

Home Computer

5 1. Jahrgang

Mai '83 5,50 DM 45 öS 6,00 sfr

In diesem Heft u.a.:

VC-20

Editor
Assembler
HC-Invaders
Car-Crash

Sharp MZ-80K

Eidechse

VC-64

Geisterjäger
Mondlandung
Wurm
Sprite-Generator

ZX-81

Ritter
Missile ZX-Command

Apple II

Disk-Schutz 2

CBM

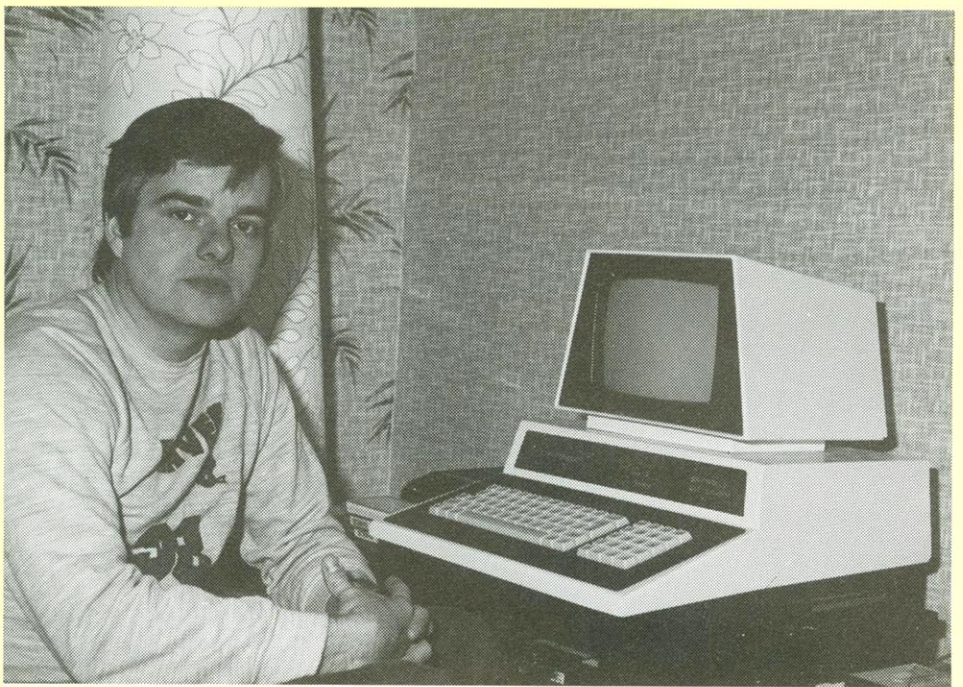
Mastermind (Computerhirn)
Mastermind II

Texas-Special

mit vielen Programmen für den TI-99/4A

Jeden Monat mehr als
15 neue Programme
+ Tips + News
aus aller Welt

Große Werbeaktion!
250 Computer
zu gewinnen!



Liebe Leserin, lieber Leser,

wenn Sie diese Zeilen lesen, ist die Hannover-Messe 83 schon beendet.

Jetzt im Moment, in dem ich diese Zeilen schreibe, steht sie wenige Stunden bevor. 8 Tage lang werden wir uns bemühen, das Interessanteste für Sie, die Leser von Homecomputer, herauszupicken, werden wir versuchen, detaillierte Informationen über alles Neue und Berichtenswertes zusammenzufassen.

Im Juni-Heft werden wir dann darüber berichten. Darum wird die Neuheitenrubrik etwas vergrößert werden. Allerdings würde ein allumfassender Bericht den Rahmen von Homecomputer bei weitem sprengen und weniger veröffentlichte Software bedeuten.

Dies möchten wir nicht. Auch in Heft 6/83 sollen, wie bei uns üblich, mehr als 15 Programme enthalten sein.

Aus diesem Grunde bringen wir ein „Sonderheft Hannover-Messe“ heraus, damit Sie sich – auch falls Sie nicht die Gelegenheit hatten länger oder überhaupt diese wichtigste aller Messen zu besuchen – über alles ausführlich informieren können.

Microcomputer, Drucker, Datenspeichergeräte fürs Heim; darüber werden wir in Wort und Bild berichten und hoffen, Ihnen damit eine interessante Ergänzung zum gewohnten Homecomputer mit dem Programmservice zu bieten.

Herzlichst
Ihr Ralph Roeske

Herausgeber, Chefredakteur

Homecomputer erscheint monatlich im:
Roeske Verlag, Eschwege

Herausgeber:
Ralph Roeske

Redaktion:
Ralph Roeske (R),
Chefredakteur
(verantwortlich)
Cornelius Geppert (Ge)
Bärbel Voigt

Herstellung:
Roeske Verlag, Eschwege

Satz und Reproduktion:
Imtal Composing GmbH
3436 Hessisch Lichtenau

Druck:
Vogt GmbH
Druckerei und Verlag
3436 Hessisch Lichtenau

Vertrieb:
Inland (Groß-, Einzel- und
Bahnhofsbuchhandel), sowie
Österreich und Schweiz:
Verlagsunion
Friedrich-Bergius-Str. 7
6200 Wiesbaden
Tel: 06121-2660

Anzeigenleitung:
Monika Roeske

Erscheinungsweise:
Erstverkaufstag von Home-
computer ist jeweils der letzte
Montag vor dem Erschei-
nungsmonat

Urheberrecht:
Alle in Homecomputer veröf-
fentlichen Beiträge sind urhe-
berrechtlich geschützt. Alle
Rechte, auch Übersetzungen,
vorbehalten. Reproduktionen
jeder Art (Fotokopie, Micro-
film, Erfassung in Datenver-
arbeitungsanlagen, usw.) be-
dürfen der schriftlichen Ge-
nehmigung des Verlages.
Alle veröffentlichte Software
wurde von Mitarbeitern des
Verlages oder von freien Mit-
arbeitern erstellt. Aus Ihrer
Veröffentlichung kann nicht
geschlossen werden, daß die
beschriebenen Lösungen
oder Bezeichnungen frei von
Schutzrechten sind.

Anschrift:
Homecomputer
Gartenstraße 47
3440 Eschwege
Tel: 05651-5993

Bezugspreise:
Einzelheft: 5,50 DM
Abonnement: Inland 55,- DM
pro Jahr (12 Ausgaben)
Ausland: Europa 80,- DM
USA 110,- DM

Anzeigenpreise:
Es gilt die Anzeigenpreisliste
Nr. 1 vom 1. Januar 1983
1/1 Seite S/W 3000,- DM
Farbzuschlag je Farbe
(Europaskala) 500,- DM

Nachdruck gestattet:
Auszüge von Homecomputer
(weniger als 10% des Redak-
tionsanteils) dürfen ohne Ge-
nehmigung in Schülerzeit-
schriften, Computer-Club
Magazinen und ähnlichen
nichtgewerblichen Publika-
tionen abgedruckt werden.

