (RIGHT) *%&(SPACE) (1)*"
104 A$(1)=" (RIGHT6) (RIGHT2) , - . / : ; <"
110 POKEC+2,15:POKEC-7,255
120 POKEC+3,238:B(1)=. :X2=. :X1=. :LE=INT(
RND(1)*5):IFLE=0 THEN120
121 PRINT" (CLEAR) ":ONLE GOSUB125,125,140,
160
122 GOSUB132:GOTO200

```

<248>
<233>
<146>
<223>
<145>
<221>
<221>
<38>
<9>
<144>
<218>
<1>
<152>

```

125 F1=1:P=A+1:A1=INT(RND(1)*8):A2=INT(R
ND(1)*7):A3=INT(RND(1)*8)
126 Q=. :A4=INT(RND(1)*8):IFA4=A3ORA3=1OR
A4=1 THEN120
127 IFA1<2ORA2<1 THENA1=7:A2=6
128 IT=(A1^2+A2^2)*2.5:A1=A1*43:A2=A2*24
:FORT=0TOA1STEP43:FORY=0TOA2STEP24
129 POKEA+T+Y,0:POKEA+1+T+Y,1:POKEA+2+T+
Y,2:POKEA+22+T+Y,1:POKEA+23+T+Y,1
130 POKEA+22+T+Y+F,A3:POKEA+23+T+Y+F,A3:
Q=Q+1:NEXT:POKEA+T+Y,206:NEXT:Q=Q-1:SC=S
C+2
131 RETURN
132 POKEP-1,3:POKEP,4:POKEP+1,5:POKEP-1+
F,A4:POKEP+F,A4:POKEP+1+F,A4:RETURN
140 P=A+22:A4=6:A3=0:PRINT" (DOHN WHITE S
PACE6) @AB
141 PRINT" (SPACE6 BLACK) AA (WHITE) @AB
142 PRINT" (SPACE5) @AB (BLACK) AA (WHITE) @AB
143 PRINT" (SPACE5 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB
144 PRINT" (SPACE4) @AB (BLACK) AA (WHITE) @AB
(BLACK) AA (WHITE) @AB
145 PRINT" (SPACE4 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB (BLACK) AA (WHITE) @AB
146 PRINT" (SPACE3) @AB (BLACK) AA (WHITE) @AB
(BLACK) AA (WHITE) @AB (BLACK) AA (WHITE) @AB
147 PRINT" (SPACE3 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB (BLACK) AA (WHITE RVSON) / (R
VSOFF SPACE2 BLACK) AA (WHITE RVSON) /
148 PRINT" (SPACE2) @AB (BLACK) AA (WHITE) @AB
(BLACK) AA (WHITE RVSON) / (RVSOFF SPACE4) @A
B
149 PRINT" (SPACE2 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE RVSON) / (RVSOFF SPACE4) @AB (BL
ACK) AA (WHITE RVSON) /
150 PRINT" (SPACE) @AB (BLACK) AA (WHITE RVSO
N) / (RVSOFF SPACE4) @AB (BLACK) AA (WHITE) @AB
151 PRINT" (SPACE BLACK) AA (WHITE) @AB (SPAC
E2) @AB (BLACK) AA (WHITE) @AB (BLACK) AA (WHITE
RVSON) /
152 PRINT" (SPACE3 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB (BLACK) AA (WHITE) @AB
153 PRINT" (SPACE5 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB (BLACK) AA (WHITE RVSON) /
154 PRINT" (SPACE7 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE) @AB
155 PRINT" (SPACE9 BLACK) AA (WHITE) @AB (BLA
CK) AA (WHITE RVSON) /
156 PRINT" (SPACE11 BLACK) AA (WHITE) @AB (SP
ACE19 BLACK) AA (WHITE RVSON) /":IT=150:Q=3
7:RETURN
160 P=A+22:A3=4:A4=7:PRINT" (DOHN SPACE6)
@AB
161 PRINT" (SPACE6 PURPLE) AA (WHITE) @AB
162 PRINT" (SPACE5) @AB (PURPLE) AA (WHITE) @A
B
163 PRINT" (SPACE5 PURPLE) AA (WHITE RVSON)
/ (RVSOFF SPACE2 PURPLE) AA (WHITE) @AB
164 PRINT" (SPACE4) @AB (SPACE5 PURPLE) AA (W
HITE) @AB
165 PRINT" (SPACE4 PURPLE) AA (WHITE) @AB (SP
ACE5 PURPLE) AA (WHITE) @AB
166 PRINT" (SPACE3) @AB (PURPLE) AA (WHITE) @A
B (SPACE5 PURPLE) AA (WHITE) @AB
167 PRINT" (SPACE3 PURPLE) AA (WHITE) @AB (PU
RPLE) AA (WHITE) @AB (SPACE5 PURPLE) AA (WHITE
RVSON) /
168 PRINT" (SPACE2) @AB (PURPLE) AA (WHITE) @A
B (PURPLE) AA (WHITE) @AB (SPACE2) @AB
169 PRINT" (SPACE2 PURPLE) AA (WHITE RVSON)
/ (RVSOFF SPACE2 PURPLE) AA (WHITE) @AB (PURP
LE) AA (WHITE) @AB (PURPLE) AA (WHITE RVSON) /
170 PRINT" (SPACE) @AB (SPACE5 PURPLE) AA (WH
ITE) @AB (PURPLE) AA (WHITE) @AB

```

<244>
<108>
<31>
<176>
<49>
<158>
<17>
<134>
<30>
<215>
<105>
<104>
<11>
<91>
<15>
<159>
<62>
<239>
<175>
<255>
<20>
<204>
<251>
<168>
<203>
<165>
<87>
<1>
<157>
<120>
<100>
<212>
<135>
<150>
<39>
<120>

```

171 PRINT" (SPACE PURPLE) AA (WHITE) @AB (SPA
CE5 PURPLE) AA (WHITE) @AB (PURPLE) AA (WHITE
RVSON) /
172 PRINT" (SPACE3 PURPLE) AA (WHITE) @AB (SP
ACE5 PURPLE) AA (WHITE) @AB
173 PRINT" (SPACE5 PURPLE) AA (WHITE) @AB (SP
ACE5 PURPLE) AA (WHITE RVSON) /
174 PRINT" (SPACE7 PURPLE) AA (WHITE) @AB (SP
ACE2) @AB
175 PRINT" (SPACE9 PURPLE) AA (WHITE) @AB (PU
RPLE) AA (WHITE RVSON) /
176 PRINT" (SPACE11 PURPLE) AA (WHITE) @AB
177 PRINT" (SPACE13 PURPLE) AA (WHITE RVSON)
/":IT=150:Q=31:RETURN
200 REM
202 POKEC+2,15:IFRND(1)*2<1 THEN350
205 IFRND(1)*2<1ORB(1)=0 THENGOSUB300
210 IFZ1 THENZ1=0:GOTO500
220 IFPEEK(B(1)+1)=2 THENPOKEB(1),1
225 IFPEEK(B(1)+1)=5 THENPOKEB(1),4
228 IFB1 THENB(1)=B1
232 P1=INT(RND(1)*8):IFP1=0 THENP1=24
233 IFP1=1 THENP1=-24
234 IFP1=2 THENP1=43
235 IFP1=3 THENP1=-43
236 IFP1=4 THENP1=67
237 IFP1=5 THENP1=-67
238 IFP1=6 THENP1=19
239 IFP1=7 THENP1=-19
240 IFPEEK(B(1)+P1)=1 THENIFPEEK(B(1)+P1+
1)=2 THEN250
242 IFPEEK(B(1)+P1)=4 THEN250
244 IFPEEK(B(1)+P1)=8 THENTD=2:GOSUB600:G
OTO562
245 POKEB(1),6:GOTO500
250 B(1)=B(1)+P1:POKEB(1),6:IFPEEK(B(1)+
1)=5 ANDRND(1)*3<1 THEN260
252 FORT=127TO150STEP8:POKEC-2,T:FORT=15
0TO127STEP-8:POKEC-2,T:NEXT
255 POKEC-2,0:GOTO500
260 POKEB(1)-1,0:POKEB(1),1:POKEB(1)+1,2
:POKEB(1),6
265 POKEB(1)-1+F,1:POKEB(1)+F,1:POKEB(1)
+1+F,1:Q=Q+1:SC=SC-2
267 IFPEEK(B(1)+23)=206 THENPOKEB(1)+F+23
,1
275 FORT=220TO240:POKEC,T:NEXT:FORT=240T
O220STEP-1:POKEC,T:NEXT
280 POKEC,0:GOTO500
300 B1=INT(RND(1)*506)+7658:IFPEEK(B1)<>
8 THENIFPEEK(B1+1)=2 THEN320
305 IFPEEK(B1)=4 THEN320
310 Z1=1:B1=0
320 RETURN
350 IFX1=0 THENX3=INT(RND(1)*12):X2=INT(R
ND(1)*X3):X1=1:GOTO355
352 IFX1 THENFORT=0TOX2:GOSUB400:X2(T)=0:
NEXT:X1=0:GOTO500
355 FORT=0TOX2:GOSUB300:IFZ1 THENZ1=0:NEX
T:GOTO500
360 X2(T)=B1:IFPEEK(X2(T))=8 THENTD=2:GOS
UB600:GOTO562
370 POKEX2(T),7:NEXT:GOTO500
400 IFPEEK(X2(T)+1)=2 THENPOKEX2(T),1
410 IFPEEK(X2(T)+1)=5 THENPOKEX2(T),4:IFR
ND(1)*2<1 THENGOSUB425
420 RETURN
425 POKEC+2,15:POKEX2(T)-1,0:POKEX2(T)-1
+F,1:POKEX2(T),1:POKEX2(T)+F,1:POKEX2(T)
+1,2
428 IFPEEK(X2(T)+23)=206 THENPOKEX2(T)+F+
23,1
430 POKEX2(T)+1+F,1:FORY=255TO127STEP-1:
POKEC-2,Y:NEXT:FORY=127TO255:POKEC-2,Y:N
EXT

```

<171>
<63>
<121>
<8>
<48>
<196>
<253>
<87>
<158>
<199>
<225>
<111>
<212>
<236>
<157>
<140>
<135>
<162>
<211>
<244>
<195>
<225>
<231>
<231>
<190>
<158>
<192>
<131>
<140>
<72>
<146>
<81>
<168>
<117>
<4>
<109>
<18>
<207>
<136>
<105>
<13>
<56>
<159>
<94>
<60>
<51>
<59>
<5>
<255>


```

435 POKEC-2,0:SC=SC-2:T=Q=Q+1:RETURN <149>
500 J1=PEEK(37137):POKE37154,127:J2=PEEK
(37152):POKE37154,255 <116>
501 PRINT"<HOME WHITE RIGHT12 BLACK>OPQR
S"INT(SC)"<LEFT SPACE>" <237>
502 IT=IT-1:IFIT<=0THENTD=2:GOSUB600:GOT
0562 <103>
503 PRINT"<HOME DOWN21 RIGHT>[\]^"INT(IT
)"<LEFT SPACE>" <167>
504 IFJ1=126ANDJ2=247THENIT=IT+.5:GOTO20
0 <2>
506 POKEP-22,32:POKEP-22+F,A3:POKEP+F,A4
:IFPEEK(P-21)=1THENPOKEP-22,1 <124>
507 POKEP,4:IFPEEK(P-1)=0THENPOKEP,1 <236>
508 IFL1THENL1=0:POKEP+F,1 <124>
510 IF(J1AND4)=0THENP1=-43 <96>
515 IF(J1AND8)=0THENP1=43 <189>
520 IF(J1AND16)=0THENP1=-24 <187>
525 IF(J2AND128)=0THENP1=24 <72>
526 P=P+1:IFPEEK(P)=1THEN531 <100>
527 IFPEEK(P)=4THEN535 <151>
528 IFPEEK(P)=6ORPEEK(P)=7THENTD=2:GOSUB
600:GOTO562 <167>
529 TD=1:B(1)=0:GOSUB600:GOTO562 <238>
531 POKEP-1,3:POKEP,4:POKEP+1,5:POKEP+F-
1,A4:POKEP+F,A4:POKEP+F+1,A4:A5=1 <1>
532 Q=Q-1:SC=SC+2:IFPEEK(P+23)=206THENPO
KEP+23+F,A4 <54>
534 GOTO540 <103>
535 POKEP-1,0:POKEP,1:POKEP+1,2:POKEP-1+
F,1:POKEP+F,1:POKEP+1+F,1 <100>
536 Q=Q+1:SC=SC-2:L1=1:IFPEEK(P+23)=206T
HENPOKEP+23+F,1 <137>
540 FORT=15TO5:POKEC+2,T:POKEC,200:NEXT:
POKEC+2,15:POKEC,0 <6>
541 POKEP-22,9:POKEP+F-22,0:POKEP,8:POKE
P+F,0 <215>
542 IFASTHENFORY=242TO252STEP(RND(1)*7)+
.3:POKEC,Y:POKEC,0:NEXT:A5=0 <154>
544 IFQ=0THEN550 <153>
545 GOTO200 <96>
550 GOSUB630 <122>
562 POKEC+3,25:SC=INT(SC) <32>
563 IFTD<>1THENPRINT"<CLEAR DOWN6 RVSON
RED>B<SPACE CYAN>O<SPACE PURPLE>N<SPACE
GREEN>U<SPACE BLUE>S:<RVSOFF>"BO"<RVSON
LEFT SPACE>POINTS!":SC=SC+BO+IT:BO=BO+25 <56>
564 IFTD<>1THENGOSUB630 <137>
565 PRINT"<CLEAR DOWN WHITE>" <246>
566 POKEC+3,8:DD=DD+1:FORT=1TOLEN(A$(DD)
):PRINTLEFT$(A$(DD),T)"<UP>" <7>
568 FORY=1TO6:POKEC,200:POKEC,0:NEXTY,T:
PRINT"<DOWN2>":IFDD<>2THEN566 <117>
569 IFSC<>CSTHENCS=CS+1000:PRINT"<RVSON
RIGHT3 DOWN>BONUS<SPACE>-<SPACE>Q*BERT<S
PACE>:<UP2>":LI=LI+1:GOSUB630 <193>
571 IFL1=0THENPRINT"<DOWN RVSON RIGHT2>Y
OU'VE<SPACE>KILLING<SPACE>ALL<RIGHT8 DO
W N>Q*BERT'S<SPACE>!":GOSUB630:GOTO596 <63>
572 IFTD=1THENPRINT"<DOWN RVSON RIGHT3>Y
OU<SPACE>LOSE<SPACE>AT<SPACE>THIS<RIGHT3
DOWN RIGHT8>LEVEL<SPACE>!!!":TD=0:GOTO5
80 <211>
575 PRINT"<DOWN RVSON RIGHT3>CONGRATULAT
IONS<SPACE>:<RIGHT2 DOWN>YOU'VE<SPACE>WO
N<SPACE>THIS<SPACE>STAGE!" <158>
580 PRINT"<DOWN RIGHT3 RVSON>YOUR<RVSOFF
SPACE>OPQRS<SPACE RVSON>:<RVSOFF>"INT(S
C):PRINT"<RIGHT3 DOWN>"LI"<SPACE>W<SPACE
>X<SPACE>YVZT<RVSON>"S!":GOSUB630 <197>
590 IFPEEK(37137)=94THENPOKEC,0:DD=-1:GO
TO120 <79>
592 FORT=200TO250STEP.5:POKEC,T:NEXT:FOR
T=250TO200STEP-.5:POKEC,T:NEXT:GOTO590 <205>
596 POKEC+3,8:PRINT"<DOWN RVSON RIGHT3>P

```