Bedingungen:
Es darf nur Originalmaterial
kopiert werden (also keine
Kopie einer Kopie). Jeder
Nachdruck muß an seinem
Beginn folgende Copyright-
Notiz in Original-Größe bein-
halten:
Copyright 1983
Homecomputer,
Gartenstraße 47
3440 Eschwege, BRD
Probeexemplar 5,50 DM
Jahresabonnement 55,- DM
Wir bitten um freie Zusen-
dung von 2 kostenlosen
Exemplaren jeder Publika-
tion, die nachgedrucktes Ma-
terial enthält.

Autoren, Manuskripte:
Der Verlag nimmt Manuskripte und Software zur Veröf-
fentlichung gerne entgegen.
Honorare nach Vereinbarung.
Bei Zusendung von Manuskripten und Software erteilt
der Autor dem Verlag die Ge-
nehmigung zum Abdruck und Versand der veröffentlichten
Programme auf Datenträger.
Rücksendung erfolgt nur bei
angeforderten Beiträgen, an-
sonsten nur gegen Erstattung
der Unkosten.

Zusendungen von Software
zur Veröffentlichung sollte
bitte folgendes enthalten:
Kopierfähige Kassette oder
Diskette mit dem Programm,
von Drucker erstelltes Listing
oder Serie von Bildschirmfo-
tos (keine Schreibmaschinen-
Listings), eventuell Bild-
schirmfotos von einem Probe-
lauf.

Homecomputer bringt in diesem Monat:

Leserbriefe	2
News & Trends	3
Texas 99	
Panzerkrieg	5
Panzer bahnen sich den Weg durch unwegsames Gelände.	
TI-UFO	8
Ein Raumflug mit wachsendem Schwierigkeitsgrad	
Killersatellit	9
Der TI-99 versucht den Space-Shuttle des Spielers zu vernichten.	
Sharp MZ-80K	
Eidechse	10
Ein lustiges Spiel. Eine Eidechse muß so viele Fliegen wie mögl. fangen.	
Lesersoftware	
Chiffrier-/Dechiffrier-Programm	12
Der TI-99 als Ver- und Entschlüssler von geheimen Nachrichten.	
Rangliste	13
Der TI-99 sortiert erreichte Punktzahlen bei Sport und Spiel.	
Starship VC-20	
HC-Invaders	14
Ein Basic/Maschinenspracheprogramm für den 16K-VC-20	
Editor für 6502-Systeme	25
Fortsetzung der Reihe "Programmieren in Assembler"	
Labyrinth in 3.5 K	30
Es geht doch! Labyrinth im VC-20 ohne Erweiterung!	
Car-Crash	35
Ein Arkadenspiel bei dem man höllisch schnell schalten muß!	
Software	
Grandmaster - ein Meister seines Fachs.	34
Wir stellen ein interessantes Schachprogramm vor.	
Die ZX-Seiten	
Ritter	37
Ein Ritter (bewegt vom Spieler) versucht, in eine Burg zu gelangen und wird dabei von Pechwolken und wehrhaften Burgverteidigern bedrängt.	
Missile ZX-Command	41
Noch ein Arkadenspiel - die bekannte Städteverteidigung!	
VC-64 VideoChips	
Mondlandung VC-64	44
Ein Programm das es in sich hat. Nicht leicht, den "Adler" zu landen.	
Wurm	46
Ein Wurm kämpft ums Überleben.	
Geisterjäger	48
Ein hübsches Spiel mit der VC-64-Sprite-Grafik.	
VC-64-Sprites-Generator	50
Ein Programm zum einfachen Erzeugen und Speichern von Sprites.	
Apple-Kiste	
Diskschutz für Apple II	52
Disketten vor unbefugten Anwendern schützen.	
Trollhöhlen	55
Mit gefährlichen Bandwürmern, Trollen und sonstigen Ungeheuern.	
Pet-Bytes	
Computerhirn = Mastermind	57
Mastermind 2	59
Zwei Denkspiele - einmal anders.	
PEEKs + POKEs	
Der VC-64-Videochip	62
Serienstart. Blicke in die Bytes des VC-64.	
Korrekturen	64
Software-Service	64

Heft 3 noch lieferbar

Ich bin angenehm beeindruckt von dem Inhalt Ihrer Zeitschrift Homecomputer. Leider habe ich das erste Heft nicht mehr am Kiosk bekommen. Bitte senden Sie mir das Heft 3/83 und die VC20-Kassette Heft 4.

P. Hirsch Köln
Es ist ab Mai noch eine begrenzte Zahl von Heft 3 zu haben. Bitte bestellen Sie über den Verlag.

zu Slalom Heft 3/83

Zu Heft 3 habe ich folgende Frage:
Wie steuert man im Programm Slalom den Fahrer?

Johann Breitenfellner Linz
Die Tasten Z und / steuern den Alpinisten. Das haben wir vergessen abzdrukken.

ZX-Drucke

Mit großem Interesse habe ich Ihre Zeitschrift "Homecomputer" gelesen.

Beim Labyrinth-Programm für den ZX81 treten bei mir jedoch Schwierigkeiten auf. Das Unterprogramm in Maschinensprache, welches in einem Stringausdruck untergebracht ist, ist leider kaum lesbar.

Die Zeichen 8 und B sind teilweise nicht zu unterscheiden. Aus diesem Grund läuft das Unterprogramm bei mir nicht.

Mein Tip für zukünftige Ausgaben: Geben Sie das Programm in Maschinensprache in Dezimal an (wie bei U-Boot-Jagd), da dort eine Verwechslung zwischen Bund 8 nicht auftreten kann.

Thomas Kaus Burladingen 1

Negativ...

Positiv finde ich ...

Negativ fällt es daher auf, daß Eure Zeitschrift eben gar nicht niedriger im Preis rangiert. Sie ist sogar, vergleicht man Preis und Umfang, so ziemlich die teuerste deutsche Zeitschrift ihrer Art! (z.B. kostet Computer persönlich nur 4,- DM!)

Anzeigen in denen sich der Leser über den deutschen Geräte- und Softwaremarkt informieren kann, suchte ich in Homecomputer so gut wie vergeblich. Das ist nicht nur eine inhaltliche Lücke (ich kaufe eine Fachzeitschrift gerade auch wegen der Anzeigen!) - es heißt auch, daß der Käufer die Herstellungskosten alleine zu decken hat!

Franz Nagel Darmstadt

Besitzen Sie einen Homecomputer?
Sicher nicht, sonst hätten Sie das Preis/Leistungsverhältnis (von dem in der Computerbranche oft die Rede ist) schon er-

kannt. Für einen VC-20-Anwender z.B. sind allein die Programme der ersten drei Ausgaben von Homecomputer einige Jahresabos wert - und das bei nur 16.50 DM!

Nicht vollständig?

Wäre es nicht möglich, in zukünftigen Ausgaben alle (auch öde Tippereien wie Apple-Steno) Programme vollständig abzdrukken, da 1 Disk/Monat die Taschengeldgrenze manches Kleincomputeraners haushoch übersteigt.

Wolfgang Schmidt München
Teilen Sie uns bitte mit, welche Seite in Ihrem Heft gefehlt hat. Oder haben Sie etwa nur den Absatz zum Leser-Service auf Seite 33 mißverstanden?

Umsonst geärgert - Handbuch lesen.

Die Märzangabe von Homecomputer hat mich 505,50 DM gekostet! 5,50 für das Heft und für 500,00 DM habe ich mich geärgert. Ich habe nämlich das Robotprogramm (fehlerfrei!) eingetippt und dazu brauche ich ja anscheinend keine 16K-Erweiterung. Trotzdem läuft das Programm nicht. Nachdem die Leertaste gedrückt wurde ist es aus und vorbei, Errors, so weit das Auge reicht. Da ich fehlerfrei getippt habe (das habe ich bis zur "Vergasung" nachgeprüft), glaube ich nicht, daß der Fehler bei mir liegt.

Thomas Schiefer Eggenfelden

Und doch ist es so. Weil die meisten VC-20-Besitzer eine Speichererweiterung um mindestens 8 K besitzen, schreiben wir Programme fast immer für 8- oder 16-K-Versionen.

Daß dann die Adressen für Video- und Farbmatrix angepaßt werden müssen, wenn man weniger als 11.5 K freien Speicher hat, ist aus dem Handbuch zu ersehen.

Der VC-20 ist aber glücklicherweise der einzige Microcomputer, der uns auf diese Art und Weise angeschmiert hat, denn bei anderen Computern stellt sich das Problem nicht.

MID\$ mit Dreien?

Bei Ihrem Kalender-Programm für den PET bin ich mit Ihrer Zeile 135 nicht ganz einverstanden. MID\$ hat doch 3 Argumente. Ich kann leider aus Zeitgründen nicht das ganze Programm durchgehen und möchte Sie bitten den Fehler zu korrigieren.

Stefan Munch Erlensee

Gibt es etwas zu korrigieren? Wer das Computerhobby ernsthaft betreibt, muß sich hin und wieder ein wenig Zeit nehmen, zu lernen, was noch nicht gelernt oder verstanden wurde. Probieren Sie den MID\$ mit 2 Argumenten aus. Sie werden überrascht sein.

Ferien im Computer-Camp

Super Freizeitpaß mit Computer-learning für junge Leute!

Im Luftkurort "Haltern am See" findet am 310 ha großen Trinkwassersee der Stevertalsperre eines der ersten europäischen Computercamps statt. Junge Leute werden hier spielerisch mit dem Mikrocomputer und der Programmiersprache BASIC vertraut gemacht. Durch intensive Beschäftigung mit dem Mikrocomputer

COMMODE (für je zwei Teilnehmer steht eine Commodore-Anlage zur Verfügung) wird die Angschwelle schnell abgebaut und der Umgang mit der EDV-TECHNIK zur Selbstverständlichkeit. Vorgesehen sind: 50 Unterrichtsstunden Computerlearning unter Anleitung von Fachdozenten, umfangreiches Unterrichtsmaterial, Einsatz von Video, zusätzlicher theore-

tischer Unterricht, Einführung in die EDV und Mikrocomputer, sowie deren Möglichkeiten und Grenzen. Neben dem Computerlearning steht der Kontakt der Teens und Twens mit Sport, Spiel und Freizeitpaß im Vordergrund. Das Freizeit-Aktionsprogramm bietet (teilweise gegen geringe Unkostenbeteiligung) Fußball, Schwimmen, Reiten, Fahrradverleih u. a.. Für das Rundumprogramm sind weiterhin vorgesehen: Planwagenfahrten, Nachtwanderungen, Lagerfeuer und heiße Disconächte. Die Unterbringung, die gemeinsamen Mahlzeiten, die Programmierung der Mikrocomputer COMMODE findet in der "Jugendherberge Haltener-Stausee" Overath 39, 4358 Haltern-Stausee, statt.

111 Cassetten im Programm

**BUG-BYTE
SOFTWARE** 
AUTHORISED DEALER

Software-Anbieter für die populären Homecomputer sind zur Zeit noch nicht sehr zahlreich gesät. Einer der größten Anbieter für ZX 81, Spectrum, VC 20, Acorn, BBC und Dragon ist der Softwareversand Augsburg, der jetzt auch den Spectrum-Computer im Programm hat. Produkte der Firmen Artic, Bug Byte, Melbourne House, Publishers Personal Soft-

ware A+F und Impact werden zum Teil exklusiv vertrieben.

Alles mit ausführlicher (allerdings englischer) Beschreibung und original verpackt. Ein Katalog mit dem vorhandenen Programmspektrum ist gegen Gebühr erhältlich.

Softwareversand
Thomas Wagner, Augsburg

Microdrive bald lieferbar.

Noch immer wartet die Homecomputerwelt auf die Fertigstellung des vor 1½ Jahren angekündigten Microdrives.

Laut Sinclair stehen lediglich noch die speziell für die Firma hergestellten Halbleiter-Chips aus, die zur Zeit noch eine Testphase durchlaufen. Das Design steht ebenfalls und in Kürze sollen für £40 per Stück die ersten 100.000 Geräte zunächst an die ersten Spectrum-Kunden geliefert und dann über den Handel auch der Rest der Interessenten bedient werden.

Somit wird dann auch ein Gerücht sein Ende finden,

nämlich, ob es sich um Microflopies oder um "Microbänder", ähnlich den sogenannten "Stringy Floppies", handelt. Laut unseren letzten vorliegenden Informationen handelt es sich um ein Endlosband, das sich mit hoher Geschwindigkeit innerhalb 7 Sekunden einmal am Übertragungskopf vorbeibewegt. So wird eine durchschnittliche Zugriffszeit von 3,5 Sekunden erreicht.

TRS-80/VG-Club in Wuppertal

Zu unserem Aufruf anlässlich der Clubecke im letzten Heft sandte uns Herr Gerd Breidenbruch aus Wuppertal folgende Mitteilung.

Der TRS-80/Mod.1 hat meines Erachtens einen sehr großen Marktanteil und ist immerhin der 'Homecomputer mit dem größten Software-Angebot'

Vielleicht interessiert es Sie in diesem Zusammenhang, daß es in Wuppertal einen TRS-80/Video-Genie Club gibt (Anzahl der Mitglieder zur Zeit ca. 40, eigene Clubzeitschrift, Clubbeitrag keiner).

Die Mitglieder des TRS-80/VG Clubs treffen sich monatlich einmal in der Universität Wuppertal Raum 113.34. Jeder Computer-Fan ist dort herzlich willkommen.

Da die Treffen noch in unregelmäßigen Abständen stattfinden, sollten sich Interessenten erst bei dem Leiter der Gruppe, Herrn Horst Brodowski, Gördelestr. 7, 5657 Haan/Rhl. Tel.: 02129/4891 erkundigen.