```

LAY{SPACE}*AGAIN{SPACE}??{SPACE}(J/N)"      <108>
597 GETA$: IFA$<"J"ANDA$<"N"THEN597          <206>
598 IFA$="J"THENRUN                             <255>
599 POKEC+3,27:POKEC-7,240:PRINT"<CLEAR
BLUE DOWN3>RUN<UP3 LEFT3>":END                <172>
600 REM A.J.S.A                                  <63>
610 POKEC+2,15:IFTD=1THENFORT=255T0127ST
EP-2:POKEC,T:FORY=15T00STEP-2:POKEC+2,Y:
NEXTY,T                                          <211>
612 IFTD=1THENFORT=15T00STEP-.5:POKEC+2,
T:POKEC+1,180+3*T:NEXT:POKEC+2,15:POKEC+
1,0                                             <96>
613 IFTD=2THENY=9:FORT=1T05:Y=Y+1:POKEP-
22,Y:POKEP,Y:POKEP-22+Y,A3:POKEP+Y,1        <216>
614 IFTD=2THENFORX=15T00STEP-.4:POKEC+2,
X:POKEC+1,200+3*X:POKEC-2,200+3*X:NEXTX,
T                                              <168>
615 IFTD=2THENPOKEC+1,0:POKEC-2,0             <32>
616 IFTD=2THENIFPEEK(P-21)=1THENPOKEP-22
,1                                              <87>
617 POKEC+2,15:IFTD=2ANDPEEK(P-21)=32THE
NPOKEP-22,32                                  <63>
618 IFTD=2THENFORT=1T014:POKEC+1,120+9*T
:POKEP,6:FORY=1T0100:NEXT:POKEC+1,0        <200>
619 IFTD=2THENPOKEP,7:FORY=1T0100:NEXTY,
T                                              <50>
628 LI=LI-1:TD=1:RETURN                       <116>
630 C1=INT(RND(1)*6)+1:FORT=127T0255STEP
C1:POKEC+3,8:SYS7670:POKEC+3,238:POKEC,T
:NEXT                                           <123>
632 FORT=255T0127STEP-C1:POKEC+3,8:SYS76
70:POKEC+3,238:POKEC,T:NEXT                 <123>
634 POKEC,0:POKEC+3,8:SYS7670:RETURN        <193>

```

Listing: The End



Auch auf dieser „High-Society“-Party gibt es nur ein Gesprächsthema: „Die tollen Programme der Compute mit.“ Achten Sie auf unser Super-Software-Angebot (S. 32,33)

6502 Assembler-Kurs

Teil 13

RTS Return from Subroutine (Kehre von Unterprogramm zurück)

Dieser Rücksprungbefehl eines Unterprogrammes ist vergleichbar mit dem Basic-Befehl „RETURN“. Er holt einen neuen Befehlszähler-Wert von der Spitze des Stapels und inkrementiert ihn, bevor er zum Holen eines Befehles verwendet wird. Beachten Sie, daß der Stapelzeiger vor dem Laden jedes Datenbytes inkrementiert wird und sein endgültiger Wert um den Wert 2 größer ist als sein Anfangswert.

HEX	Adressierung	Bytes	Syntax
60	Impliziert	1	RTS

7 6 5 4 3 2 1 0
NV BD I Z C
nicht beeinflusst
jedoch abgespeichert

SBC – Subtract Memory from Accumulator (Subtrahiere Speicher vom Akkumulator)

Dieser Befehl subtrahiert den Inhalt des gewählten Speicherbytes und das invertierte Übertragungsflag (Carry) vom Inhalt des Akkumulators und bietet die gleichen Speicher-Adressierungsmöglichkeiten wie der ADC-Befehl.

HEX	Adressierung	Bytes	Syntax	NV	BD	I	Z	C
E1	Indirekt, vor-indiziert mit X	2	SBC (addr,x)	x	x			
E5	Null-Seite (direkt)	2	SBC addr					
E9	Unmittelbar	2	SBC #data					
ED	Absolut (direkt)	3	SBC addr					
F1	Indirekt, nach-indiziert mit Y	2	SBC (addr),Y					
F5	Null-Seite indiziert mit X	2	SBC addr,X					
F9	Absolut indiziert mit Y	3	SBC addr,Y					
FD	Absolut indiziert mit X	3	SBC addr,X					

Das nachfolgende Beispiel demonstriert eine 8-Bit-Subtraktion. Nachdem man das Programm eingegeben hat, wird der erste Operand in Speicherstelle 832 und der zweite Operand in 833 gepoked. Danach kann unsere Routine mit SYS 49152 aufgerufen werden. Das Ergebnis der Subtraktion wird über eine ROM-Routine auf dem Bildschirm ausgegeben.

2			
		;	
		;	BEISPIEL
30:	C000		
		;	8 BIT SUBTRAHTION MIT DEM SBC-BEFEHL
		;	
50:	C000	.OPT	P1,00
70:	C000	*=	49152
		;	
90:	C000	ZPRINT	= \$BDCD ;ROUTINE ZUR ZAHLENAUSGABE
100:	C000	ZAHL1	= 832 ;SPEICHER FUER ZAHL 1
110:	C000	ZAHL2	= 833 ;SPEICHER FUER ZAHL 2
120:	C000	ZAHL3	= 834 ;SPEICHER FUER ZAHL DIVERENZ
		;	
140:	C000	18	CLC
150:	C001	AD 40 03	LDA ZAHL1 ;LADE ERSTE ZAHL
160:	C004	ED 41 03	SBC ZAHL2 ;SUBTRAHIERE ZAHL2 VON ZAHL1
170:	C007	8D 42 03	STA ZAHL3 ;SPEICHERE ERGEBNIS AB
180:	C00A	AA	TAX ;ZAHL FUER ZPRINT NACH X-REGISTER LEG

EN
 190: C00B A9 00 LDA #0 ;HIGH BYTE LOESCHEN
 200: C00D 20 CD BD JSR ZPRINT
 210: C010 60 RTS

Erweitern wir unser Programm etwas, läßt sich sehr schnell auch eine 16-Bit-Subtraktion bewerkstelligen. Hier müssen die einzelnen Operanden jeweils in Low- und Highbyte aufgetrennt und in zwei Speicherstellen abgelegt werden. Der Arbeitsablauf erfolgt wie in Beispiel 1.

SEC Set Carry (Setze Übertrag)

Dieser Befehl setzt das C-Flag auf 1. Es werden keine anderen Status oder Register-Inhalte beeinflusst. Beachten Sie, daß dieser Befehl als Teil einer normalen Subtraktions-Operation benötigt wird, da der einzige beim 6502- und 6510-Prozessor verfügbare Subtraktions-Befehl der SBC ist, welcher auch den komplementierten Übertrags-Status subtrahiert. Dieser Befehl ist auch beim Beginn einer Multibyte-Subtraktion erforderlich, da es niemals ein Borgen vom niedrigsten Byte gibt.

Hex	Adressierung	Bytes	Syntax	7 6 5 4 3 2 1 0
38	Impliziert	1	SEC	NV B D I Z C x

SED Set Decimal Mode (Setze Dezimal-Betriebsart)

Mit diesem Befehl läßt sich der C-64 auf den Dezimal-Modus umschalten. Dieser Modus ist eine Besonderheit des 6510 und 6502. Die dezimale Betriebsart wird dazu verwandt, um den Processor 6502 in die Dezimal-Betriebsart zu versetzen, in der ADC- und SBC-Befehle BCD-Ergebnisse anstatt binäre Resultate liefern. Der Programmierer sollte sehr sorgfältig auf die Tatsache achten, daß das gleiche Programm unterschiedliche Ergebnisse erzeugen wird, abhängig vom Zustand des Dezimal-Betriebsarts-Status. Dies kann zu unangenehmen und überraschenden Fehlern führen, wenn der Zustand des Dezimal-Betriebsarts-Status nicht sorgfältig beobachtet wird. Beim C-64 wird nach dem Einschalten das D-Flag gelöscht und damit auf die binäre Arithmetik umgeschaltet. Kleinere Maschinen können deshalb immer davon ausgehen, daß sich der C-64 im Binären-Modus befindet, einige Ausnahmen sind lediglich einige Routinen des Basic-Interpreters, welche mit Fließkomma-Zahlen hantieren.

Hex	Adressierung	Bytes	Syntax	7 6 5 4 3 2 1 0
F8	Impliziert	1	SED	NV B D I Z C x

SEI Set Interrupt Mask (disable Interrupt) (Setze Unterbrechungs-Maske-Sperre Unterbrechung)

Dies ist das Gegenstück zum CLI-Befehl. Dieser Befehl setzt das I-Flag des Statusregister und sperrt dadurch alle Unterbrechungen, welche durch den IRQ-Eingang des Processors aktiviert werden. Der NMI-Eingang wird durch diesen Befehl nicht gesperrt und bleibt deshalb weiterhin wirksam. Der SEI sowie der CLI-Befehl sollte beim C-64 mit größter Vorsicht behandelt werden, da hier normalerweise 50 mal in der Sekunde eine Interrupt-Routine aufgerufen wird, welche die Uhr stellt und die Tastatur abfragt. Wird in einem Ihrer Programme der Interrupt ausgeschaltet, kann der Rechner die Tastatur nicht mehr kontrollieren und läßt sich nur über den NMI (RUN/STOP RESTORE) wieder unter Kontrolle bringen. Wir werden später noch aufzeigen, wie sich der Interrupt beim C-64 ausnutzen läßt um Grafikeffekte und Sound zu erzeugen, ohne das ein nebenbei ablaufendes Basic-Programm dabei stört.

Hex	Adressierung	Bytes	Syntax	7 6 5 4 3 2 1 0
78	SEI	1	SEI	NV B D I Z C x

Literatur:

6502 Microcomputer-Programmierung,
 Peter Heuer, Hofacker Verlag
 6502 Programmierung in Assembler,

Lance A. Leventhal
 65 Intern, Angershausen,
 Becker, Englisch, Gerits,
 Data-Becker Buch

6502/65c02, Christian Persson,
 Heinz Heise Verlag
 Reverenz Map, F. Brall
 Butterfield's Lehrbuch,
 J. Butterfield, Hanser-Verlag

Ein MINI-FORTH auf dem SCHNEIDER CPC-464

Interessieren Sie sich für Forth? Diese Programmiersprache hat ja in letzter Zeit regelrecht Furore gemacht und liegt nun auch für den Schneider CPC-464 in verschiedenen Versionen vor. Alle zusammen haben denselben Nachteil: Sie kosten einen Batzen Geld. Mit dem abgedruckten Forth-Simulator können Sie die Grundzüge diese Sprache erlernen und dann entscheiden, ob Forth für Sie in Frage kommt.

Die Vor- und Nachteile von Forth wurden ja schon oft genug erörtert, hier deshalb die wichtigsten Eigenschaften nur in Stichworten:

- Fast so schnell wie Maschinensprache, aber nicht so schwierig zu programmieren.
- Geringer Speicherplatzbedarf sowohl des Interpreters selbst als auch der erzeugten Programme.
- Aufgrund der umgekehrten Polnischen Notation (s. u.) schwieriger zu erlernen als z. B. Basic oder Pascal.
- Möglichkeit zur dynamischen Definition von Befehlen. Somit kann für jede Anwendung (Grafik, Textverarbeitung, Arithmetik) ein spezieller Befehlssatz kreiert werden.

Zuerst zur umgekehrten Polnischen Notation (UPN oder RPN), die dem Forth-Anfänger am meisten zu schaffen macht: Forth kennt zwar Variablen, benutzt sie aber kaum, da alle Daten auf Stacks, also Stapelspeichern, abgelegt werden. Maschinenprogrammierer wissen damit sicher etwas anzufangen (PUSH und POP), für die übrigen folgt hier die wohl meistverwendete Erklärung: Stellen Sie sich einen Tellerstapel vor; auf diesen können Sie oben Teller auflegen und oben auch wieder entnehmen. Solange Sie kein zirkusreifer Artist sind, dürfte es unmöglich sein, weiter unten liegende Teller direkt wegzunehmen. Sie müssen alle darüberliegenden Teller beiseite legen, um an diesen Teller heranzukommen. Diese Speicherorganisation nennt man LIFO-Last in, first out: was zuletzt auf den Stapel kommt, kann auch zuerst wieder weggenommen werden. Stellen Sie sich das Ganze nun statt mit

Tellern mit Zahlen vor: Sie können beliebig viele Zahlen auf den Stapel, neudeutsch „Stack“, legen. Die Rechenbefehle, z. B. die vier Grundrechenarten, verknüpfen jeweils die beiden obersten Stapелеlemente miteinander: Legen Sie die Zahlen 33 und 44 auf den Stapel und geben Sie ein Plus-Zeichen ein, so nimmt der Computer beide Werte vom Stack, addiert sie und legt das Ergebnis wieder auf den Stapel. Die Zahlen 33 und 44 selbst sind dann verschwunden.

Nach dieser kurzen Einführung sollten Sie die weiteren Übungen direkt am Computer durchführen. Geben Sie dazu den Forth/UPN-Simulator ein. Dieser ist in Basic geschrieben und damit natürlich erheblich langsamer als Forth normalerweise. Es besteht aber damit auch kaum die Gefahr, den Computer zum Abstürzen zu bringen, was Anfänger – und auch Profis – bei Forth des öfteren schaffen. Außerdem kennt das Programm natürlich nicht den gesamten Befehlssatz, so fehlen z. B. Schleifen- und Vergleichsbefehle. Um die Grundzüge der Programmierung in Forth zu erlernen, reicht das Programm aber voll aus. Da der Simulator in Basic geschrieben ist, steht er Ihren Erweiterungen natürlich offen.

Haben Sie das Programm abgetippt, sollten Sie es sicherheitshalber abspeichern. Auch ist es zu empfehlen, bei der Austestphase des Programms die ON-ERROR-Anweisung in Zeile 540 erst einmal wegzulassen, damit Sie Tippfehler in der eingegebenen Version entdecken können. Sobald Sie das Programm starten, schaltet es auf 80 Zeichen pro Zeile um und zeigt den Arbeitsbildschirm. Auf der rechten Bildschirmhälfte finden Sie alle residenten Forth-Befehle, die im folgenden Text an-

gesprochen werden. Links oben ist der Stack-Pointer SP zu finden, der immer auf das oberste Element des Stacks zeigt. So können Sie während der Rechnungen immer die Stapel-Aktivitäten verfolgen. Beim Programmstart wird der Stack auf die Adresse 24576 gesetzt. Sie haben 16 KByte frei für den Stack und können 8192 16-Bit-Integerzahlen speichern. Dies ist erheblich mehr als bei anderen Forth-Systemen und reicht auf jeden Fall aus. Das verbleibende umrahmte Bildschirmfenster, in dem der Cursor steht, ist Ihr Eingabefeld. Auch werden die Antworten des Computers hier ausgegeben. Sie können bis zu 253 Zeichen in eine Eingangszeile packen; längere Eingaben werden zurückgewiesen.

Wiederholen Sie doch noch einmal die Rechnung von vorhin. Tippen Sie ein:

33 <ENTER> – der Stapelzeiger erhöht sich auf 24578
 44 <ENTER> – der Stackzeiger steht jetzt auf 24580
 + <ENTER> – jetzt wird der Stapel wieder auf 24578 erniedrigt.

Nun wollen Sie aber auch ein Ergebnis sehen. In Basic würden Sie jetzt PRINT eingeben, Forth hat dafür den Punkt „.“ reserviert. Geben Sie also noch einen Punkt ein und drücken Sie wieder ENTER. Kurz darauf erscheint das Ergebnis 77.

Natürlich können Sie auch alle Eingaben in eine einzige Zeile packen:

33 44 + . <ENTER>

Das Ergebnis ist dasselbe. Die übrigen drei Grundrechenarten funktionieren genauso:

3 4 * . ergibt 12
 12 3 / . ergibt 4
 4 3 - . ergibt 1

Eine Kombination dieser Rechnungen ist natürlich auch möglich:

3 4 * 5 + . ergibt 17

Achten Sie unbedingt darauf, zwischen allen Operanden und Operatoren mindestens ein Leerzeichen einzufügen. Mehr Leerzeichen schaden nicht, drücken aber etwas die Verarbeitungsgeschwindigkeit. Alle Eingaben können sowohl in Klein- als auch in Großbuchstaben erfolgen.

Jetzt bleibt nur noch eines zu tun: Üben, üben, üben! Denn dies ist die wohl wichtigste Grundlage von Forth. Ohne Kenntnis der UPN kein Forth!

Nun gibt es noch weitere Befehle, mit denen der Stapelspeicher manipuliert werden kann. Dazu sind noch zwei Fachbegriffe zu erklären: TOS heißt „Top of Stack“, meint also das oberste Stapелеlement. „2OS“ heißt „Second Element of Stack“ und ist das zweite Stapелеlement, das direkt unter dem TOS liegt. Es gibt Befehle, die nur den TOS verändern, und andere, die TOS und 2OS benötigen.

1+ und 2+ sind nur der Vollständigkeit halber aufgenommen und funktionieren genauso wie „1 +“ und „2 +“. Bei richtigen Forth-Interpretern bringen sie einen kleinen Geschwindigkeitsgewinn.

SQR ist ähnlich der Pascal-Funktion SQR, nicht zu verwechseln mit Basic's SQR. Dieser Befehl quadriert den TOS. Achten Sie aber – wie bei allen anderen Berechnungen darauf, daß nur Integer-Zahlen verarbeitet werden können, die im Bereich von -32768 bis 65535 liegen können.

Beispiel: 3 SQR . ergibt 9

DUP dupliziert den TOS, d. h. der Inhalt des TOS wird noch einmal auf den Stapel gelegt. So können z. B. Zwischenergebnisse gespeichert werden, wenn diese Zahl nochmals benötigt wird.

Beispiel: 3 DUP * . ergibt 3 3 * . und hat das Ergebnis 9.

DROP vergibt den TOS und erniedrigt den Stapelzeiger dementsprechend.

Beispiel: 3 4 DROP . ergibt 3, da der TOS-Wert 4 wieder vergessen wurde.

SWAP tauscht den Inhalt des TOS und des 2OS aus. Damit können z. B. Divisionen und Subtraktionen vorbereitet werden, bei denen es auf die Stellung der Operanden ankommt.

Beispiel: 4 3 - . ergibt 4-3=1
4 3 SWAP - . ergibt 3-4=-1

SWPB ist vor allem für Maschinenprogrammierer interessant, denn SWPB heißt „Swap Bytes“ und tauscht das High- und das Low-Byte des TOS aus.

Beispiel: &0003 SWPB . ergibt 768 (hexadezimal &0300)

Hier sehen Sie auch, daß Sie überall auch Hexadezimal- und Binärzahlen verwenden können – bis auf den Sonderfall Null: Diese Zahl muß dezimal eingegeben werden. Das hat das Programm um einige Zeilen verkürzt.

ABS entspricht der Basic-Funktion ABS. Es bildet den Absolutwert des TOS.

Beispiel: -3 ABS . ergibt 3

NEG bildet den Minuswert des TOS.

Beispiel: 3 NEG . ergibt -3

MIN und MAX vergleichen den Inhalt des TOS und 2OS und legen das Minimum bzw. Maximum wieder auf den Stapel.

Beispiel: 3 4 MIN . ergibt 3
4 5 MAX . ergibt 5

AND, OR, XOR erlauben logische Verknüpfungen des TOS und 2OS. Sie entsprechen den gleichnamigen Basic-Kommandos.

Beispiel: 1 2 OR . ergibt 3

NOT bildet das Zweier-Komplement des TOS – genau wie der Basic-Befehl gleichen Namens:

Beispiel: 3 NOT . ergibt -4

EMIT gibt den Inhalt des TOS als ASCII-Zeichen aus, vergleichbar der CHR\$-Funktion in Basic.

Beispiel: 7 EMIT gibt einen Beep-Ton aus.

SPACES gibt so viele Leerzeichen aus, wie sie dem TOS-Inhalt entsprechen.

Beispiel: 20 SPACES 42 EMIT gibt nach 20 Leerzeichen einen Stern aus.

CR, LF, CRLF geben einen Wagenrücklauf, ein Line-Feed oder beides zusammen aus:

CR entspricht 13 EMIT

LF entspricht 10 EMIT

CRLF entspricht 10 EMIT 13 EMIT

CLS löscht den Bildschirm.

BEEP gibt einen Beep-Ton aus und entspricht 7 EMIT.

GOTOXY setzt den Cursor entsprechend dem TOS und 2OS.

Beispiel: 1 1 GOTOXY setzt den Cursor in die linke obere Ecke.

BASIC führt einen Rücksprung nach Basic aus.

Wortdefinitionen

Nun zu einem weiteren hervorstechenden Merkmal von Forth: den Wortdefinitionen. Ihnen ist sicher schon aufgefallen, daß man mit dem Grundwortschatz von Forth nicht allzuviel anfangen kann. Dafür kann man aber komplizierte Befehlsfolgen unter einen Namen zusammenfassen und unter diesem Namen aufrufen. Die Definitionen bestehen aus einem Doppelpunkt, einem Leerzeichen, dem Namen der Definition, einem weiteren Leerzeichen, den Befehlen, wiederum einem Leerzeichen und einem Strichpunkt als Kennzeichen für das Definitionsende.

Dabei können die neuen Definitionen wiederum auf alte bereits definierte Worte zurückgreifen. Vielleicht stört Sie der Punkt als Druckbefehl und Sie wollen das Basic-Wort PRINT übernehmen. Nichts leichter als das:

: PRINT . ;

Ab sofort können PRINT und der Punkt gleichwertig nebeneinander verwendet werden.

: KUBIK DUP DUP ** ;

definiert ein Wort KUBIK, mit dem der TOS hoch drei genommen werden kann:

4 KUBIK . ergibt 64

: KUBIKPRINT KUBIK PRINT ;

verwendet beide definierten Worte in einer neuen Definition. Übersetzt wird es vom

Computer als DUP DUP ** .

4 KUBIKPRINT druckt dann sofort den Wert 64 aus.

LIST listet alle vom Benutzer definierten Worte auf.

DEF erlaubt es, die Definition eines Wortes noch einmal anzuzeigen. Dazu muß die Nummer des Wortes ermittelt werden. Sie können ganz einfach beim LIST-Befehl die Worte abzählen, indem Sie bei Null begin-

nen.

Beispiel:

LIST ergibt PRINT KUBIK KUBIKPRINT
PRINT ist dann Definition 1, KUBIK Nr.1,
KUBIKPRINT Nr.2

1 DEF ergibt: KUBIK = DUP DUP **

FORGET ermöglicht es, gezielt Worte aus dem Vokabular wieder zu streichen, wenn sie fehlerhaft sind oder nicht mehr benötigt werden.

Beispiel:

4 KUBIKPRINT ergibt 64

FORGET KUBIKPRINT

4 KUBIKPRINT ergibt KUBIKPRINT?

Als letzte Gruppe gibt es noch sechs „gefährliche“ Kommandos. Diese können nämlich den Computer zum Aussteigen bringen – genau wie ihre Basic-Äquivalente PEEK, POKE, DEEK, DOKE, PRINT PEEK und PRINT DEEK.

C!	entspricht dem Basic-POKE:	255 24576 C!	POKE 24576,255
!	ist ein Doppel-POKE:	&3FAC 24576 !	POKE 24576,&AC:POKE24577,&3F
C@	entspricht dem Basic-PEEK	24576 Ca	TOS=PEEK(24576)
@	entspricht einem Doppel-PEEK	24576 a	TOS=PEEK(24576)+PEEK(24577)*256
C?	ist ein PRINT PEEK	24576 C?	PRINT PEEK(24576)
?	ist ein PRINT Doppel-PEEK	24576 ?	PRINT PEEK(24576)+PEEK(,4577)*256

Überlegen Sie sich die Verwendung dieser Befehle aber gut! Ein falscher Wert – und der Computer ist abgestürzt. Viel Spaß beim Forth-Lernen! Vielleicht finden Sie ja soviel Interesse daran, daß Sie sich einen richtigen Forth-Interpreter zulegen.

Martin Kotulla