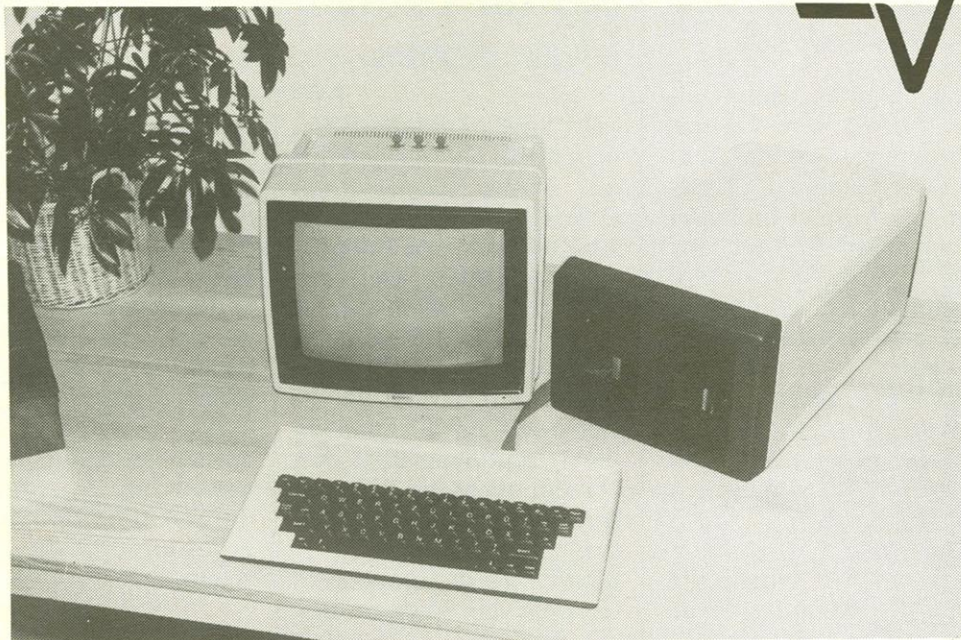
Wir bedanken uns für die freundliche Unterstützung bei der Titelgestaltung beim

Round up Western Shop

Frankfurter Straße 88 · 3500 Kassel

ORANGE 2 COMPUTER (voll APPLE kompatibel)

Groß-/Kleinschreibung, Umlaute bei allen Ausführungen



ARHELGER
electronic

Komplettsystem wie abgebildet

Bausatz: Hauptplatine 48K-RAM und alle Teile, fertiges Netzteil, Cherry-Tastatur (4 Cursortasten, 2 Editiertasten, Hometaste, Gehäuse)

Fertiggerät: wie oben, jedoch fertig aufgebaut,

Orange 2 6 Monate Garantie

Gehäuse für Rechner und 2 Drives auf Anfrage.

Profitaustatur zur Textverarbeitung:

Zehnerblock, Funktionstasten, Cursortasten, Umlaute, anschlußfertig im Gehäuse.

PERIPHERIE:

NEC-8023 BC m. Graphikinterf.

BMC-Monitor Typ 12 grün geätzte Röhre

Zenit-Monitor ZUM 121 (LOW COST)

16K Language Card

80 Zeichenkarte m. 40/80 Z. Softswitch

CP/M Z-80 CPU-Karte

Verbatim Datalife Disketten 5¼" (10)

Diskettenlaufwerke und weiteres Zubehör auf Anfrage!

Alle Preise verstehen sich inkl. 13% MwSt.
Weitere Informationen gegen DM 1,50 in BM.

ab DM **4499,-**

DM **1388,-**

nur DM **1688,-**

Aufpreis nur DM **488,-**

DM **1950,-**

DM **540,-**

DM **305,-**

DM **220,-**

DM **495,-**

DM **375,-**

DM **79,-**

Versand erfolgt unfrei.

3544 WALDECK-SACHSENHAUSEN
TELEFON 0 56 34 - 17 24

WINTERHAGEN 2
99 1160 AEW D

Panzerkrieg

TI-99/4A + Fernbedienungen

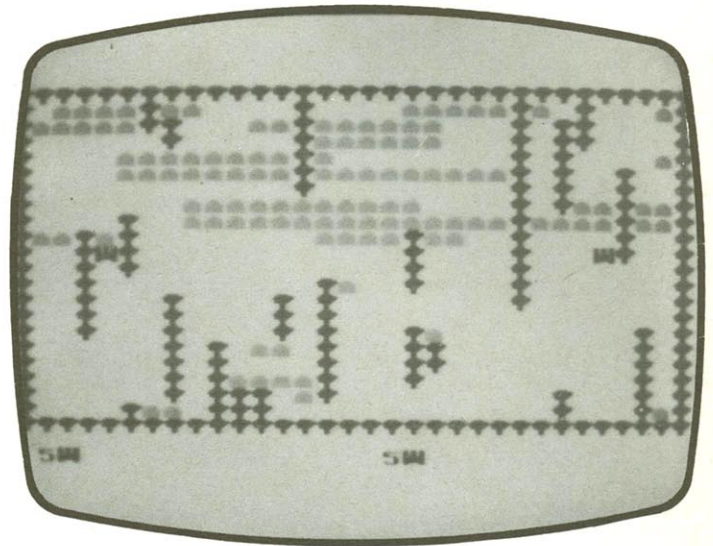
Der Sinn dieses Zwei-Personen-Spieles besteht darin, die gegnerischen Panzer abzuschießen und dabei möglichst wenig Flur- und Sachschaden anzurichten. Am Anfang stehen sich die Feinde gegenüber, getrennt durch Baum- und Häuserreihen. Die Häuser können zwar abgeschossen werden, bringen aber 300 Minuspunkte ein. Will man ein Haus umfahren,

wird der Panzer zerstört und der Gegner bekommt 1000 Pluspunkte. Ebenso bei Abschluß. Die Bäume können weder abgeschossen noch umgefahren werden, "wehren" sich jedoch mit Minuspunkten.
Anm.: Die Zeilen 740-930 verbinden, daß ein Panzer am Anfang von Dämmen umgeben und bewegungsunfähig ist. Sollten die Gegner jedoch durch eine

durchgehende Baumreihe getrennt sein, muß das Spiel mit "CLEAR" abge-

brochen und mit "RUN" neu gestartet werden.

Jürgen Hoppstädter



```

100 1ST 210-
110 1ST 210-
120 1ST 210-
130 1ST 210-
140 1ST 210-
150 1ST 210-
160 1ST 210-
170 1ST 210-
180 1ST 210-
190 1ST 210-
200 1ST 210-
210 1ST 210-
220 1ST 210-
230 1ST 210-
240 1ST 210-
250 1ST 210-
260 1ST 210-
270 1ST 210-
280 1ST 210-
290 1ST 210-
300 1ST 210-
310 1ST 210-
320 1ST 210-
330 1ST 210-
340 1ST 210-
350 1ST 210-
360 1ST 210-
370 1ST 210-
380 1ST 210-
390 1ST 210-
400 1ST 210-
410 1ST 210-
420 1ST 210-
430 1ST 210-
440 1ST 210-
450 1ST 210-
460 1ST 210-
470 1ST 210-
480 1ST 210-
490 1ST 210-
500 1ST 210-

```

```

510 X=23+RND+1
520 Y=31+RND+1
530 Z=7+RND+1
540 CALL VCHAR(X,Y,144,Z)
550 NEXT I

```

```

220 CALL CHAR(138,"C3FFFFFFFF")
230 CALL CHAR(139,"FFFF1EFEF")
240 CALL CHAR(140,"00000181")
250 CALL CHAR(141,"00003C242")
260 CALL CHAR(142,"007E42424")
270 CALL CHAR(143,"FF8181818")
280 CALL CHAR(144,"3C7FFF7E7")
290 CALL CHAR(152,"00387CFEF")
300 CALL CHAR(153,"00003C242")
310 CALL CHAR(154,"007E42424")
320 CALL CHAR(155,"FF8181818")
330 REM FARBEN DEFINIEREN

```

```

560 1ST 560-750
570 REM RAND/PUNKTETABELLE
580 CALL HCHAR(24,1,32,32)
590 CALL HCHAR(23,1,32,32)
600 REM
610 REM
620 REM
630 REM
640 REM
650 REM
660 REM
670 REM
680 REM
690 REM
700 REM
710 REM
720 REM
730 REM
740 REM
750 IF ZP1<>144 THEN 840

```

```

340 1ST 340-550
350 CALL SCREEN(15)
360 CALL COLOR(13,5,1)
370 CALL COLOR(14,14,1)
380 CALL COLOR(15,13,1)
390 CALL COLOR(16,10,1)
400 CALL CLEAR
410 REM
420 REM
430 REM
440 REM
450 REM
460 REM
470 REM
480 REM
490 REM
500 REM

```

```

760 CALL GCHAR(XP1,YP1+1,ZP1)
770 IF ZP1<>144 THEN 840
780 CALL GCHAR(XP1+1,YP1,ZP1)
790 IF ZP1<>144 THEN 840
800 CALL GCHAR(XP1,YP1-1,ZP1)
810 IF ZP1<>144 THEN 840
820 YP1=YP1+1
830 GOTO 740
840 CALL GCHAR(XP2-1,YP2,ZP2)
850 IF ZP2<>144 THEN 940
860 CALL GCHAR(XP2,YP2+1,ZP2)
870 IF ZP2<>144 THEN 940
880 CALL GCHAR(XP2+1,YP2,ZP2)
890 IF ZP2<>144 THEN 940
900 CALL GCHAR(XP2,YP2-1,ZP2)
910 IF ZP2<>144 THEN 940

```

```

920 YP2=YP2-1
930 GOTO 840
940 CALL HCHAR(XP1,YP1,128)
950 CALL HCHAR(XP2,YP2,136)
960 REM SPIELHAUPTPROGRAMM
970 CALL GCHAR(XP1,YP1,ZP1)
980 CALL JOYST(1,X1,Y1)
990 IF Y1<>4 THEN 1010
1000 GOSUB 1240

```

Texas 99

```

1000 IF SUB1 <> 4 THEN 1030
1001 IF SUB1 <> 1380 THEN 1050
1002 IF SUB1 <> 1540 THEN 1070
1003 IF SUB1 <> 15440 THEN 1100
1004 IF CALL KEY <> 2, K2, S2) THEN 1100
1005 IF CALL KEY <> 1, K1, S1) THEN 1130
1006 IF CALL KEY <> 2, K2, S2) THEN 1130
1007 CALL 6CHAR <XP2, YP2, ZP2)

```

```

1008 CALL JYST <2, X2, Y2)
1009 IF <> 4 THEN 1170
1010 IF <> 17470 THEN 1190
1011 IF <> 19110 THEN 1210
1012 IF <> 0440 THEN 1230
1013 IF <> 20170 THEN 1230
1014 REM FORTBEW. PANZER 1

```

```

1015 REM NACH DBEN
1016 IF ZP S2=128 THEN 1300
1017 CALL SOUND <-1000, -8, 5)
1018 CALL HCHAR <XP1, YP1, 128)
1019 CALL 6CHAR <XP1-1, YP1, H1)
1020 IF H1 = 144 THEN 2300
1021 IF CALL HCHAR <XP1, YP1, 32)
1022 IF H1 = 152 THEN 2340
1023 XP1 = XP1 - 1

```

```

1024 CALL SOUND <-1000, -8, 5)
1025 CALL HCHAR <XP1, YP1, 128)
1026 REM NACH RECHTS
1027 IF ZP S2=129 THEN 1430
1028 CALL SOUND <-1000, -8, 5)
1029 CALL HCHAR <XP1, YP1, 129)
1030 CALL 6CHAR <XP1, YP1+1, H1)
1031 IF H1 = 144 THEN 2300
1032 IF CALL HCHAR <XP1, YP1, 32)
1033 IF H1 = 152 THEN 2340
1034 CALL SOUND <-1000, -8, 5)
1035 CALL HCHAR <XP1, YP1, 129)
1036 REM NACH UNTEN
1037 IF ZP S2=130 THEN 1560
1038 CALL SOUND <-1000, -8, 5)
1039 CALL HCHAR <XP1, YP1, 130)
1040 CALL 6CHAR <XP1+1, YP1, H1)

```

```

1041 CALL 6CHAR <XP1+1, YP1, H1)
1042 IF H1 = 144 THEN 2300
1043 IF CALL HCHAR <XP1, YP1, 32)
1044 IF H1 = 152 THEN 2340
1045 XP1 = XP1 + 1
1046 CALL SOUND <-1000, -8, 5)
1047 CALL HCHAR <XP1, YP1, 130)
1048 REM NACH UNTEN
1049 IF ZP S2=131 THEN 1690
1050 CALL SOUND <-1000, -8, 5)
1051 CALL HCHAR <XP1, YP1, 131)
1052 CALL 6CHAR <XP1, YP1-1, H1)
1053 IF H1 = 144 THEN 2300
1054 IF CALL HCHAR <XP1, YP1, 32)
1055 IF H1 = 152 THEN 2340
1056 XP1 = XP1 - 1
1057 CALL SOUND <-1000, -8, 5)
1058 CALL HCHAR <XP1, YP1, 131)
1059 RETURN

```

```

1060 REM FORTBEW. PANZER 2
1061 IF ZP S2=136 THEN 1830
1062 CALL SOUND <-1000, -4, 5)
1063 CALL HCHAR <XP2, YP2, 136)
1064 CALL 6CHAR <XP2-1, YP2, H2)
1065 IF H2 = 144 THEN 2480
1066 IF CALL HCHAR <XP2, YP2, 32)
1067 IF H2 = 152 THEN 2520
1068 XP2 = XP2 - 1
1069 CALL SOUND <-1000, -4, 5)
1070 CALL HCHAR <XP2, YP2, 136)