```

100 ' *****
*****
110 ' * * *
* * *
120 ' * FORTH/UPN-Simulator * (C)
Martin Kotulla * 2.8.1985 *
130 ' * * *
* * *
140 ' *****
*****
150 '
160 DEF FN1sb(a)=255 AND UNT(a)
170 DEF FNmsb(a)=255 AND INT(a/256)
180 MODE 2: INK 1,0: INK 0,10: BORDER 10
190 LOCATE 14,1: PRINT CHR$(150)+STRING$(
10,154)+CHR$(156)
200 LOCATE 14,2: PRINT CHR$(149)+SPACE$(1
0)+CHR$(149)
210 LOCATE 14,3: PRINT CHR$(147)+STRING$(
10,155)+CHR$(153)
220 LOCATE 1,4: PRINT CHR$(150)+STRING$(3
6,154)+CHR$(156)
230 FOR i=5 TO 24: PRINT CHR$(149);TAB(38
);CHR$(149):NEXT i
240 LOCATE 1,25: PRINT CHR$(147)+STRING$(
36,154)+CHR$(153)
250 WINDOW 42,80,1,25: WINDOW #1,16,23,2,
2
260 CLS #1: PRINT #1,"SF=24576";
270 PRINT "+ - * / Integer-Grundrechena
rten"
280 PRINT "1+ 2+ TOS:=TOS+1, +2"
290 PRINT "SQR TOS quadrieren"
300 PRINT "DUP TOS duplizieren"

```

```

310 PRINT "DROP TOS vergessen"
320 PRINT "SWAP TOS und 2OS austausc
hen"
330 PRINT "SWPB MSB & LSB des TOS ta
uschen"
340 PRINT "ABS NEG TOS:=abs(TOS) TOS:=
-TOS"
350 PRINT "MIN MAX TOS:=min(TOS,2OS) bz
w. max"
360 PRINT "AND X-OR TOS:=TOS and 2OS (xo
r,or)"
370 PRINT "NOT TOS:=not TOS"
380 PRINT "EMIT TOS als ASCII-Zeiche
n drucken"
390 PRINT "SPACES TOS* Leerzeichen dru
cken"
400 PRINT "CR LF Wagenruecklauf, Line
-Feed"
410 PRINT "CRLF Wagenruecklauf & Lin
e-Feed"
420 PRINT "CLS Bildschirm loeschen"
430 PRINT "BEEP Beep-Ton"
440 PRINT "GOTOXY Cursor nach TOS/2OS
setzen"
450 PRINT ": ... ; Wort definieren"
460 PRINT "LIST Definierte Worte lis
ten"
470 PRINT "DEF Definition eines Wor
ts listen"
480 PRINT "FORGET Wortdefinition verge
ssen"
490 PRINT "BASIC Ruecksprung nach Bas
ic"
500 PRINT "C! C@ C? POKE, PEEK, PRINT PE

```

weiter Seite 46

Hobby-Tronic Dortmund mit 2. Fachausstellung für Computer, Software und Zubehör

In der Zeit vom 23. – 27. April fand in Dortmund die 9. Ausstellung für Funk- und Hobby-Elektronik statt. Angegliedert an diese 9. Fachausstellung war wie bereits im letzten Jahr auch die 2. Computer-Schau. Hier konnte der interessierte Fachbesucher in einer gesonderten Messehalle ein großes Angebot an Computern, Software und Zubehör bestaunen.

Große Zahl an Anbietern

Eine große Zahl von Fachhändlern und Anbietern hatte in der Messehalle ihre Stände aufgebaut und zeigten einem interessierten Publikum die neuesten Zubehörteile im Bereich der Mikroelektronik. Vom Joystick bis hin zum einfachen Adapter war für fast jeden Homecomputer das zu bekommen, was der User für den Umgang mit seinem System benötigt. Dabei war es von Vorteil, wenn man sich zu Beginn des Besuches erst einmal einen Überblick vom Angebot verschaffte, denn die Vielzahl der Anbieter garantierte oft für das gleiche Produkt einen unterschiedlichen Preis. Wer hier erst verglichen und dann gekauft hat, der hat sicherlich seinen Geldbeutel etwas entlasten können.

Kaufhäuser nutzen ihre Markchancen

Nachdem bereits im letzten Jahr einige Kaufhäuser mit Verkaufsständen vertreten waren, hat jedes größere Warenhaus in der Umgebung von Dortmund die Gunst der Stunde genutzt und mit einem breiten und vielfältigen Produktangebot die Kunden gelockt. Die Angebotspalette dieser Kaufhäuser wird ständig erweitert, denn auch hier hat man längst entdeckt, daß der Computer immer mehr Aufmerksamkeit erlangt und der Anteil der Freizeitbeschäftigung mit dem Computer ständig steigt. Auch das Verkaufspersonal steht heute schon mit Rat und Tat zur Seite und gibt Fachinformationen weiter, die man noch vor kurzer Zeit nicht erwarten konnte, da nicht alle über genügend Erfahrungen in der Mikroelektronik verfügten.

Das Softwareangebot reist nicht ab

Neben dem Hardware-Vorstellungen war der zweite große Themenbereich dieser Ausstellung das riesige und nicht mehr endenwollende Software-Angebot. An jedem Stand waren Computer aufgebaut, an denen der Besucher die neuesten Spiele, Adventures oder Anwendungsprogramme testen konnte. Man mußte schon viel Zeit mitbringen, wollte man an jedem Stand alles das testen, was angeboten wurde. Wir waren der Meinung, daß dieser Teil der Ausstellung wohl die meisten der Besucher reizte. Nach Auskunft der Händler und der großen Software-Vertriebsfirmen in Deutschland ist mit einem ablaufenden des z.Z. herrschenden Booms nicht zu rechnen. Im Gegenteil, die Zahl der Software-Neuvorstellung innerhalb eines Monats z.B. nimmt laufend zu und die Neugründung vieler Software-Häuser im In- und Ausland läßt schon erahnen, daß hier noch längst nicht das Ziel erreicht ist und in Fachkreisen wird deshalb auch mit steigenden Verkaufszahlen auf dem europäischen Markt gerechnet.

Regionale oder internationale Computer-Schau?

Leider ist es z.Z. noch nicht so, daß die Hobbytronic das Flair einer internationalen Fachausstellung besitzt. Aber bei genauerem hinschauen, konnte man schon feststellen, das Anbieter aus dem gesamten Bundesgebiet vertreten waren. Lediglich bei der Computerschau fiel auf, das es noch eine regionale Schau der umliegen-



den Firmen war. Zwar war hin und wieder ein Messestand zu sehen, der nicht direkt aus der Umgebung Dortmunds kam, aber im Großen und Ganzen waren doch eine Vielzahl von einheimischen Firmen mit ihren Ständen vertreten. Nach Auskunft der Messeleitung ist man jedoch bestrebt, diese Messe auszuweiten und versucht für die nächste Computerschau die Ausstellungsfläche zu vergrößern und die Händlerzahl zu steigern.

Fazit:

Insgesamt betrachtet, kann man die Messe als einen Erfolg für Händler und Besucher ansehen. Durch das große Angebot an Hard- und Software wurde dem User ein Überblick über das gesamte Zubehör-Sortiment im Bereich der Mikroelektronik gezeigt. Bei Preisvergleichen konnte er beim Kauf von verschiedenen Artikeln erhebliches an Geld sparen. Die Besucherzahlen während der Messetagen lassen auf eine positive Resonanz beim Anwenden schließen. Es bleibt zu sagen, daß man versuchen sollte, dieser Messe noch mehr an internationalem Flair zu verleihen.

Amstrad kauft die weltweiten Vertriebsrechte für Sinclair

Vertrieb in Deutschland über Schneider AG

Die britische Firma Amstrad Consumer Electronics plc gab am 07. 04. 86 bekannt, daß sie die weltweiten Vertriebs- und Produktionsrechte aller Sinclair Computer und Computerprodukte sowie das Markenzeichen Sinclair und das hiermit verbundene Know-How von der Firma Sinclair Research Limited übernommen hat.

Für diese Rechte wurden von Amstrad 5 Mill. Pfund Sterling bezahlt. Auch der Verkauf der vorhandenen Lagerbestände sowie einige bestehende Auftragsverpflichtungen werden von Amstrad übernommen.

Amstrad wird die Sinclair Computerprodukte weiterhin unter der Marke Sinclair vermarkten. Der Vertrieb in Deutschland,

Österreich, Schweiz und den Benelux-Staaten wird von der Schneider AG in Türkheim/Unterallgäu übernommen. Der Verkauf der Sinclair Computern wird parallel zum Verkauf der Schneider Computer laufen, wobei die Sinclair Geräte den Unterhaltungsbereich und die Schneider Rechner den gehobenen Bereich der Heimcomputer sowie den Bereich der Geschäftscomputer abdecken werden.

Die Übernahme erfolgt fast genau zwei Jahre nach dem Einstieg der Firmen Amstrad und Schneider in den Computermarkt, festigt die weltweite Stellung und erlaubt es den beiden Unternehmen, in einen speziellen Marktbereich einzudringen, der traditionsgemäß von den Marken Commodore und Sinclair beherrscht wur-

de.

Die Sinclair Produkte waren für technische Neuerungen bekannt. Die Stärken von Amstrad und Schneider liegen im Marketing, dem Vertriebsnetz und der anerkannt guten Qualität der Computer und Peripheriegeräte. Es existiert bereits ein flächendeckendes Händlernetz, über das die Sinclair Produkte ebenso vermarktet werden können wie schon die bisherige Produktpalette.

Amstrad wird auch weiterhin mit der neuorganisierten Sinclair Research Limited sowie mit Sir Clive Sinclair in Hinsicht auf eventuelle Neuentwicklungen im Dialog bleiben, die dann gegebenenfalls von Amstrad und Schneider vermarktet werden.

Probierwochen: Selbst testen, was besser ist!

An die Systeme C-64/C-128 kann man einen Standard-Drucker über ein Hardware-Interface oder mit einem einfachen Kabel an den Userport anschließen. Unser Interface ist 100% kompatibel und problemlos, das Userport-Kabel funktioniert nur eingeschränkt (nur in Verbindung mit bestimmten Programmen), blockiert den

User-Port für andere Erweiterungen, ist aber billig. Wir möchten jedem Kunden die Gelegenheit geben, zu testen, welche Lösung besser ist:

Wir bieten ab sofort ein Userport-Kabel (ohne Software) für 39,- DM inkl. MwSt. an. Innerhalb von 2 Monaten kann der Kunde dieses Kabel (bei uns oder bei ei-

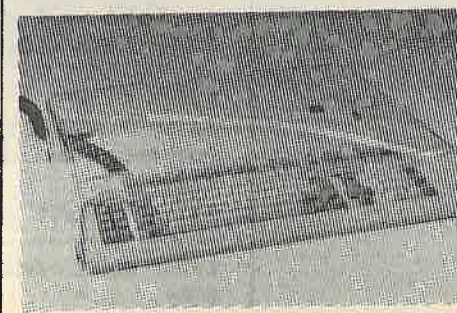
nem teilnehmenden Händler) jederzeit ohne Angabe von Gründen gegen eines unserer vollkompatiblen Hardware-Interfaces tauschen: Bei Rückgabe eines Userport-Kabels berechnen wir für unser Interface 92000/G nur noch 168,- DM (sonst 198,- DM), für den 8KByte-Typ 92008/G nur noch 248,- DM (sonst 278,- DM) – der Kaufpreis des Kabels wird also bis auf 9,- DM Handlingkosten voll erstattet! Wiesemann Computertechnik Winchenbachstraße 3-5 5600 Wuppertal 2

DAZU PRODUKTVERTRIEB GMBH,
Hans-Henny-Jahnn-Weg 41-45,
2000 Hamburg 76,
stellt ein neues Produkt vor.

Für die Zentraleinheit und den Bildschirm der Personal Computer ist die BASIS ein Untersatz, der die Tastatur aufnimmt und ein Minimum an Platz auf der Arbeitsfläche beansprucht.

Die Abmessungen, Höhe 95 mm, Breite 538 mm, Tiefe 420 mm, sind so proportioniert, daß die PC Zentraleinheiten der verschiedenen Farbfabrikate für das Produkt geeignet ist.

Bringen Sie Ihr System platzsparend unter



Die Belastbarkeit ist durch die drei Standflächen und die vordere Aufkantung gewährleistet. In der Rückwand befinden sich links und rechts Öffnungen zur Durchführung der Tastaturkabel. Sechs Gummiprofile geben der BASIS auf allen Flächen einen festen Halt.

Die DAZU BASIS ist aus dem klaren und neutralen Material Plexiglas und wird in Funktion und Design allen Ansprüchen gerecht.

Das Produkt ist im Bürofachhandel zum Preis von 139,00 DM zuzüglich MwSt. zu beziehen.