```

```

1071 RETURN
1072 REM NACH RECHTS
1073 IF ZP S2=137 THEN 1960
1074 CALL SOUND <-1000, -4, 5)
1075 CALL HCHAR <XP2, YP2, 137)
1076 REM NACH UNTEN
1077 IF ZP S2=138 THEN 2090
1078 CALL SOUND <-1000, -4, 5)
1079 CALL HCHAR <XP2, YP2, 138)
1080 CALL 6CHAR <XP2+1, YP2, H2)
1081 IF H2 = 144 THEN 2480
1082 IF CALL HCHAR <XP2, YP2, 32)
1083 IF H2 = 152 THEN 2520
1084 XP2 = XP2 + 1
1085 CALL SOUND <-1000, -4, 5)
1086 CALL HCHAR <XP2, YP2, 138)
1087 REM NACH LINKS
1088 IF ZP S2=139 THEN 2220
1089 CALL SOUND <-1000, -4, 5)

```

```

1090 CALL HCHAR <XP2, YP2, 32)
1091 IF H2 = 152 THEN 2520
1092 CALL SOUND <-1000, -4, 5)
1093 CALL HCHAR <XP2, YP2, 137)
1094 REM NACH UNTEN
1095 IF ZP S2=138 THEN 2090
1096 CALL SOUND <-1000, -4, 5)
1097 CALL HCHAR <XP2, YP2, 138)
1098 CALL 6CHAR <XP2+1, YP2, H2)
1099 IF H2 = 144 THEN 2480
1100 IF CALL HCHAR <XP2, YP2, 32)
1101 IF H2 = 152 THEN 2520
1102 XP2 = XP2 + 1
1103 CALL SOUND <-1000, -4, 5)
1104 CALL HCHAR <XP2, YP2, 138)
1105 REM NACH LINKS
1106 IF ZP S2=139 THEN 2220
1107 CALL SOUND <-1000, -4, 5)

```

```

1108 CALL HCHAR <XP2, YP2, 139)
1109 CALL 6CHAR <XP2, YP2-1, H2)
1110 IF H2 = 144 THEN 2480
1111 IF CALL HCHAR <XP2, YP2, 32)
1112 IF H2 = 152 THEN 2520
1113 CALL SOUND <-1000, -4, 5)
1114 CALL HCHAR <XP2, YP2, 139)
1115 REM NACH UNTEN
1116 IF ZP S2=140 THEN 2440
1117 CALL SOUND <500, 110, 5)
1118 CALL HCHAR <XP2, YP2, 140)
1119 CALL 6CHAR <XP2, YP2, 139)
1120 IF H2 = 144 THEN 2480
1121 IF CALL HCHAR <XP2, YP2, 32)
1122 IF H2 = 152 THEN 2520
1123 CALL SOUND <-1000, -4, 5)
1124 CALL HCHAR <XP2, YP2, 139)
1125 REM NACH UNTEN
1126 IF ZP S2=141 THEN 2440
1127 CALL SOUND <500, 110, 5)
1128 CALL HCHAR <XP2, YP2, 141)
1129 CALL 6CHAR <XP2, YP2, 140)
1130 IF H2 = 144 THEN 2480
1131 IF CALL HCHAR <XP2, YP2, 32)
1132 IF H2 = 152 THEN 2520
1133 CALL SOUND <-1000, -4, 5)
1134 CALL HCHAR <XP2, YP2, 141)
1135 REM NACH UNTEN
1136 IF ZP S2=142 THEN 2440
1137 CALL SOUND <500, 110, 5)
1138 CALL HCHAR <XP2, YP2, 142)
1139 CALL 6CHAR <XP2, YP2, 141)
1140 IF H2 = 144 THEN 2480
1141 IF CALL HCHAR <XP2, YP2, 32)
1142 IF H2 = 152 THEN 2520
1143 CALL SOUND <-1000, -4, 5)
1144 CALL HCHAR <XP2, YP2, 142)
1145 REM NACH UNTEN
1146 IF ZP S2=143 THEN 2440
1147 CALL SOUND <500, 110, 5)
1148 CALL HCHAR <XP2, YP2, 143)
1149 CALL 6CHAR <XP2, YP2, 142)
1150 IF H2 = 144 THEN 2480
1151 IF CALL HCHAR <XP2, YP2, 32)
1152 IF H2 = 152 THEN 2520
1153 CALL SOUND <-1000, -4, 5)
1154 CALL HCHAR <XP2, YP2, 143)
1155 REM NACH UNTEN
1156 IF ZP S2=144 THEN 2440
1157 CALL SOUND <500, 110, 5)
1158 CALL HCHAR <XP2, YP2, 144)
1159 CALL 6CHAR <XP2, YP2, 143)

```

```

1160 PP PANZER 1-1
1161 IF ZP S2=145 THEN 4620
1162 CALL SOUND <500, 110, 5)
1163 CALL HCHAR <XP2, YP2, 145)
1164 CALL 6CHAR <XP2, YP2, 144)
1165 IF ZP S2=146 THEN 4620
1166 CALL SOUND <500, 110, 5)
1167 CALL HCHAR <XP2, YP2, 146)
1168 CALL 6CHAR <XP2, YP2, 145)
1169 IF ZP S2=147 THEN 4620
1170 CALL SOUND <500, 110, 5)
1171 CALL HCHAR <XP2, YP2, 147)
1172 CALL 6CHAR <XP2, YP2, 146)
1173 IF ZP S2=148 THEN 4620
1174 CALL SOUND <500, 110, 5)
1175 CALL HCHAR <XP2, YP2, 148)
1176 CALL 6CHAR <XP2, YP2, 147)
1177 IF ZP S2=149 THEN 4620
1178 CALL SOUND <500, 110, 5)
1179 CALL HCHAR <XP2, YP2, 149)
1180 CALL 6CHAR <XP2, YP2, 148)
1181 IF ZP S2=150 THEN 4620
1182 CALL SOUND <500, 110, 5)
1183 CALL HCHAR <XP2, YP2, 150)
1184 CALL 6CHAR <XP2, YP2, 149)
1185 IF ZP S2=151 THEN 4620
1186 CALL SOUND <500, 110, 5)
1187 CALL HCHAR <XP2, YP2, 151)
1188 CALL 6CHAR <XP2, YP2, 150)
1189 IF ZP S2=152 THEN 4620
1190 CALL SOUND <500, 110, 5)
1191 CALL HCHAR <XP2, YP2, 152)
1192 CALL 6CHAR <XP2, YP2, 151)
1193 IF ZP S2=153 THEN 4620
1194 CALL SOUND <500, 110, 5)
1195 CALL HCHAR <XP2, YP2, 153)
1196 CALL 6CHAR <XP2, YP2, 152)
1197 IF ZP S2=154 THEN 4620
1198 CALL SOUND <500, 110, 5)
1199 CALL HCHAR <XP2, YP2, 154)
1200 CALL 6CHAR <XP2, YP2, 153)

```

```

1201 CALL 6CHAR <XP1, YP1, ZP1)
1202 IF ZP S2=155 THEN 2720
1203 CALL SOUND <500, 110, 5)
1204 CALL HCHAR <XP1, YP1, 155)
1205 CALL 6CHAR <XP1, YP1, 154)
1206 IF ZP S2=156 THEN 2720
1207 CALL SOUND <500, 110, 5)
1208 CALL HCHAR <XP1, YP1, 156)
1209 CALL 6CHAR <XP1, YP1, 155)
1210 IF ZP S2=157 THEN 2720
1211 CALL SOUND <500, 110, 5)
1212 CALL HCHAR <XP1, YP1, 157)
1213 CALL 6CHAR <XP1, YP1, 156)
1214 IF ZP S2=158 THEN 2720
1215 CALL SOUND <500, 110, 5)
1216 CALL HCHAR <XP1, YP1, 158)
1217 CALL 6CHAR <XP1, YP1, 157)
1218 IF ZP S2=159 THEN 2720
1219 CALL SOUND <500, 110, 5)
1220 CALL HCHAR <XP1, YP1, 159)
1221 CALL 6CHAR <XP1, YP1, 158)
1222 IF ZP S2=160 THEN 2720
1223 CALL SOUND <500, 110, 5)
1224 CALL HCHAR <XP1, YP1, 160)
1225 CALL 6CHAR <XP1, YP1, 159)
1226 IF ZK1 <> 144 THEN 2850

```


Texas 99

TI-UFO

Ziel des Spieles ist es, ein Ufo um Hindernisse zu lenken. Die horizontale Steuerung erfolgt mit den Tasten A (nach links) und L (nach rechts). Wird keine Taste gedrückt, so bewegt

sich das Ufo vertikal nach unten. Wird ein Hindernis senkrecht überfahren, so werden 2 Fehlerpunkte berechnet, beim waagrechten Überfahren 1 Fehlerpunkt. Ebenso werden beim waagrechten Zurückfahren und bei Erreichen des Bildschirmrandes Fehlerpunkte errechnet.

Die ersten drei Runden be-

stehen aus drei Durchgängen. Jede weitere Runde aus zwei Durchgängen. Der Schwierigkeitsgrad nimmt ständig zu. Werden in einer Runde weniger als 8 Fehlerpunkte verursacht, so kommt man in die nächste Runde (Freispiel). Der Schwierigkeitsgrad kann in Zeile 170, der Schwierigkeitszuwachs in der Zeile

770 verstellt werden.

Runden 1-3: # \$ %

Runden 4 - 8: A B C D E

Runden 9 - : Zufallszeichen

Nachteil des Programms: Das Ufo kann aus Zeitgründen nicht gelöscht werden. Es bildet sich deshalb eine Ufoschlange.

Hans-Peter Adrion



```
100 CALL SCREEN(14)
110 CALL COLOR(1,16,14)
120 BU=1
130 JU=0
140 FE=0
150 ZE=35
160 REM HINDERNISSE PRO ZEIL
E
170 PIT=5
180 SC=0
190 FOR T=3 TO BU STEP -1
200 IF FE>=8 THEN 1050
210 CALL CLEAR
220 G=0
230 RANDOMIZE
240 GOSUB 560
250 IF FE>=8 THEN 1050
260 CALL SOUND(400,131,2,165
,2,196,2)
270 REM **UFO SETZEN
280 SPAL=16
290 CALL CHAR(130,"000000183
C7E7E99")
300 G=G+1
310 CALL GCHAR(G,SPAL,WE)
320 IF WE>33 THEN 330 ELSE 3
60
330 CALL SOUND(100,-2,1)
340 CALL COLOR(1,5,14)
350 SC=SC+2
360 CALL HCHAR(G,SPAL,130)
370 IF G=24 THEN 660
380 CALL KEY(0,KEY,STATUS)
390 IF STATUS=0 THEN 300
400 IF KEY=65 THEN 460
410 IF KEY=76 THEN 430
420 GOTO 300
430 SPAL=SPAL+1
440 IF SPAL=33 THEN 460
450 GOTO 480
460 SPAL=SPAL-1
470 IF SPAL=0 THEN 430
480 CALL GCHAR(G,SPAL,0)
490 IF Q>33 THEN 500 ELSE 53
0
500 CALL SOUND(100,-1,1)
510 CALL COLOR(1,16,14)
520 SC=SC+1
530 CALL HCHAR(G,SPAL,130)
540 GOTO 380
550 REM **HINDERNISSE SETZEN
560 Z=3
570 FOR M=1 TO 100
580 FOR C=1 TO PIT
590 X=INT(32*RND)+1
600 IF Z>24 THEN 650
610 CALL HCHAR(Z,X,ZE)
620 NEXT C
630 Z=Z+T
640 NEXT M
650 RETURN
660 CALL SOUND(2500,131,3,17
5,3,220,3)
670 NEXT T
680 CALL CLEAR
690 FE=FE+1
700 PRINT
710 REM **ASCII-CODE
720 PRINT
730 ZE=ZE+1
740 PRINT
750 PRINT
760 REM HINDERNISZUWACHS
770 PIT=PIT+4
780 REM AUSGABE
790 PRINT "FEHLERPUNKTE :";S
C
800 JU=JU+1
810 IF JU=3 THEN 1020
820 REM FREISPIEL SETZEN!
830 IF SC<=8 THEN 840 ELSE 9
30
840 PRINT
850 IF SC=0 THEN 860 ELSE 87
0
860 PRINT " O P T I M U M !"
870 PRINT
880 PRINT " ***FREISPIEL***"
890 REM PAUSE
900 FOR V=1 TO 800
910 NEXT V
920 GOTO 180
930 PRINT
940 PRINT "C BY HOMECOMPUTER
"
950 PRINT
960 PRINT "NOCH EIN SPIEL? (
J/N)"
970 CALL KEY(0,KEY,STATUS)
980 IF STATUS=0 THEN 970
990 IF KEY=74 THEN 120
1000 STOP
1010 REM ZEICHENSETZUNG AB
RUNDE 8
1020 BU=2
1030 ZE=65
1040 GOTO 830
1050 CALL CLEAR
1060 TR=TR+50
1070 FOR OP=1 TO TR
1080 WA=INT(120*RND)+1
1090 WS=INT(22*RND)+3
1100 WD=INT(32*RND)+1
1110 CALL HCHAR(WS,WD,WA)
1120 NEXT OP
1130 G=0
1140 GOTO 260
1150 END
```



Killersatellit

Sobald das Spiel gestartet wurde, fragt der Computer nach der Wahl des Schiffes. Es stehen 4 Schiffe zur

Wahl, die sich jedoch nur äußerlich unterscheiden. Die Steuerung des Schiffes erfolgt über die 4 Cursor-

Tasten (E, S, D und X). Über die Space-Taste können Schüsse abgegeben werden. Nach jedem erfolgtem Abschluß erhöht sich die Schwierigkeit und Punktzahl der einzelnen Gegner. Außerdem wird nach je-

dem Abschluß die eigene, sowie die Höchstpunktzahl angezeigt.