```

EK"
510 PRINT "! @ ? DOKE, DEEK, PRINT DE
EK"
520 WINDOW 3,36,5,24:MEMORY 24575
530 stack=24576:DIM array$(128),helpary$(
128),word$(128),vocab$(128)
540 ON ERROR GOTO 2090 ' Fehlerbehandlun
g
550 PRINT:LINE INPUT a$:a$=UPPER$(a$)
560 IF LEN(a$)>253 THEN PRINT CHR$(7);"*
Eingabe zu lang!":GOTO 550
570 IF ASC(a$+CHR$(0))=32 THEN a$=MID$(a
$,2):GOTO 570 ELSE a$=a$+" "
580 in=INSTR(1,a$," "):IF in THEN a$=LE
FT$(a$,in-1)+MID$(a$,in+1):GOTO 580
590 IF ASC(a$+CHR$(0))<32 THEN PRINT a$;
:GOTO 550
600 IF LEFT$(a$,1)<>" " THEN 650 ' ****
Wort definieren
610 posi=INSTR(3,a$," ")-3:word$(vocab)=
MID$(a$,3,posi)
620 vocab$(vocab)=MID$(a$,posi+4)
630 vocab$(vocab)=LEFT$(vocab$(vocab),LE
N(vocab$(vocab))-2)
640 vocab=vocab+1:GOTO 550
650 count=0
660 ERASE array$,helpary$:DIM array$(255
),helpary$(128)
670 in=INSTR(1,a$," ")
680 IF in THEN array$(count)=LEFT$(a$,in
-1):count=count+1
690 IF in THEN a$=MID$(a$,in+1):GOTO 670
700 ' KOMMANDOS AUSFUEHREN =====
710 loop=0
720 a$=array$(loop)
730 IF a$="." THEN 1200 '
Drucke Top-of-Stack
740 IF a$="+" THEN 1220 '
Addition
750 IF a$="-" THEN 1230 '
Subtraktion
760 IF a$="*" THEN 1240 '
Multiplikation
770 IF a$="/" THEN 1250 '
Division
780 IF a$="DUP" THEN 1290 '
DUPLICATE TOS
790 IF a$="DROP" THEN 1300 '
DROP TOS
800 IF a$="SWAP" THEN 1310 '
SWAP TOS & 20S
810 IF a$="EMIT" THEN 1330 '
ASCII-Zeichen des TOS ausgeben
820 IF a$="SWPB" THEN 1340 '
Swap MSByte & LSByte des TOS
830 IF a$="1+" THEN 1370 '
TOS:=TOS+1
840 IF a$="2+" THEN 1380 '
TOS:=TOS+2
850 IF a$="SQR" THEN 1400 '
TOS:=TOS^2
860 IF a$="ABS" THEN 1410 '

```

```

TOS:=abs(TOS)
870 IF a$="NEG" THEN 1420 '
TOS:=-TOS
880 IF a$="MIN" THEN 1440 '
TOS:=min(TOS,20S)
890 IF a$="MAX" THEN 1450 '
TOS:=max(TOS,20S)
900 IF a$="AND" THEN 1470 '
TOS:=TOS and 20S
910 IF a$="OR" THEN 1480 '
TOS:=TOS or 20S
920 IF a$="XOR" THEN 1490 '
TOS:=TOS xor 20S
930 IF a$="NOT" THEN 1500 '
TOS:=not TOS
940 IF a$="LIST" THEN 1640 '
Definierte Worte listen
950 IF a$="DEF" THEN 1660 '
Definition eines Worts listen
960 IF a$="FORGET" THEN 1680 '
Wortdefinition vergessen
970 IF a$="SPACES" THEN 1740 '
TOS* Leerzeichen ausgeben
980 IF a$="CR" THEN 1760 '
Wagenruecklauf
990 IF a$="LF" THEN 1770 '
Line-Feed
1000 IF a$="CRLF" THEN 1780 '
CR & Line-Feed
1010 IF a$="CLS" THEN 1790 '
Bildschirm loeschen
1020 IF a$="BEEP" THEN 1800 '
Beep-Ton ausgeben
1030 IF a$="GOTOXY" THEN 1810 '
Cursor nach TOS/20S setzen
1040 IF a$="BASIC" THEN 1880 '
Ruecksprung nach Basic
1050 IF a$="C@" THEN 1910 '
TOS:=peek(TOS)
1060 IF a$="@" THEN 1920 '
TOS:=deek(TOS)
1070 IF a$="C!" THEN 1930 '
poke TOS,20S
1080 IF a$="!" THEN 1940 '
doke TOS,20S
1090 IF a$="C?" THEN 1960 '
print peek(TOS)
1100 IF a$="?" THEN 1970 '
print deek(TOS)
1110 IF ASC(a$+CHR$(0))>64 AND ASC(a$+CH
R$(0))<91 THEN 1520 ' Wort uebersetzen
1120 IF VAL(a$)<>0 OR a$="0" THEN 1180 '
Zahl auf den Stack
1130 loop=loop+1:IF loop<=count THEN 720
ELSE 550
1140 '
1150 ' BEFEHLE =====
1160 '
1170 ' Zahl auf den Stack schreiben ----
1180 zahl=INT(VAL(a$)):GOSUB 2020:GOTO 1
130

```

```

1190 ' Drucke Top-of-Stack-Element ----
1200 GOSUB 2060:PRINT zahl;:GOTO 1130
1210 ' Grundrechenarten + - * / ----
1220 GOSUB 2060:zahl2=zahl:GOSUB 2060:za
hl=zahl+zahl2:GOSUB 2020:GOTO 1130
1230 GOSUB 2060:zahl2=zahl:GOSUB 2060:za
hl=zahl-zahl2:GOSUB 2020:GOTO 1130
1240 GOSUB 2060:zahl2=zahl:GOSUB 2060:za
hl=zahl*zahl2:GOSUB 2020:GOTO 1130
1250 GOSUB 2060:zahl2=zahl:GOSUB 2060
1260 IF zahl2=0 THEN PRINT CHR$(7);"* Di
vision durch Null!":GOTO 1130
1270 zahl=INT(zahl/zahl2):GOSUB 2020:GOT
O 1130
1280 ' DUP DROP SWAP EMIT SWPB ----
1290 GOSUB 2060:GOSUB 2020:GOSUB 2020:GO
TO 1130
1300 GOSUB 2060:GOTO 1130
1310 GOSUB 2060:zahl2=zahl:GOSUB 2060:za
hl3=zahl:zahl=zahl2:GOSUB 2020
1320 zahl=zahl3:GOSUB 2020:GOTO 1130
1330 GOSUB 2060:PRINT CHR$(zahl AND 255)
;:GOTO 1130
1340 ms=PEEK(stack-1):ls=PEEK(stack-2):P
OKE stack-1,ls
1350 POKE stack-2,ms:GOTO 1130
1360 ' 1+ 2+ ----
1370 GOSUB 2050:zahl=zahl+1:GOSUB 2010:G
OTO 1130
1380 GOSUB 2050:zahl=zahl+2:GOSUB 2010:G
OTO 1130
1390 ' SQR ABS NEG ----
1400 GOSUB 2050:zahl=zahl*zahl:GOSUB 201
0:GOTO 1130
1410 GOSUB 2050:zahl=ABS(zahl):GOSUB 201
0:GOTO 1130
1420 GOSUB 2050:zahl=-zahl:GOSUB 2010:G
OTO 1130
1430 ' MIN und MAX: Minimum und Maximum
des TOS und 20S ----
1440 GOSUB 2050:zahl2=zahl:GOSUB 2050:za
hl=MIN(zahl2,zahl):GOSUB 2010:GOTO 1130
1450 GOSUB 2050:zahl2=zahl:GOSUB 2050:za
hl=MAX(zahl2,zahl):GOSUB 2010:GOTO 1130
1460 ' AND OR XOR NOT - logische Verknue
pfungen
1470 GOSUB 2050:zahl2=zahl:GOSUB 2050:za
hl=zahl AND zahl2:GOSUB 2010:GOTO 1130
1480 GOSUB 2050:zahl2=zahl:GOSUB 2050:za
hl=zahl OR zahl2:GOSUB 2010:GOTO 1130
1490 GOSUB 2050:zahl2=zahl:GOSUB 2050:za
hl=zahl XOR zahl2:GOSUB 2010:GOTO 1130
1500 GOSUB 2050:zahl=NOT zahl:GOSUB 2010
:GOTO 1130
1510 ' Wort suchen und interpretieren --
1520 FOR i=0 TO vocab:IF word$(i)<>a$ TH
EN NEXT i:GOTO 1620

```

```

1530 array$(loop)="" :lp2=loop
1540 k=0:FOR j=loop+1 TO count
1550 helpary$(k)=array$(j):array$(j)="" :
k=k+1:NEXT j
1560 v$=vocab$(i)
1570 in=INSTR(1,v$," ")
1580 IF in THEN array$(lp2)=LEFT$(v$,in-
1):count=count+1:lp2=lp2+1
1590 IF in THEN v$=MID$(v$,in+1):GOTO 15
70
1600 FOR m=0 TO k:array$(m+lp2+1)=helpar
y$(m):NEXT m
1610 loop=loop-1:GOTO 1130
1620 PRINT CHR$(7);a$;"?":GOTO 550
1630 ' LIST: Definierte Worte listen ----
1640 FOR i=0 TO vocab:PRINT word$(i);" "
;:NEXT i:GOTO 1130
1650 ' DEF: Wortdefinition listen ----
1660 GOSUB 2050:PRINT word$(zahl AND 127
);" = ";vocab$(zahl AND 127):GOTO 1130
1670 ' FORGET: Wortdefinition loeschen -
----
1680 loop=loop+1:v$=array$(loop)
1690 FOR i=0 TO vocab:IF v$<>word$(i) TH
EN NEXT i:GOTO 1720
1700 FOR m=i TO vocab:word$(m)=word$(m+1
):vocab$(m)=vocab$(m+1)
1710 NEXT m:word$(m)="" :vocab$(m)="" :voc
ab=vocab-1:GOTO 1130
1720 PRINT CHR$(7);v$;"?":GOTO 550
1730 ' SPACES: TOS* Leerzeichen ausdruck
en ----
1740 GOSUB 2060:PRINT SPACE$(zahl);:GOTO
1130
1750 ' CR, LF, CRLF und CLS: Bildschirmb
efehle ----
1760 PRINT CHR$(13);:GOTO 1130
1770 PRINT CHR$(10);:GOTO 1130
1780 PRINT CHR$(10);CHR$(13);:GOTO 1130
1790 CLS:GOTO 1130
1800 PRINT CHR$(7);:GOTO 1130
1810 GOSUB 2060:zahl2=zahl:GOSUB 2060
1820 IF zahl2=0 OR zahl=0 THEN 1850
1830 zahl=zahl AND 255:zahl2=zahl2 AND 2
55
1840 IF zahl2=0 OR zahl=0 THEN 1850 ELSE
1860
1850 PRINT CHR$(7);"* Unzulaessige Curso
rposition!":GOTO 1130
1860 LOCATE zahl2,zahl:GOTO 1130
1870 ' BASIC: Ruecksprung nach Basic ----
1880 CALL &BB00:CALL &BB4E:CALL &BBBA '
Tastatur, Text und Grafik zuruecksetzen
1890 CALL &BBFF:PEN 1:END ' SCR SET INIT
IALIZE aufrufen
1900 ' PEEK & POKE: C@ @ C! ! C? ? ----
1910 GOSUB 2060:zahl=PEEK(zahl):GOSUB 20
20:GOTO 1130
1920 GOSUB 2060:zahl=PEEK(zahl)+PEEK(zah

```



```

1+1)*256:GOSUB 2020:GOTO 1130
1930 GOSUB 2060:zahl2=zahl:GOSUB 2060:PO
KE zahl2,FNlsb(zahl):GOTO 1130
1940 GOSUB 2060:zahl2=zahl:GOSUB 2060:PO
KE zahl2,FNlsb(zahl)
1950 POKE zahl2+1,FNmsb(zahl):GOTO 1130
1960 GOSUB 2060:PRINT PEEK(zahl):GOTO 1
130
1970 GOSUB 2060:PRINT PEEK(zahl)+PEEK(z
ahl+1)*256:GOTO 1130
1980 '
1990 ' UNTERPROGRAMME =====
=====
2000 '
2010 ' - PUSH zahl
2020 IF zahl>65535 OR zahl<-32768 THEN P
RINT CHR$(7);"* Overflow!":RETURN

```

CHECK V1:PRUEFSUMMEN:

ZEILENNR.: SUMMEN

```

100- 110: 46030
120- 130: 70124
140- 150: 39352
160- 170: 83279
180- 190: 55616
200- 210: 85837
220- 230: 110414
240- 250: 61637
260- 270: 75324
280- 290: 46759
300- 310: 58989
320- 330: 108535
340- 350: 99864
360- 370: 71410
380- 390: 65797
400- 410: 120397
420- 430: 58761
440- 450: 94320
460- 470: 68526

```

```

480- 490: 116369
500- 510: 70751
520- 530: 52101
540- 550: 62973
560- 570: 76284
580- 590: 58869
600- 610: 30530
620- 630: 116928
640- 650: 32757
660- 670: 59340
680- 690: 54626
700- 710: 54364
720- 730: 51936
740- 750: 112875
760- 770: 64950
780- 790: 101626
800- 810: 71135
820- 830: 103858
840- 850: 91766
860- 870: 103916
880- 890: 18614
900- 910: 65653

```

```

920- 930: 64385
940- 950: 89165
960- 970: 19200
980- 990: 66359
1000- 1010: 104468
1020- 1030: 84846
1040- 1050: 112840
1060- 1070: 114578
1080- 1090: 58210
1100- 1110: 64810
1120- 1130: 71246
1140- 1150: 42984
1160- 1170: 19045
1180- 1190: 54840
1200- 1210: 19334
1220- 1230: 72435
1240- 1250: 65162
1260- 1270: 104732
1280- 1290: 19367
1300- 1310: 64073
1320- 1330: 66387
1340- 1350: 53131

```

```

1360- 1370: 110345
1380- 1390: 111888
1400- 1410: 53602
1420- 1430: 84344
1440- 1450: 102695
1460- 1470: 76864
1480- 1490: 77232
1500- 1510: 73762
1520- 1530: 100753
1540- 1550: 105098
1560- 1570: 28529
1580- 1590: 68516
1600- 1610: 53926
1620- 1630: 34200
1640- 1650: 42466
1660- 1670: 39179
1680- 1690: 56773
1700- 1710: 75948
1720- 1730: 44239
1740- 1750: 54417
1760- 1770: 11923

```

```

1780- 1790: 14621
1800- 1810: 33915
1820- 1830: 63846
1840- 1850: 70291
1860- 1870: 40191
1880- 1890: 45851
1900- 1910: 111215
1920- 1930: 67579
1940- 1950: 106825
1960- 1970: 33867
1980- 1990: 44287
2000- 2010: 10299
2020- 2030: 27557
2040- 2050: 57582
2060- 2070: 74864
2080- 2090: 56381
2100- 2110: 92453

```

GESAMTCHECKSUMME: 32904

```

2030 POKE stack,FNlsb(zahl):POKE stack+1
,FNmsb(zahl):stack=stack+2
2040 CLS #1:PRINT #1,"SP=";MID$(STR$(sta
ck),2);:RETURN
2050 ' - POP zahl
2060 stack=stack-2:zahl=UNT(PEEK(stack)+
PEEK(stack+1)*256)
2070 IF stack<24576 THEN PRINT CHR$(7);"
* Stack leer!":stack=stack+2:zahl=0
2080 CLS #1:PRINT #1,"SP=";MID$(STR$(sta
ck),2);:RETURN
2090 ' Fehlerbehandlung -----

```

```

2100 PRINT CHR$(7);"* Fehler";ERR;"in";E
RL:RESUME 550
2110 END ' *****
*****

```

TIPS & TRICKS für den CPC 6128

Der CPC-6128 Tip

Welcher CPC-6128-Besitzer hat sich noch nicht überlegt, wie er zu seinem Dateiverwaltungsprogramm oder ähnlichem, die neuen RSX-Befehle zu den zusätzlichen 64k, nutzen oder noch besser einbauen könnte?

Aber meistens scheiterte der Versuch weil es dem Benutzer zu umständlich war, immer vor Programmbeginn das Programm „Bankman“ zu Starten. Und zu allem Übel ist der Basic-Teil noch geschützt.

Um nun allen denjenigen zu helfen, die sich mit dem BANKSWITCHING nicht besonders auskennen, ist sie hier eine entsprechende Hilfe.

1. Das Listing von Bankman.bas mit ausführlicher Beschreibung:

```

10 ON ERROR GOTO 0
20 '
30 mcode=HIMEM*15:MEMORY mcode-1:'Speicher für M-Code reservieren
40 '
50 FOR p=mcode TO mcode+15
60 READ p$:POKE p,VAL("&"+p$):'Datas in HEX wandeln u. in Speicher POKEN
70 NEXT
80 '
90 DATA e,ff,cd,15,b9,eb,dd,66,1,dd,6e,0,73,23,72,c9
100 '
110 '
120 a%=0:CaLL mcode,@a%:'dem MCode die Variable a% z. verarbeiten übergeben
130 MEMORY mcode+15:'MCode wieder löschen
140 '
150 version=(a%/256)+356*(a%MOD 256):'CPC Version durch a% ermitteln
160 IF version<&102 then 320:'Bei Falscher Version zu 320

```

```

170 '
180 'Eigentliche Laderoutine
190 '
200 mcode=HIMEM-&524:'Freien Speicher für Bankman.bin reservieren
210 ON ERROR GOTO 310:'Bei memoryfehler zu 310
220 IF mcode<=&7fff THEN ERROR 7:'Wenn res.Speicher unter &8000 MEMORY FULL
230 '
240 MEMORY mcode-1:LOAD "bankman.bin",mcode:'Bankman.bin in freien Speicher
250 CALL mcode:'RSX-Befehle aktivieren
260 IBANKOPEN,0
270 END:' Das danachfolgende Programm müßte dann bei 330 anfangen
280 'und 270 müßte lauten 270 GOTO 330
290 'Ende der Laderoutine
300 '
310 PRINT"Memory FEHLER !!!":END
320 PRINT"Falsche CPC-Version":end

```

2.) Nun muß nur noch Bankman.bas abgeSAVED werden und Bankman.bin auf die Diskette kopiert werden, wo auch Bankman.bas ist. (J.M.)

Schneider CPC 464: Das Speichern von Programmausschnitten

In der Praxis kann es immer wieder vorkommen, daß man nicht ein ganzes BASIC-Programm, sondern nur einen Ausschnitt davon abspeichern möchte. Bisher hat man diese Möglichkeit nicht gehabt, aber die folgende Routine ermöglicht jeden Anwender, daß Abspeichern einzelner Programmabschnitte. Mit diesem Programm läßt sich das bewerkstelligen.

Beim Schneider CPC 464 gibt es keinen Befehl, mit dem man aus einem längeren Programm einen Teil, etwa eine Subroutine, getrennt speichern kann, sowie umgekehrt der MERGE-Befehl funktioniert. Wenn man eine Subroutine von einem Programm in ein anderes kopieren möchte, müssen die nicht benötigten Programmzeilen manuell gelöscht werden. Das Utility, löst das Problem, denn beliebige Teile eines Programmes können auf Cassette oder Diskette abgespeichert werden! Das heißt, bereits in anderen Programmen geprüfte Subroutinen können ohne weiteres in ein neues Programm übernommen werden. Wer einen Assembler hat, kann das Programm direkt eingeben, sonst muß man das BASIC-Programm abtippen und laufen lassen. Nachdem dann das BINÄR-File abgespeichert ist, wird das BASIC-Programm nicht mehr benötigt.

Will man nun aus einem Programm einen bestimmten Zeilenbereich abspeichern, so läßt man das Utility mit 'MEMORY & ASFF' und 'LOAD „S-SAVE.BIN“' dazu und initialisiert es mit 'CALL &A600'. Jetzt kann man den Zeilenbereich mit dem Befehl '↑(erste Zeilennummer),(zweite Zeilennummer),,(Name)'' abspeichern.

Programmbeschreibung:

Da die Art der RSX-Erweiterung nicht sonderlich übersichtlich ist und sie sich durch die umständliche Stringübergabe auszeichnet, bauen wir den Befehl ein, indem wir den 'Patch für Befehl ausführen' auf diese Routine umbiegen. Dort wird getestet, ob der Hochpfeil eingegeben wurde. Wenn nicht, erfolgt der Rücksprung. (Bei

dem Text in Zeile 70 kann an der Stelle von &F8 auch jeder andere Tokenwert über &DD eingetragen werden.) In Zeile 90 wird die letzte Rücksprungadresse vom Stack geholt. Der Programmzeiger wird erhöht und die erste Zeilennummer geholt. Der Zeiger wird gerettet und in 130 wird versucht, die Zeilenadresse der Zeilennummer zu holen. Existiert die Zeile nicht, so wird 'LINE DOES NOT EXIST' ausgegeben und ins BASIC zurückgekehrt. Ansonsten wird die Zeilenadresse abgespeichert und getestet, ob ein Komma folgt. Ist die Bedingung nicht erfüllt, wird wieder zur Fehlerausgabe verzweigt. Ab 210 wird die zweite Zeilennummer geholt und ebenfalls umgerechnet. Dann wird noch die Zeilenlänge (in BC) addiert und in 250 abgespeichert. Ab Zeile 260 beginnt der wichtigste Teil des Programmes. Zunächst werden die Pointer auf Programmanfang und -ende gesichert und dann die vorher berechneten Werte eingetragen. Nun wird die ROM-Routine für 'SAVE' angesprungen und ganz normal ausgeführt, abgespeichert wird aber nur der von uns gewünschte Programmteil.

Im folgenden werden dann die alten Werte wieder hergestellt, der Programmzeiger vom Stack geholt und ins BASIC zurückgesprungen.

Bemerkenswert ist noch, daß zu dieser Zeit das obere ROM freigeschaltet ist.

Programmlisting linke Seite

Wenn auch Sie Tips und Tricks für die Schneider-Systeme kennen und dies an unsere Leser weitergeben möchten, dann senden Sie uns diese umgehend an:

Tronic-Verlag GmbH
- Programmentwicklung -
Postfach 870
3440 Eschwege

Starcommand

Diese kurze Überschrift gibt einen Hinweis über die Aufgabe des Raumschiffkommandanten, der dringend ein Gegenmittel zur Bekämpfung der Raumkrankheit zu den einzelnen, weitentfernten Raumstationen fliegen muß.

Dieser Flug ist jedoch mit zahlreichen Gefahren versehen. So müssen Sie u.a. durch Meteoritenfelder, vorbei an gefährlichen Strahlungen und über Minen fliegen.

Zum Schutz steht Ihnen ein Schutzschirm zur Verfügung, der bei jeder Kollision aktiviert wird. Das Schutzschild oder die Laserskanone, die Sie zur Abwehr der Meteoriten und Minen verwenden können, benötigen jedoch eine gewisse Energie.

Diese Energie wird links durch einen Balken angezeigt. Jede Kollision oder jeder Schuß verringert diese vorhandene Energie. Ist Ihre Energie aufgebraucht, wird das Spiel beendet.

```
10 '
20 ' * * * * SUPER-SAVE * * * *
30 '
40 'ABSPEICHERN VON BELIEBIGEN ZEILENBEREICHEN
50 '
60 'MIT DEM BEFEHL :
70 '
80 '^ [ERSTE ZEILE] , [LETZTE ZEILE] , "[PROGRAMM-NAME]"
90 '
100 DATA 3E,C3,32,7,AC,21,C,A6,22,8
110 DATA AC,C9,FE,F0,C0,D1,23,CD,91,CE
120 DATA E5,CD,9A,E7,2B,22,5B,A6,E1,7E
130 DATA FE,2C,C2,C6,DD,23,CD,91,CE,E5
140 DATA CD,9A,E7,9,22,5D,A6,2A,81,AE
150 DATA 22,5F,A6,2A,83,AE,22,61,A6,2A
160 DATA 5B,A6,22,81,AE,2A,5D,A6,22,83
170 DATA AE,E1,23,CD,9,EC,E5,2A,5F,A6
180 DATA 22,81,AE,2A,61,A6,22,83,AE,E1
190 DATA C9
200 FOR I=42496 TO 42586:READ A$:POKE I,VAL("&"*A$):NEXT I
210 SAVE "S-SAVE.BIN",B,42496,90
220 END
```



Bei einer Kollision mit einer Mine ist ebenfalls das Spielende erreicht. Auffüllen können Sie Ihren Energievorrat durch Einsammeln von Tanks, die hin und wieder auf dem Bildschirm erscheinen. Am Ende einer Runde müssen Sie auf einer Raumstation landen. Für die hier noch vorhandene Energie erhalten Sie dann Bonuspunkte gutgeschrieben.

Steuern können Sie Ihr Raumschiff mit dem Joystick oder mit den Tasten Cursor rechts, Cursor links und Space.

```
100 REM *****
102 REM *
104 REM * Winklbauer Markus
106 REM * Oesterreich
108 REM *
110 REM * Starcommand
112 REM * TEIL 1
114 REM *
116 REM *****
118 REM
120 REM
122 REM
124 REM *****
126 REM *
128 REM * Titelbild
130 REM *
132 REM *****
134 REM
136 MODE 0
138 REM
140 SYMBOL AFTER 149
142 SYMBOL 151,120,220,166,222,174,202,18,44
144 SYMBOL 200,1,3,6,14,27,51,125,0
146 SYMBOL 201,128,192,96,112,216,204,190,0
148 SYMBOL 202,1,1,2,2,7,5,15,1
150 SYMBOL 203,128,128,64,64,224,160,240,128
152 SYMBOL 204,3,14,53,78,123,87,62,13
154 SYMBOL 205,16,168,126,218,125,232,208,126
156 SYMBOL 206,31,63,120,126,126,126,63,31
158 SYMBOL 207,248,252,62,254,254,190,252,248
160 SYMBOL 208,3,15,21,21,15,3,31,3
162 SYMBOL 209,192,240,168,168,240,192,248,192
164 f=3
166 FOR x=3 TO 15:INK x,26:NEXT
168 INK 1,0:INK 0,26:PAPER 0:PEN 1:BORDE R 26:CLS
170 INK 2,0,26:PEN 2:LOCATE 3,25:PRINT"P lease Waite ...."
172 k=3
174 PLOT 146,250,0
176 FOR g=1 TO 2
178 FOR x=1 TO 50:PLOTR 1,2,k:DRAWR 8,0:PLOTR -8,0
180 k=k+1:IF k=15 THEN k=3
182 NEXT
184 FOR x=1 TO 50:PLOTR 0,-2,k:DRAWR 8,0:PLOTR -8,0
186 k=k+1:IF k=15 THEN k=3
188 NEXT
190 NEXT
192 PLOTR 34,0,0:FOR x=1 TO 20:PLOTR 0,1,k:DRAWR 20,0:PLOTR -20,0
194 k=k+1:IF k=15 THEN k=3
196 NEXT
198 PLOT 350,350,0
200 FOR g=1 TO 2
202 FOR x=1 TO 50:PLOTR 0,-2,k:DRAWR 8,0
```

```
:PLOTR -8,0
204 k=k+1:IF k=15 THEN k=3
206 NEXT
208 FOR x=1 TO 50:PLOTR 1,2,k:DRAWR 8,0:PLOTR -8,0
210 k=k+1:IF k=15 THEN k=3
212 NEXT
214 NEXT
216 PLOTR 10,-100,0
218 FOR x=1 TO 20:PLOTR 0,1,k:DRAWR 20,0:PLOTR -20,0
220 k=k+1:IF k=15 THEN k=3
222 NEXT
224 GOSUB 236
226 a=3:b=4:c=5:d=6:e=7:f=8:g=9:h=10:i=11:j=12:k=13:l=14
228 PEN 1:LOCATE 6,21:PRINT"loading .."
230 FOR x=3 TO 14:INK x,x:NEXT
232 LOCATE 3,25:PRINT"
;
234 RUN"!"
236 PLOT 96,210
238 READ a:READ b
240 IF a=-111 THEN 274
242 DRAWR a,b,f:f=f+1:IF f=15 THEN f=3
244 GOTO 238
246 DATA -10,10,-10,0,-10,-10,-5,-10,0,-10,10,-10,5,0,5,-10,0,-10,-10,-10,-20,0
248 DATA 22,0,54,66,-25,-66,4,12,-4,4,-4,0,-4,-4,0,-4,4,-4,4,0
250 DATA 36,34,-14,-30,4,-8,20,0,8,8,10,26,-4,6,-17,0,-8,-2,-4,-6,4,-4,24,0,5,5,2,6,-13,-30,6,-8,4,0
252 DATA 24,38,-4,-15,5,-2,5,0,5,2,5,15,-10,-36,4,-2,2,0,20,28,10,8,10,0,8,-4
254 DATA -8,4,-10,0,-10,-8,-6,-20,5,-8,10,-2,5,0,5,2
256 DATA 30,36,-14,-30,4,-8,20,0,8,8,10,26,-4,6,-17,0,-8,-2,-4,-6,4,-4,24,0,15,4
258 DATA 8,6,-14,-38,12,30,6,6,6,2,4,-2,2,-2,0,-2,-1,-4,-10,-30
260 DATA 12,32,6,6,6,2,4,-2,2,-2,0,-2,-1,-4,-8,-22,2,-4,6,-2,2,0,2,2,20,36
262 DATA -14,-38,12,30,6,6,6,2,4,-2,2,-2,0,-2,-1,-4,-10,-30
264 DATA 12,32,6,6,6,2,4,-2,2,-2,0,-2,-1,-4,-8,-22,2,-4,6,-2,2,0,2,2,20,34
266 DATA -14,-30,4,-8,20,0,8,8,10,26,-4,6,-17,0,-8,-2,-4,-6,4,-4,24,0,5,5,2,6,-13,-30,6,-10,4,0
268 DATA 4,2,20,36,-12,-38,11,34,8,4,4,2,4,0,4,-2,2,-4,-10,-30,2,-4,4,-2,2,0,4,2
270 DATA 22,36,-14,-30,4,-8,20,0,8,8,10,26,-4,6,-17,0,-8,-2,-4,-6,4,-4,24,0,5,5,2,6,8,26,-19,-56,4,-8,4,0
272 DATA -111,-111
274 IF dd=0 THEN PLOT 100,210 ELSE RETURN
276 RESTORE:dd=1:GOTO 238
```

Die Checksummen von Teil 1 finden Sie auf Seite 59


```

100 REM *****
**
101 REM *
*
102 REM *      STARCOMMAND TEIL II
*
103 REM *
*
104 REM *****
**
105 REM
106 REM
107 REM
108 REM
109 REM *****
110 REM * Maschinensprachprogramme *
111 REM *****
112 REM
113 his=1000
114 MEMORY 22500
115 DATA 33,0,192,17,0,90,1,0,64,237,176
,201,33,0,90,17,0,192,1,0,64,237,176,201
116 DATA &21,01,01,&cd,&75,&bb,&3e,08,&c
d,&5a,&bb,&cd,&5a,&bb,&c9
117 DATA &3e,01,&cd,&90,&bb,&21,19,10,&c
d,&75,&bb,&3e,200,&cd,&5a,&bb,&3e,201,&c
d,&5a,&bb,&c9
118 DATA &3e,01,&cd,&90,&bb,&21,19,10,&c
d,&75,&bb,&3e,202,&cd,&5a,&bb,&3e,203,&c
d,&5a,&bb,&c9
119 DATA &3e,01,&cd,&90,&bb,&21,1,10,&cd
,&75,&bb,&3e,151,&cd,&5a,&bb,&c9
120 DATA &3e,01,&cd,&90,&bb,&21,1,10,&cd
,&75,&bb,&3e,205,&cd,&5a,&bb,&3e,205,&cd
,&5a,&bb,&c9
121 DATA &3e,02,&cd,&90,&bb,&21,1,10,&cd
,&75,&bb,&3e,208,&cd,&5a,&bb,&3e,209,&cd
,&5a,&bb,&c9
122 DATA &3e,03,&cd,&90,&bb,&21,1,10,&cd
,&75,&bb,&3e,206,&cd,&5a,&bb,&3e,207,&cd
,&5a,&bb,&c9
123 FOR a=39500 TO 39523:READ w:POKE a,w
:NEXT
124 FOR a=22520 TO 22534:READ w:POKE a,w
:NEXT
125 FOR a=22540 TO 22561:READ w:POKE a,w
:NEXT
126 FOR a=22570 TO 22591:READ w:POKE a,w
:NEXT
127 FOR a=22600 TO 22616:READ w:POKE a,w
:NEXT
128 FOR a=22620 TO 22641:READ w:POKE a,w
:NEXT
129 FOR a=22650 TO 22671:READ w:POKE a,w
:NEXT
130 FOR a=22680 TO 22701:READ w:POKE a,w
:NEXT
131 REM
132 REM *****
133 REM *      Titeltitel      *
134 REM *****
135 REM
136 GOSUB 160

```

```

137 CALL 39500:GOTO 141
138 a$=INKEY$:MODE 0:INK 1,0:INK 0,26
139 BORDER 26:PEN 0:PAPER 1:CLS
140 PRINT CHR$(22);CHR$(0):CALL 39512
141 a=3:b=4:c=5:d=6:e=7:f=8:g=9:h=10:i=1
j=12:k=13:l=14
142 WINDOW 2,19,12,21:PAPER 1:PEN 0:PRIN
T CHR$(22);CHR$(1):LOCATE 11,8:PRINT his
;PRINT CHR$(22);CHR$(0)
143 INK 2,26:INK a,3:a=a+1:IF a=15 THEN
a=3
144 INK b,4:b=b+1:IF b=15 THEN b=3
145 INK c,5:c=c+1:IF c=15 THEN c=3
146 INK d,6:d=d+1:IF d=15 THEN d=3
147 INK e,7:e=e+1:IF e=15 THEN e=3
148 INK f,8:f=f+1:IF f=15 THEN f=3
149 INK g,9:g=g+1:IF g=15 THEN g=3
150 INK h,10:h=h+1:IF h=15 THEN h=3
151 INK i,11:i=i+1:IF i=15 THEN i=3
152 INK j,12:j=j+1:IF j=15 THEN j=3
153 INK k,13:k=k+1:IF k=15 THEN k=3
154 INK l,14:l=l+1:IF l=15 THEN l=3
155 a$=INKEY$
156 IF a$="s" OR a$="S" THEN GOTO 172
157 IF a$="l" OR a$="L" THEN GOTO 517
158 IF a$="e" OR a$="E" THEN MODE 2:INK
0,0:INK 1,26:PEN 1:PAPER 0:BORDER 0:CLS:
END
159 GOTO 143
160 WINDOW 2,19,12,21:PAPER 1:PEN 0:CLS
161 PRINT CHR$(22);CHR$(1)
162 FOR x=1 TO 180 STEP 4:y=y+2:n=n+1:PL
OT 30+x,50+y,n:DRAW 610-x,50+y:DRAW 610-
x,230-y:DRAW 30+x,230-y:DRAW 30+x,50+y:I
F n=14 THEN n=2
163 NEXT
164 LOCATE 1,3:PRINT          " S=S
TART E=ENDE                  L=ERKI
AERUNG
HISCORE:"
165 PRINT CHR$(22);CHR$(0)
166 RETURN
167 REM
168 REM *****
169 REM *      Spiel - Start      *
170 REM *****
171 REM
172 PRINT CHR$(23);CHR$(0):i1=20
173 i4=3
174 i1=0.5:i2=1:i3=1
175 SC=0:sco=0
176 PEN #1,1:PAPER #1,0:CLS #1
177 gc=15
178 GOTO 206
179 RESTORE 187
180 GOTO 182
181 PLOT 516,200
182 READ aa:READ bb
183 IF aa=-101 THEN 181
184 IF aa=-111 THEN RETURN
185 DRAW aa/s1,bb/s2,0
186 GOTO 182
187 DATA -10,10,-10,0,-10,-10,-5,-10,0,-

```

```

10,10,-10,5,0,5,-10,0
188 DATA -10,-10,-10,-20,0
189 DATA 22,0,54,66,-25,-66,4,12,-4,4,-4
,0,-4,-4,0,-4,4,-4,4,0
190 DATA 36,34,-14,-30,4,-8,20,0,8,8,10,
26,-4,6,-17
191 DATA 0,-8,-2,-4,-6,4,-4,24,0,5,5,2,6
,-13,-30,6,-8,4,0
192 DATA 24,38,-4,-15,5,-2,5,0,5,2,5,15,
-10,-36,4,-2,2,0,-101,-101
193 DATA -8,4,-10,0,-10,-8,-6,-20,5,-8,1
0,-2,5,0,5,2
194 DATA 30,36,-14,-30,4,-8,20,0,8,8,10,
26,-4,6
195 DATA -17,0,-8,-2,-4,-6,4,-4,24,0,15,
4
196 DATA 8,6,-14,-38,12,30,6,6,6,2,4,-2,
2,-2,0,-2,-1,-4,-10,-30
197 DATA 12,32,6,6,6,2,4,-2,2,-2,0,-2,-1
,-4,-8,-22,2,-4,6,-2,2,0,2,2,20,36
198 DATA -14,-38,12,30,6,6,6,2,4,-2,2,-2
,0,-2,-1,-4,-10,-30
199 DATA 12,32,6,6,6,2,4,-2,2,-2,0,-2,-1
,-4,-8,-22,2,-4,6,-2,2,0,2,2,20,34
200 DATA -14,-28,4,-8,20,0,8,8,10,26,-4,
6,-17,0,-8,-2,-4,-6,4,-4,24
201 DATA 0,5,5,2,6,-13,-28,6,-10,4,0
202 DATA 4,2,20,36,-12,-38,11,34,8,4,4,2
,4,0,4,-2,2,-4
203 DATA -10,-30,2,-4,4,-2,2,0,4,2
204 DATA 22,36,-14,-28,4,-8,20,0,8,8,10,
26,-4,6,-17,0,-8,-2,-4,-6,4,-4,24,0,5,5,
2,6,8,26,-19,-56,4,-8,4,0
205 DATA -111,-111
206 ENT 1,100,10,1
207 ENT 2,115,30,1
208 ENT 3,100,10,1
209 ENV 4,50,15,30
210 MODE 1
211 INK 0,0
212 INK 1,26
213 INK 2,9
214 INK 3,15
215 BORDER 15
216 WINDOW 11,30,4,22
217 WINDOW #2,31,40,1,25:PAPER #2,3:CLS
#2
218 WINDOW #3,1,10,1,25:PAPER #3,3:CLS
#3
219 WINDOW #4,1,40,1,3:PAPER #4,3:CLS#
4
220 WINDOW #1,1,9,5,21:PAPER #1,1:PEN #1
,0:CLS#1
221 WINDOW #5,1,40,23,25:PAPER #5,3:CLS#
5
222 WINDOW #6,6,7,7,15:PAPER #6,2:CLS#6
223 WINDOW #7,32,40,5,21:PAPER #7,1:CLS#
7
224 cv=80:x=15:full=172
225 level=1
226 PAPER #7,0:PEN #7,1:PEN #1,3
227 PAPER 4:PEN 1:INK 2,9
228 GOSUB 333

```

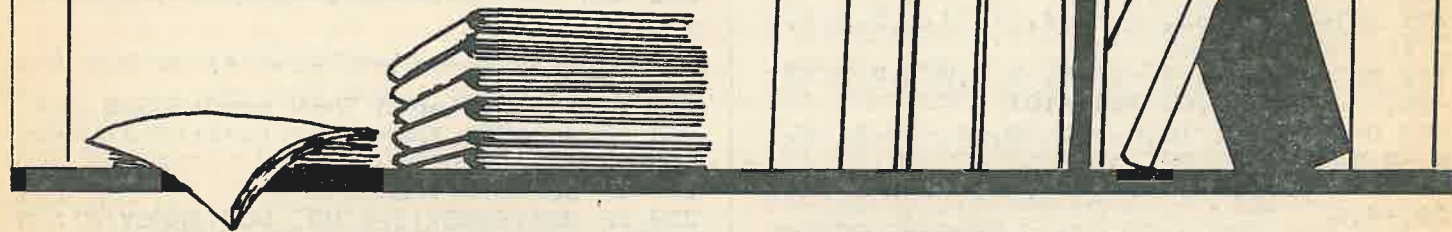
```

229 REM
230 REM *****
**
231 REM *      Hauptteil
*
232 REM *****
**
233 REM
234 REM
235 sp=sp+1:IF sp=5 THEN sp=0:GOSUB 354
236 IF sco=290 THEN i1=1:i2=1:i3=1:L4=0:
CC=1
237 IF sco=310 THEN 392
238 IF NOT(INKEY(1)) OR NOT(INKEY(8)) O
R NOT(INKEY(47)) THEN 283
239 jo=JOY(0):IF jo=4 OR jo=8 OR jo=16 T
HEN 283
240 POKE 22547,x:CALL 22540
241 REM
242 REM *****
**
243 REM *      Hindernisse setzen
*
244 REM *****
**
245 REM
246 IF 14=0 THEN 253
247 i1=i1+RND*2-1:IF i1<10+i4 THEN i1=i1
+1
248 IF i1>30-i4 THEN i1=i1-1
249 PLOT i1*16-i4*16,338,1:DRAW 160,338:
PLOT i1*16+i4*16,338:DRAW 476,338
251 REM *      METORIT
*
253 g=RND:IF g<11 THEN 258
254 dd=INT(RND*19+1):POKE 22607,dd:CALL
22600
256 REM *      Grosser Meteorit
*
258 g1=RND:IF g1<12 THEN 263
259 dd=RND*18+1:POKE 22627,dd:CALL 22620
261 REM *      MINEN ZEICHNEN
*
263 g2=RND:IF g2<13 THEN 268
264 dd=RND*18+1:POKE 22657,dd:CALL 22650
266 REM *      TANK ZEICHNEN
*
268 PEN 3:cc=cc+1:IF cc=50 THEN cc=1:GOT
O 270
269 GOTO 274
270 bb=INT(RND*12+4):POKE 22687,bb:CALL
22680
272 REM *      Bildpunkte pruefen
*
274 t1=TEST ((x+10)*16-9,67):IF t1 =1 OR
t1=2 OR t1=4 THEN 294
275 IF t1=3 THEN 303
276 t1=TEST ((x+10)*16+7,67):IF t1 =1 OR
t1=2 OR t1=4 THEN 294
277 IF t1=3 THEN 303
278 CALL 22520
279 GOTO 235
280 REM *****
281 REM *      Tastaturabfrage
*
282 REM *****
283 jo=JOY(0):IF INKEY(47)=0 OR jo=16 TH
EN 316

```

Fortsetzung Seite 56

Was fehlt auf Ihrem Bücherbord?



Computer – mal anders betrachtet

Ein ganz anderes Computerbuch, als man gewöhnt ist, hat der Weltkreisverlag herausgebracht.

Es stellt in einfachen Beispielen die grundlegende Funktionsweise des Computers dar. Insofern ein ideales Buch für den absoluten Einsteiger. Aber halt! Was ist daran so anders, als in den vielen anderen Einsteigerbüchern? Das Buch, das schließlich auch „DAS ANDERE COMPUTERBUCH“ heißt, führt ausführlich in die Geschichte des Computers ein, und scheut auch nicht die Diskussion solcher Themen wie Datenschutz, Rationalisierung durch Computer oder auch die Bedeutung von Computern in der Kriegstechnik. Auch das „Kriegsspiel“ im Kinderzimmer wird eingehend problematisiert.

Das andere Computer-Buch: für Benutzer, Neugierige u. Skeptiker, Gero von Randow (Hrsg.), Dortmund, 1985, Weltkreisverlag



Neben einem Computer-ABC, das den Leser durch das fachchinesisch führt und eine Reihe von wichtigen Adressen, die man zum Teil in keiner Publikation erfährt (z. B. die der Datenschutzbeauftragten), bietet der Verlag noch einen Diskettenservice mit einfachen Hilfsprogrammen an.

Insgesamt legt das Buch mehr Wert auf den gesellschaftlichen Weltblick der frischgebackenen User als auf fachmännisches Wissen. Expertenbücher gibt es ja nun auch schon reichlich. Allerdings völlig mißglückt muß man den Vergleich der politischen Systeme von Bundesrepublik und DDR betrachten. Denn noch ist nicht zu erkennen, daß die DDR mit den neuen Technologien humaner umgeht, als andere.

Computerspiele und Computergrafik

von Keith S. Reid-Green

Anleitung zur eigenen Software-Entwicklung

Erstes Ziel des Buches ist es, dem Leser beim Schreiben eigener Strategie- und Spielprogramme zu helfen. Es beginnt mit einfachen Grundlagen, die dann im weiteren Verlauf des Buches ausgebaut werden.

Der Inhalt gliedert sich in drei Teile. Teil 1 beschreibt die zum Programmieren notwendigen Grundlagen. Zum Verständnis des Spielbaus und der Spielbeschreibungen werden hier alle Vereinbarungen (Codierung etc.) aufgeführt, die in den Programmbeispielen verwendet werden.

Teil 2 enthält einige einfache Beschreibungen von Spieltypen unter Verwendung von trigonometrischen, geometrischen und physikalischen Prinzipien. Jedes Kapitel beschreibt einen speziellen Spieltyp, bzw. hebt eine bestimmte Idee oder Spieltechnik hervor. Außerdem findet



der Leser hier Lehrreiches zum Einsatz der Kenntnisse aus Mathematik, Physik und Computerwissenschaften.

In Teil 3 wird die Entwicklung von Grafiken und Texten in Spielprogrammen aufgezeigt. Er enthält viele Anregungen für Projekte, die über den Rahmen des Buches hinausgehen.

Das Buch bietet keine Spielesammlung

an. Der Schwerpunkt liegt auf der Erklärung und Beschreibung der Problemstellung (Ausgangspunkt), der Pseudocodierung ist im Anschluß daran erläutert. Alle Programme sind zwar in BASIC geschrieben, obwohl nicht die Eigenschaften eines bestimmten BASIC-Dialekts die Programmierung bestimmt haben. Zum besseren Verständnis wird ein sogenanntes Strukturiertes BASIC eingesetzt, das Pascal-ähnliche Sprachkonzepte anbietet. Die Codierung ist auf Tektronix BASIC bezogen. Durch die weitgehende Unabhängigkeit der Spielprogrammerkklärung von einem BASIC-Dialekt kann das Buch als Lehr- und Grundlagenbuch eingesetzt werden. Braunschweig: Vieweg 1986. VI, 314 S. Kart. DM 56,- ISBN 3 528 04357 1

Karl-Heinz Becker und Michael Dörfler

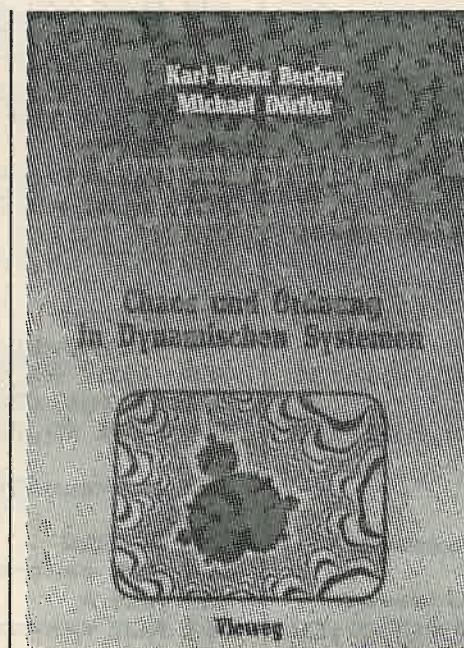
Computergrafische Experimente mit Pascal

Ordnung und Chaos in Dynamischen Systemen

Das „Apfelmännchen“ ist nur eine der vielen Figuren aus der Palette der mit dem Computerprogramm entworfenen Bilder dieses Buches, das in seiner Art der Verknüpfung von Wissenschaftstheorie und Computeralgorithmen wohl einmalig ist, ja eine Pionierleistung darstellt. Dazu tragen einerseits der Bilderreichtum, dann die Pascal-Programme auf Mikrocomputern (Apple II, Apple Macintosh und IBM PC) und die eingängige und verständliche Präsentation der Theorie komplexer Dynamik – Chaos und Ordnung – bei.

Mit ihrer gelungenen Synthese von mathematischer Theorie – die für jeden mit Kenntnissen der Mathematik der Sekundarstufe II nachvollzogen werden kann – und Programmierpraxis in Pascal betreten die Autoren K.-H. Becker und M. Dörfler publizistisches Neuland.

Das Buch wendet sich an alle, die über ein Computersystem verfügen und Spaß am Experimentieren mit Computergrafiken haben. Die verwendeten mathematischen Formeln sind so einfach, daß sie leicht verstanden und auf einfache Weise benutzt werden können. Dabei wird der Leser mit einem Grenzgebiet aktueller, wissenschaftlicher Forschung bekannt gemacht, in dem ohne Computereinsatz und grafische Datenverarbeitung kaum eine Erkenntnisgewinnung möglich wäre.



Es gliedert sich in drei Teile:

Im ersten Teil leitet eine große Zahl von Aufgaben zu eigenem experimentellen Arbeiten und selbständigen Lernen an. Im zweiten Teil werden die wichtigsten Informationen zur Benutzung des UCSD-Pascal vermittelt und im dritten Teil werden die Bausteine angegeben, mit deren Hilfe eigene computergrafische Experimente durchgeführt werden können.

Die Angaben im zweiten und dritten Teil beziehen sich auf die Apple Pascal Version 1.1 des UCSD-Pascal-Systems, das auf dem Apple II und allen Applekompatiblen

Rechnern läuft; Implementierungen auf anderen Rechnern (Macintosh und IBM PC) weichen geringfügig ab und sind ausführlich erläutert. Braunschweig: Vieweg 1986. XII, 319 S. mit 105 Abbildungen. Kart. DM 42,-

Das ist die SUPER-GEWINN-CHANCE für alle C-16/116 Besitzer!

Die Redaktion der Zeitschrift „Aktueller Software-Markt“ verlost in Ihrer nächsten Ausgabe (Heft 5) insgesamt 50 Programmpakete mit je 10 Programmkassetten für den Commodore 16/116. Es handelt sich hierbei um die neuesten Produkte aus dem Hause Mastertronic. Wenn Sie sich diese tolle Gewinn-Chance nicht entgehen lassen wollen, fragen Sie Ihren Zeitschriftenhändler nach „ASM“ und lösen Sie das Preisrätsel.

Nutzen Sie Ihre Chance und gewinnen Sie 10 von insgesamt 500 zu verlosenden Programmkassetten.


```

284 IF x=1 THEN 287
285 jo=JOY(0):IF INKEY(8)=0 OR jo=4 THE
N x=x-1:t6=TEST((x+10)*16-9,51)
286 IF x=19 THEN 289
287 jo=JOY(0):IF INKEY(1)=0 OR jo=8 THE
N x=x+1:t6=TEST((x+10)*16+7,51)
288 IF t6=1 OR t6=2 OR t6=4 THEN 294
289 POKE 22577,x:CALL 22570
290 GOTO 246
291 REM *****
**
292 REM *           Kollision
*
293 REM *****
**
294 IF t1=2 OR t6=2 THEN GOTO 509
295 SOUND 1,40,10,15,0,0,15
296 BORDER 20
297 d=d+1:full=full-2:IF full <33 THEN 5
09
298 SOUND 1,full+30,10,15,1,1
299 PEN 3: LOCATE x,19:PRINT CHR$(200);C
HR$(201);
300 GOSUB 351
301 IF d<20 THEN 297
302 BORDER 15:d=0:GOTO 278
303 SOUND 1,100,50
304 REM *****
*
305 REM *           (Tank) neue Energie
*
306 REM *****
*
307 gi=0
308 gi=gi+1:full=full+2:IF full>172 THEN
full=172:GOTO 278
309 PLOT 80,full+126,2:DRAW 110,full+126
,2
310 PLOT 80,full+128,1:DRAW 110,full+128
,1
311 IF gi<15 THEN 308
312 GOTO 278
313 REM *****
314 REM *           Schuss
*
315 REM *****
316 PEN 1:LOCATE x,19:PRINT CHR$(200);CH
R$(201);ss=51
317 SOUND 3,20,60,7,1,1
318 IF ss>335 THEN 322
319 ss=ss+16:t2=TEST((x+10)*16-9,ss)
320 t3=TEST((x+10)*16+7,ss)
321 IF t2=0 AND t3=0 THEN 318
322 PLOT (x+10)*16+7,60,1:DRAW (x+10)*16
+7,ss:PLOT (x+10)*16-9,60:DRAW (x+10)*16
-9,ss
323 PLOT (x+10)*16+7,60,0:DRAW (x+10)*16
+7,ss:PLOT (x+10)*16-9,60:DRAW (x+10)*16
-9,ss:d=0
324 d=d+1:full=full-2:IF full <33 THEN 5
09
325 GOSUB 351
326 IF d<5 THEN 324 ELSE d=0
327 IF t2=0 AND t3=0 THEN 246

```

```

328 LOCATE x,(22-ss/16):PRINT " "
329 GOTO 246
330 REM *****
331 REM *           Bildschirmaufbau
*
332 REM *****
333 PLOT 158,46,1:DRAW 480,46:DRAW 480,3
52:DRAW 158,352:DRAW 158,46
334 PLOT 148,36,1:DRAW 490,36:DRAW 490,3
62:DRAW 148,362:DRAW 148,36
335 PLOT 158,46,1:DRAW 148,36:PLOT 480,4
6:DRAW 490,36:PLOT 480,352
336 DRAW 490,362:PLOT 158,352:DRAW 148,3
62:PLOT 158,46:DRAW 148,36
337 PLOT 1,1:DRAW 148,36,1:PLOT 640,1:DR
AW 490,36:PLOT 1,400:DRAW 148,362
338 LOCATE #7,2,14:PRINT#7," Score "
339 LOCATE #7,2,15:PRINT#7," 0 "
340 LOCATE #1,3,3:PRINT#1,"F."
341 LOCATE #1,3,11:PRINT#1,"E."
342 PEN #1,1:PAPER #1,0:CLS
343 LOCATE #1,2,14:PRINT#1," Level "
344 LOCATE #1,2,15:PRINT#1," "
345 ATE #1,3,15:PRINT#1,level
346 PLOT 640,400:DRAW 490,362
347 PLOT 580,266,0:DRAW 30,0:PLOT 580,2
54:DRAW 30,0
348 GOSUB 179
349 RETURN
350 full=full-2
351 PLOT 80,full+130,2:DRAW 110,full+13
0,2:IF full<1 THEN END
352 PLOT 80,full+128,1:DRAW 110,full+128
,1
353 RETURN
354 full=full-2
355 PLOT 80,full+130,2:DRAW 110,full+13
0,2:IF full<33 THEN 509
356 PLOT 80,full+128,1:DRAW 110,full+128
,1
357 sco=sco+5:LOCATE #7,2,15:PRINT #7,sc
o+5C
358 RETURN
359 REM *****
360 REM *           Game Over
*
361 REM *****
362 CLS
363 11=1:12=1:13=1:14=0
364 INK 2,0:GOSUB 374
365 PRINT CHR$(23);CHR$(1)
366 IF sco>sc>his THEN his=sco+sc:sco=0
367 FOR d=198 TO 48 STEP -2:SOUND 1,d+30
0,1,13,0,0:PLOT 160,d,0:DRAW 478,d,1:PLD
T 160,398-d,0:DRAW 478,398-d,1:NEXT
368 FOR x=1 TO 400:IF INKEY(47)=0 THEN
138
369 NEXT
370 FOR d=48 TO 198 STEP 2:SOUND 1,d+300
,1,13,0,0:PLOT 160,d,0:DRAW 478,d,1:PLOT
160,398-d,0:DRAW 478,398-d,1:NEXT
371 FOR x=1 TO 200:IF INKEY(47)=0 THEN 1
38
372 NEXT

```

```

373 GOTO 367
374 f=0
375 PLOT 220+f,220+f,2
376 DRAW 0,50:DRAW 25,0:DRAW -25,0:DRA
WR 0,-50:DRAW 25,0:DRAW 0,25:DRAW -10
,0
377 PLOT 25,-25
378 DRAW 25,50:DRAW 25,-50:DRAW -12,2
5:DRAW -25,0
379 PLOT 52,-25
380 DRAW 0,50:DRAW 25,-50:DRAW 25,50:
DRAW 0,-50
381 PLOT 20,0:DRAW 0,50:DRAW 25,0:PLD
TR -25,-25:DRAW 25,0:PLOT -25,-25:DRAW
R 25,0
382 PLOT 230+f,120+f
383 DRAW 0,50:DRAW 25,0:DRAW 0,-50:DR
AW -25,0
384 PLOT 60,0:DRAW -25,50:PLOT 50,0:D
RAW -25,-50
385 PLOT 36,0:DRAW 0,50:DRAW 25,0:PLD
TR -25,-25:DRAW 25,0:PLOT -25,-25:DRAW
R 25,0
386 PLOT 20,0:DRAW 0,50:DRAW 25,0:DRA
WR 0,-25:DRAW -25,0:DRAW 10,0:DRAW 15
,-25
387 f=f+1:IF f<10 THEN 375
388 LOCATE 2,19:PEN 2:PRINT" Space drue
cken ":RETURN
389 REM *****
**
390 REM *           Basisschiff erreicht
*
391 REM *****
**
392 d=294:PRINT CHR$(23);CHR$(0)
393 RESTORE 401
394 LOCATE 1,1:PRINT CHR$(8)
395 GOSUB 445
396 PLOT 280,d,3:d=d-16:IF d<80 THEN 448
397 READ a:READ b:IF a=-1 AND b=-1 THEN
409
398 a=a*2:b=b*2
399 DRAW a,b
400 GOTO 397
401 DATA 36,0,0,20,-8,8,-20,0,-8,-8,0,-2
0,8,8,0,20,0,-20,20,0,8,-8,-8,8,0,20
402 DATA -1,-1
403 DATA -2,0,0,10,-4,4,-8,0
404 DATA -4,-4,0,-10,4,0,0,14,0,-14,8,0,
0,14
405 DATA -1,-1
406 DATA 6,0,6,6,-6,6,-28,0
407 DATA -6,-6,6,-6,6,0,-6,0,-6,6
408 DATA -1,-1
409 LOCATE 1,1:PRINT CHR$(8):GOSUB 445
410 LOCATE 1,1:PRINT CHR$(8)
411 GOSUB 445
412 PLOT 336,318,3
413 READ a:READ b:IF a=-1 AND b=-1 THEN
417
414 a=a*2:b=b*2
415 DRAW a,b

```

```

416 GOTO 413
417 LOCATE 1,1:PRINT CHR$(8)
418 PLOT 332,314
419 GOSUB 445
420 READ a:READ b:IF a=-1 AND b=-1 THEN
424
421 a=a*2:b=b*2
422 DRAW a,b
423 GOTO 420
424 FOR hj=1 TO 5:PLOT 276,328:DRAW 276,
350
425 PLOT 356,328:DRAW 356,350
426 PLOT 286,334:DRAW 286,350
427 PLOT 346,334:DRAW 346,350
428 GOSUB 445
429 FOR gg=1 TO hj*20:NEXT
430 LOCATE 1,1:PRINT CHR$(8)
431 NEXT
432 y=-2:FOR hj=0 TO 20 STEP 5:y=y+3
433 PLOT 356-hj,328:DRAW 351-hj,350
434 PLOT 276+hj,328:DRAW 281+hj,350
435 PLOT 286+y,334:DRAW 289+y,350
436 PLOT 346-y,334:DRAW 343-y,350
437 FOR gg=1 TO 100+hj*20:NEXT
438 GOSUB 445
439 LOCATE 1,1:PRINT CHR$(8)
440 GOSUB 445
441 NEXT
442 PLOT 302,334:DRAW 331,334
443 GOSUB 448
444 GOTO 457
445 IF x<10 THEN x=x+1
446 IF x>10 THEN x=x-1
447 PEN 1:LOCATE x,19:PRINT CHR$(200);CH
R$(201):RETURN
448 REM
449 hj=0:hk=1
450 LOCATE 1,4
451 PRINT CHR$(23);CHR$(1)
452 hj=hj+1:hk=hk-1
453 IF hj>20 THEN RETURN
454 SOUND 1,hk+45,5,12,0,0,15:PLOT hj+31
6,200,1:DRAW hj+316,230:DRAW hk+316,230:
DRAW hk+316,200:DRAW hj+316,200
455 gh=gh+1:IF gh=4 THEN gh=1:ff=ff-1
456 GOTO 452
457 LOCATE x,19:PRINT " ";
458 TAG:FOR d=60 TO 224 STEP 2:MOVE 300,
d:PEN 1:PRINT CHR$(200);CHR$(201);
459 MOVE 300,d:PRINT CHR$(200);CHR$(201)
;:NEXT
460 MOVE 300,d:PRINT CHR$(200);CHR$(201)
;
461 TAGOFF
462 LOCATE 1,5:PRINT CHR$(23);CHR$(0)
463 FOR hj=0 TO 20 STEP 2:SOUND 1,hj+45,
5,12,0,0,15:PLOT 336-hj,200,1:DRAW 336-h
j,230,1:PLOT 338-hj,200,0:DRAW 338-hj,23
0,0:PLOT 294+hj,200,0:DRAW 294+hj,230,0:
PLOT 296+hj,200,1:DRAW 296+hj,230,1:NEXT
464 PLOT 316,200,0:DRAW 316,230,0
465 REM *****

```



```

*
466 REM *          NEUE RUNDE
*
467 REM *****
**
468 SOUND 1,40+full,4,15,1,1
469 IF full>36 THEN GOSUB 354 ELSE GOTO
472
470 full=full+1
471 GOTO 468
472 level =level +1
473 LOCATE #1,3,15:PRINT#1,level
474 GOSUB 448
475 d=224
476 TAG:FOR f=0 TO 15:d=d+f:MOVE 300,d:P
EN 1:PRINT CHR$(200);CHR$(201);
477 MOVE 300,d:PRINT CHR$(200);CHR$(201)
;:NEXT
478 LOCATE 5,5
479 TAGOFF
480 PRINT CHR$(23);CHR$(0);
481 full=172;x=15;cv=80
482 IF LEVEL=2 THEN L1=0.7:L2=0.8
483 IF LEVEL=3 THEN L1=0.7:L2=0.7
484 IF LEVEL=4 THEN L1=0.7:L2=0.7:L3=0.9
485 IF LEVEL=5 THEN L1=0.7:L2=0.7:L3=0.8
:14=1:i4=8
486 IF LEVEL=6 THEN L1=0.7:L2=0.8:L3=0.7
487 IF level=7 THEN L1=0.7:L2=0.7:L3=0.6
488 IF level=8 THEN L1=0.7:L2=0.7:L3=0.7
489 IF level=9 THEN L1=0.6:L2=0.6:L3=0.7
:14=1:i4=7
490 IF level=10 THEN L1=0.6:L2=0.7
491 IF level=11 THEN L1=0.6:L2=1:L3=0.6
492 IF level=12 THEN L1=0.7:L2=0.7:L3=0.
5
493 IF level=13 THEN L1=0.6:L2=0.7:L3=0.
5:14=1:i4=7
494 IF level=14 THEN L1=0.6:L2=0.7:L3=0.
5
495 IF level=15 THEN L1=0.5:L2=0.9:L3=0.
5:14=0
496 IF level=16 THEN L1=0.4:L2=0.4:L3=0.
7:14=0
497 IF level=17 THEN L1=0.4:L2=0.4:L3=0.
7:14=1:i4=8
498 IF level=18 THEN L1=0.9:L2=0.4:L3=0.
5
499 IF level=19 THEN L1=0.8:L2=0.6:L3=0.
4:14=1:i4=7
500 IF level=20 THEN L1=0.3:L2=0.9:L3=0.
5
501 IF level>20 THEN L1=1/(level/11):L2=
1/(level/12):L3=1/(level/11)
502 IF level=21 OR level=26 OR level=lev
el=30 OR level= 32 OR level= 39 THEN 14=
1:i4=8
503 IF level=25 OR level=28 OR level =35
OR level=36 OR level=38 THEN 14=1:i4=6
504 IF level>40 THEN 14=1:i4=7
505 SC=SCO+SC:SCO=0
506 SPEED INK 1,1:INK 0,0,26:FOR d=1 TO
1500:NEXT:INK 0,0

```

```

507 CLS:CLS #6:GOTO 235
508 REM *****
*
509 REM *          TOT
*
510 REM *****
*
511 IF x=19 THEN x=18
512 IF x=1 THEN x=2
513 SOUND 1,284,500,1,4,0,15:FOR h1=1 TO
500:NEXT:LOCATE x-1,19:PRINT" ";:FOR
h1=1 TO 150:PLOT 160+x*16,60:DRAW 160+x
*16+RND*64-32,48+RND*80,RND*4:NEXT
514 FOR h1=1 TO 1600:NEXT
515 GOTO 359
516 PRINT CHR$(23);CHR$(0)
517 REM
518 REM *****
**
519 REM *          Erklaerung
*
520 REM *****
**
521 REM
522 REM
523 REM
524 INK 0,0:INK 1,26
525 MODE 2:BORDER 0
526 WINDOW 2,79,2,24
527 PEN 0:PAPER 1:PRINT STRING$(78," ");
528 PRINT" ";:PRINT CHR$(200);CHR$(2
01);:PRINT" S T A R C O M M A N D
geschrieben von Winklbaue Markus ";:P
RINT CHR$(200);CHR$(201);:PRINT" ";
529 PRINT STRING$(78," ");
530 LOCATE 1,6:PRINT"Auf den Raumstation
en , Milliarden von Kilometern von der E
rde entfernt , ist ";
531 LOCATE 1,7:PRINT"eine Seuche ausgebr
ochen . Auf der Erde wurde ein Gegenmitt
el entwickelt , ";
532 LOCATE 1,8:PRINT"welches sie so schn
ell wie moeglich zu den Raumstationen br
ingen müssen , um ";
533 LOCATE 1,9:PRINT"das Leben von viele
n Menschen zu retten . ";
534 LOCATE 1,11:PRINT "Um noch rechtze
itig die Raumstationen zu erreichen müe
ssen sie durch Gebiete ";
535 LOCATE 1,12:PRINT "fliegen in dene
n Gefahren auf sie lauern . Diese Gefahr
en sind Meteoriten = felder , Minen un
d gefaehrliche Strahlungen . Sie haben e
inen Schutz = ";
536 LOCATE 1,14:PRINT "schild der sie
vor den Meteoriten schuetzt . Bei jeder K
ollision mit einem Meteoriten wird ih
r Schutzschild automatisch eingeschaltet
. Der Schutzschild ";
537 LOCATE 1,16:PRINT "benoetigt eine
gewisse Energie . Die Energie wird links
durch einen Balken angezeigt . Sie b

```

```

esitzen auch eine Laserkanone . Jeder Sc
huss braucht widerum ";
538 LOCATE 1,18:PRINT "Energie . D
iese Energie koennen sie wider auffrisch
en in dem sie -Tanks- aufsammeln .
Bei einer Kollision mit einer Miene kann
sie auch der Schutz = ";
539 LOCATE 1,20:PRINT "schild nich
t retten .

```

```

";
540 PEN 1:PAPER 0:LOCATE 1,22:PRINT "
< < < Bitte S P A C E Tas
te druecken > > > ";
541 LOCATE 1,1
542 PRINT CHR$(23);CHR$(1)
543 FOR x=32 TO 48 STEP 2
544 IF INKEY(47)=0 THEN 548
545 PLOT 7,x,1:DRAW 631,x
546 NEXT
547 GOTO 543
548 PRINT CHR$(23);CHR$(0)
549 PAPER 1:PEN 0
550 LOCATE 1,6:PRINT" Die Energie die s
ie beim Fliegen brauchen wird ihnen auch
abgezogen . ";
551 LOCATE 1,7:PRINT" I c h w u e n
s c h e i h n e n v i e l G
l u e c k ! ! ";
552 PEN 1:PAPER 0:LOCATE 1,8:PRINT STRIN
G$(156," ");
553 LOCATE 1,10:PRINT STRING$(156," ")
554 PAPER 1:PEN 0:LOCATE 25,10:PRINT " S
T E U E R U N G ";
555 PEN 0:PAPER 1:LOCATE 1,12:PRINT STRI
NG$(234," ");:LOCATE 3,13:PRINT "Mit Joy
stick oder mit:";
556 PRINT STRING$(234," ");

```

```

557 PRINT STRING$(156," ");
558 PEN 0:PAPER 1:LOCATE 55,12:PRINT CHR
$(150);CHR$(154);CHR$(156)
559 LOCATE 55,13:PRINT CHR$(149);CHR$(24
2);CHR$(149)
560 LOCATE 55,14:PRINT CHR$(147);CHR$(15
4);CHR$(153);
561 LOCATE 65,12:PRINT CHR$(150);CHR$(15
4);CHR$(156)
562 LOCATE 65,13:PRINT CHR$(149);CHR$(24
3);CHR$(149)
563 LOCATE 48,13:PRINT"Links":LOCATE 70,
13:PRINT"Rechts"
564 LOCATE 65,14:PRINT CHR$(147);CHR$(15
4);CHR$(153);
565 LOCATE 10,16:PRINT CHR$(150);STRING$
(40,154);CHR$(156)
566 LOCATE 10,17:PRINT CHR$(149);"
S P A C E ";CHR$
(149)
567 LOCATE 57,17:PRINT"Feuer";
568 LOCATE 10,18:PRINT CHR$(147);STRING$
(40,154);CHR$(153)
569 PEN 1:PAPER 0:LOCATE 1,19:PRINT STRI
NG$(234," ");
570 LOCATE 1,22:PRINT " < < <
Bitte S P A C E Taste druecken
> > > ";
571 LOCATE 1,1
572 PRINT CHR$(23);CHR$(1)
573 FOR x=32 TO 48 STEP 2
574 IF INKEY(47)=0 THEN 578
575 PLOT 7,x,1:DRAW 631,x
576 NEXT
577 GOTO 573
578 PRINT CHR$(22);CHR$(0)
579 GOTO 138

```

Checksummen Teil 1:

CHECK V1:PRUEFSUMMEN:

ZEILENNR.: SUMMEN

100-	102:	23914
104-	106:	48545
108-	110:	21715
112-	114:	8634
116-	118:	22444
120-	122:	394
124-	126:	23914
128-	130:	18182
132-	134:	22444
136-	138:	412
140-	142:	31934
144-	146:	41234
148-	150:	38810
152-	154:	50118
156-	158:	63136
160-	162:	49376
164-	166:	22700
168-	170:	69341
172-	174:	5695
176-	178:	47201
180-	182:	38759
184-	186:	85547
188-	190:	352
192-	194:	103406
196-	198:	2624
200-	202:	52892
204-	206:	38759
208-	210:	79856
212-	214:	352
216-	218:	47002

220-	222:	38759
224-	226:	11118
228-	230:	46257
232-	234:	5587
236-	238:	8706
240-	242:	32460
244-	246:	58350
248-	250:	19572
252-	254:	102201
256-	258:	73235
260-	262:	107142
264-	266:	97504
268-	270:	98911
272-	274:	27941
276-	:	9975

GESAMTCHECKSUMME: 17963

Checksummen
Teil 2:

CHECK V1:PRUEFSUMMEN:

ZEILENNR.: SUMMEN

100-	101:	29878
102-	103:	27048
104-	105:	28240
106-	107:	394
108-	109:	22444
110-	111:	68577
112-	113:	6608

114-	115:	2604
116-	117:	72633
118-	119:	93873
120-	121:	51325
122-	123:	88411
124-	125:	97371
126-	127:	97747
128-	129:	100222
130-	131:	51388
132-	133:	37376
134-	135:	21100
136-	137:	6705
138-	139:	20200
140-	141:	24476
142-	143:	25069
144-	145:	123197
146-	147:	123633
148-	149:	124069
150-	151:	129528
152-	153:	129984
154-	155:	69260
156-	157:	63347
158-	159:	22248
160-	161:	18002
162-	163:	60894
164-	165:	44079
166-	167:	398
168-	169:	34993
170-	171:	17320
172-	173:	20830
174-	175:	41926
176-	177:	9620
178-	179:	2111
180-	181:	3309
182-	183:	23295

184-	185:	42948
186-	187:	4945
188-	189:	16353
190-	191:	108604
192-	193:	67412
194-	195:	67974
196-	197:	51956
198-	199:	107008
200-	201:	42794
202-	203:	71609
204-	205:	40024
206-	207:	4877
208-	209:	4835
210-	211:	668
212-	213:	1165
214-	215:	869
216-	217:	16456
218-	219:	26394
220-	221:	34136
222-	223:	26033
224-	225:	42150
226-	227:	12099
228-	229:	759
230-	231:	47954
232-	233:	28240
234-	235:	61691
236-	237:	54248
238-	239:	81193
240-	241:	7351
242-	243:	61787
244-	245:	28240
246-	247:	61401
248-	249:	67711
251-	253:	35538
254-	256:	73037

258-	259:	65312
261-	263:	47373
264-	266:	56374
268-	269:	5967
270-	272:	83574
274-	275:	34992
276-	277:	34921
278-	279:	2826
280-	281:	51916
282-	283:	82248
284-	285:	43303
286-	287:	43627
288-	289:	54775
290-	291:	29277
292-	293:	46135
294-	295:	39573
296-	297:	42051
298-	299:	53261
300-	301:	7833
302-	303:	9963
304-	305:	55116
306-	307:	29653
308-	309:	95657
310-	311:	60128
312-	313:	21246
314-	315:	32450
316-	317:	65128
318-	319:	34955
320-	321:	65119
322-	323:	38317
324-	325:	42400
326-	327:	48789
328-	329:	24197
330-	331:	46795
332-	333:	78099

334- 335: 116492	384- 385: 54534	434- 435: 56382	484- 485: 48438	534- 535: 76740
336- 337: 118101	386- 387: 33733	436- 437: 57117	486- 487: 110751	536- 537: 46053
338- 339: 22568	388- 389: 71317	438- 439: 6380	488- 489: 105139	538- 539: 65220
340- 341: 14609	390- 391: 64731	440- 441: 1186	490- 491: 53606	540- 541: 34742
342- 343: 21024	392- 393: 17701	442- 443: 8166	492- 493: 81444	542- 543: 18090
344- 345: 66906	394- 395: 6380	444- 445: 26664	494- 495: 45955	544- 545: 21256
346- 347: 60631	396- 397: 68096	446- 447: 66264	496- 497: 59156	546- 547: 560
348- 349: 1166	398- 399: 29903	448- 449: 11327	498- 499: 90767	548- 549: 7425
350- 351: 69637	400- 401: 50490	450- 451: 6650	500- 501: 74177	550- 551: 21880
352- 353: 51541	402- 403: 12608	452- 453: 47031	502- 503: 11908	552- 553: 34592
354- 355: 79372	404- 405: 34998	454- 455: 82837	504- 505: 71559	554- 555: 52313
356- 357: 53171	406- 407: 26954	456- 457: 7387	506- 507: 52530	556- 557: 10019
358- 359: 17324	408- 409: 13004	458- 459: 71041	508- 509: 33264	558- 559: 92479
360- 361: 29862	410- 411: 6380	460- 461: 30444	510- 511: 41840	560- 561: 73693
362- 363: 38269	412- 413: 5373	462- 463: 44410	512- 513: 35516	562- 563: 80890
364- 365: 9560	414- 415: 29903	464- 465: 39426	514- 515: 11702	564- 565: 80635
366- 367: 67863	416- 417: 6253	466- 467: 42475	516- 517: 5555	566- 567: 73426
368- 369: 37577	418- 419: 2504	468- 469: 45119	518- 519: 48253	568- 569: 68805
370- 371: 98953	420- 421: 28937	470- 471: 18394	520- 521: 28240	570- 571: 7950
372- 373: 875	422- 423: 5347	472- 473: 41233	522- 523: 394	572- 573: 18090
374- 375: 18438	424- 425: 32169	474- 475: 4860	524- 525: 3725	574- 575: 21706
376- 377: 67415	426- 427: 14556	476- 477: 47798	526- 527: 15688	576- 577: 680
378- 379: 35856	428- 429: 23092	478- 479: 706	528- 529: 51389	578- 579: 6154
380- 381: 47726	430- 431: 5546	480- 481: 39856	530- 531: 105348	
382- 383: 39217	432- 433: 92905	482- 483: 73746	532- 533: 109678	

GESAMTCHECKSUMME: 55955

Werkstatt:

Diesmal finden Sie unter der Rubrik „Werkstatt“ ein Programm, mit dem alle Diskettenbesitzer angesprochen werden. Gehören Sie zu diesem Kreis, dann sollten Sie genau aufpassen, denn hier wird etwas veröffentlicht, auf das Sie sicher schon lange warten und was eigentlich jeder User für den Umgang mit dem Schneider benötigt.

Disksort für den CPC 464

Mit diesem Programm kann man endlich die eigenen Disketten ordnen und bis zu 651 Directory-Einträge speichern und ausdrucken.

Gehen Sie dabei so vor:

Starten Sie den Disk-Sort und wählen Sie Menue-Punkt 1. Sie werden nun nach der Disketten-Identifikation gefragt. Sinnvoll ist es, jeder Diskette eine Buchstaben-Nummern-Kombination zu geben. Lesen Sie darauf nach Anweisung die Diskette ein und drücken Sie eine Taste. Das Directory wird angezeigt. Nun entsteht eine kurze Wartezeit, in der es aus dem Speicher in das Feld b\$ eingelesen wird. Dann

erscheint wieder das Menue. Laden Sie nun so viele Directories wie sie möchten. Sie können dann durch Menue-Punkt 3 alphabetisch sortiert werden. Leider kann es hier passieren, daß sich der CPC zu einer Garbage Collection (sprich: Müllabfuhr) verabschiedet. Das kann unter Umständen eine Zeitlang dauern und ist leider nicht zu umgehen.

Mit Menue-Punkt 4 kommen Sie in den Editier-Modus. Es werden jeweils 5 Ele-

mente angezeigt, das aktuelle in selber Schrift. Mit den <Cursor-Up> und <Cursor-Down> Tasten kann gescrollt werden. Mit <COPY> kann das aktuelle Element in den Speicher gelesen werden. Mit kann das aktuelle Element gelöscht werden. Mit <CLR> oder <CTRL> P kann das Element im Speicher an aktueller Stelle eingefügt werden. Und mit der Taste ~ kann der gesamte Speicher von Hand eingegeben werden. Ferner stehen

folgende Tastenfunktionen zur Verfügung: Bei <CTRL> <Cursor up> wird an den Anfang, mit <CTRL> <Cursor down> an das Ende gesprungen. Drücken Sie <CTRL> <Cursor right>, werden Sie nach einem Namen und Typ gefragt. Das Programm springt dann zu dieser Stelle.

Mit dem EDIT-Modus können Sie also zusammengehörige Teile entgegen der alphabetischen Sortierung, zusammenstellen. Den EDIT-Modus können Sie mit <ENTER> verlassen.

Bei den Punkten 6 und 7 werden Sie jeweils nach dem Dateinamen gefragt. Als

Typ wird immer ein .SRT angehängt. Mit Punkt 2 kann alles gelöscht werden. Mit Punkt 5 wird alles in 2 Spalten ausgedruckt. Sie können auch eine Überschrift eingeben, die zentriert ausgegeben wird. Mit Punkt 0 beenden Sie den Disk-Sort.

Variablenliste:	
a\$ (0-65)	- Dateinamen für ein Directory
l (0-65)	- Dateilängen für ein Directory
b\$ (0-650)	- alle eingelesenen Dateinamen
k (0-650)	- alle eingelesenen Dateilängen
d\$ (0-650)	- Diskettennamen für jede Datei
s (0-99)	- Hilfsfeld für Quicksort
r	- Anzahl der Dateinamen
a\$,a,e\$	- Tastatur-Abfrage
e\$	- Name der eingelegten Diskette/Dateiname zum Saven + Laden
k,l,i	- Schleifen
n,p,l,r,i,j,d,s,\$	- Hilfsvariablen für Quicksort
sp\$,sd\$,sk	- Dateiname, Länge + Diskette im Speicher
y	- y-Koordinate im Editiermodus
ei\$,e2\$	- Hilfsvariablen
ss,s,\$\$	- Hilfsvariablen zur Menuearstellung

Programmteilbeschreibung

0 - 340	: Initialisierung	5000 - 5520	: Editiermodus
350 - 610	: Menue	6000 - 6170	: Ausgabe auf Drucker
1000 - 1060	: Ende des Diskortes	7000 - 8080	: Abspeichern + Laden
2000 - 2180	: Das Directory wird geladen und in a\$ und b\$ abgelegt	9000 - 9100	: Editiermodus (Fortsetzung)
3000 - 3030	: Löschen des Buffers, indem r=0 gesetzt wird		
4000 - 4230	: Alphabetischer Quicksort		

In Zeile 2030 können Steuerzeichen vor dem Ausdruck an den Drucker geschickt werden.

```

100 * %Z
XXXXXXXXXXXXXXXXXXXXXXXXXXXX
110 * %
120 * % DISK-SORT (C) 1985 %
130 * % by %
140 * % JeBaSoft Utilities %
150 * %
160 * % von Jens Barth am 18.8.1985 %
170 * %
180 * % V 1.3 %
190 * %
200 * XXXXXXXXXXXXXXXXXXXXXXXX
210 :
220 :
230 SYMBOL 240,31,31,30,30,30,31,30,28
240 SYMBOL 241,248,248,120,120,120,248,1
20,56
250 SYMBOL 242,30,31,16,16,16,31,0,0
260 SYMBOL 243,120,248,8,8,8,248,0,0
270 CALL &BC02:MODE 1
280 PRINT "Disk-Sort V1.3":PRINT CHR$(164)
)" JeBaSoft Utilities "85"
290 ON BREAK GOSUB 350
300 KEY 140,CHR$(13)+"goto 350"+CHR$(13)
310 OPENOUT "dum":MEMORY HIMEM-1:CLOSEOUT
320 DEFINT 1,k
330 DIM a$(65),l(65),b$(650),k(650),d$(6
50),s(99)
340 r=0
350 CALL &BC02:MODE 1:PEN 1:PRINT"

```

```

I S K - S O R T (V1.3)
360 PEN 1:PRINT:PRINT" Menue :
":PRINT"
370 PEN 2
380 RESTORE 390:FOR gg=1 TO 8:READ g:REA
D g$:LOCATE 7,g:PRINT g$:NEXT gg
390 DATA 7,"Directory laden"
400 DATA 9,"Alles loeschen"
410 DATA 11,"Alphabetisch sortieren"
420 DATA 13,"Editieren"
430 DATA 16,"Liste ausdrucken"
440 DATA 18,"Daten abspeichern"
450 DATA 20,"Daten laden"
460 DATA 23,"Ende des Disk-Sort"
470 PEN 3
480 RESTORE 390:FOR gg=1 TO 7:READ g:REA
D g$:LOCATE 3,g:PRINT gg:NEXT gg
490 READ g:LOCATE 3,g:PRINT 0
500 PEN 2
560 PRINT CHR$(7)
570 a$=""
580 WHILE a$<"0" OR a$>"7":a$=INKEY$:WEN
D
590 a=VAL(a$)
600 ON a+1 GOSUB 1000,2000,3000,4000,500
0,6000,7000,8000
610 GOTO 350
1000 ' Ende des Disk-Sort
1010 MODE 2:PEN 1
1020 PRINT"

```

Mit

Compute mit

6/86

Compute mit