Das Spiel ist beendet, wenn der Satellit das Schiff abgeschossen hat. Alles Andere ergibt sich beim Spielen.

Ch. Bladoschewski

```

10 REM *****
20 REM * KILLERSATELLIT *
30 REM * ENTWICKELT VON *
40 REM * CH. BLADOSCHEWSKI *
50 REM *****
100 CALL CLEAR
110 CALL COLOR(1,2,2)
120 CALL SCREEN(2)
130 PRINT TAB(7);"KILLERSATE
LLIT": : : : : : : : :
140 CALL CHAR(96,"00183C7E7E
E3C18")
150 CALL CHAR(104,"00183C7E7
E3C18")
160 CALL HCHAR(10,6,104,21)
170 CALL HCHAR(14,6,104,21)
180 CALL VCHAR(10,6,104,5)
190 CALL VCHAR(10,26,104,5)
200 FOR I=6 TO 26 STEP 2
210 CALL HCHAR(10,I,96)
220 CALL HCHAR(14,I,96)
230 NEXT I
240 CALL HCHAR(12,6,96)
250 CALL HCHAR(12,26,96)
260 CALL COLOR(1,16,1)
270 CALL SCREEN(16)
280 FOR I=1 TO 20
290 FOR J=1 TO 2
300 CALL COLOR(9,11,9)
310 CALL COLOR(10,5,13)
320 NEXT J
330 CALL COLOR(9,5,9)
340 CALL COLOR(10,11,13)
350 NEXT I
360 CALL COLOR(1,1,3)
370 CALL SCREEN(3)
380 REM
390 SC=0
400 CALL CLEAR
410 RANDOMIZE
420 M=2
430 P1=1
440 U=.1
450 A$="10101038BABAFE82"
460 R$="42425A665A4242"
470 C$="183C7E3C3C7EFF"
480 D$="82B2B20082B2B2"
490 E$="1818181818181818"
500 B$="1818183C66FFFF66"
510 G$="42425A5A7EFF42"
520 INPUT "WELCHES SCHIFF WI
LLST DU (1-4)?" : SCHIFF
530 IF (SCHIFF<1)+(SCHIFF>4)
<0 THEN 520
540 ON SCHIFF GOTO 640,580,6
10,550
550 Z$=R$
560 Y$=E$
570 GOTO 670
580 Z$=B$
590 Y$=E$
600 GOTO 670
610 Y$=D$

```

```

620 Z$=G$
630 GOTO 660
640 REM
650 Z$=A$
660 Y$=D$
670 X=10
680 Y=10
690 CALL CLEAR
700 B=12
710 A=16
720 CALL SCREEN(2)
730 CALL COLOR(1,1,1)
740 CALL COLOR(9,4,1)
750 CALL COLOR(10,14,1)
760 CALL COLOR(11,9,1)
770 CALL COLOR(12,5,1)
780 CALL CHAR(96,Y$)
790 CALL CHAR(104,"5A5A5A5A5
A5A5A5A")
800 CALL CHAR(112,C$)
810 CALL CHAR(120,Z$)
820 X=X-(INT((RND*2)-.5))
830 Y=Y-(INT((RND*2)-.5))
840 IF X<1 THEN 1240
850 IF X>24 THEN 1260
860 IF Y<1 THEN 1280
870 IF Y>32 THEN 1300
880 CALL KEY(0,K,S)
890 IF K=32 THEN 1320
900 IF K=83 THEN 1040
910 IF K=68 THEN 1070
920 IF K=88 THEN 1100
930 IF K=69 THEN 1130 ELSE 9
80
940 IF B<1 THEN 1160
950 IF B>24 THEN 1180
960 IF A<1 THEN 1200
970 IF A>32 THEN 1220
980 CALL CLEAR
990 CALL HCHAR(B,A,120)
1000 CALL HCHAR(X,Y,112)
1010 IF X>21 THEN 820
1020 IF RND<U THEN 1580
1030 GOTO 820
1040 A=A-1
1050 CALL SOUND(-500,-6,0)
1060 GOTO 940
1070 A=A+1
1080 CALL SOUND(-500,-6,0)
1090 GOTO 940
1100 B=B+1
1110 CALL SOUND(-500,-6,0)
1120 GOTO 940
1130 B=B-1
1140 CALL SOUND(-500,-6,0)
1150 GOTO 940
1160 B=B+24
1170 GOTO 980
1180 B=B-24
1190 GOTO 980
1200 A=A+32
1210 GOTO 980
1220 A=A-32

```

```

1230 GOTO 980
1240 X=X+24
1250 GOTO 880
1260 X=X-24
1270 GOTO 880
1280 Y=Y+32
1290 GOTO 880
1300 Y=Y-32
1310 GOTO880
1320 FOR P=B-1 TO B-M STEP -
1
1330 CALL SOUND(100,-5,0)
1340 CALL GCHAR(P,A,L)
1350 IF L=112 THEN 1390
1360 CALL HCHAR(P,A,96)
1370 NEXT P
1380 GOTO 980
1390 CALL CLEAR
1400 PRINT "GEGNER ZERSTOERT
": : : : :
1410 SC=SC+P1
1420 IF HS>=SC THEN 1490
1430 HS=SC
1440 FOR SI=1 TO 5
1450 FOR SJ=-3 TO -1
1460 CALL SOUND(100,SJ,0)
1470 NEXT SJ
1480 NEXT SI
1490 PRINT "HIGHSCORE:";HS
1500 PRINT "PUNKTZAHL:";SC:
: : : : : : : : :
1510 CALL COLOR(1,3,3)
1520 CALL SCREEN(3)
1530 P1=P1+1
1540 U=U+.02
1550 FOR I=1 TO 1000
1560 NEXT I
1570 GOTO 540
1580 FOR P=X+1 TO X+3
1590 CALL SOUND(100,-3,2)
1600 CALL GCHAR(P,Y,L)
1610 IF L=120 THEN 1650
1620 CALL HCHAR(P,Y,104)
1630 NEXT P
1640 GOTO 820
1650 CALL CLEAR
1660 PRINT TAB(5);"DU WURDES
T ZERSTOERT": : : "
GAME OVER": : :
1670 PRINT "HIGHSCORE:";HS
1680 PRINT "PUNKTZAHL:";SC:
: : : : : : :
1690 CALL COLOR(1,3,3)
1700 CALL SCREEN(3)
1710 PRINT "WILLST DU NOCH E
INMAL SPIE- LEN? DRUECKE <J>
<N>"
1720 CALL KEY(0,K,S)
1730 IF S=0 THEN 1720
1740 IF K=ASC("J") THEN 380
1750 IF K<>ASC("N") THEN 1710
1760 END

```

Eidechse Sharp MZ-80K

Das Programm EIDECHSE wurde auf einem Sharp MZ-80 K geschrieben und ist ungefähr 5,5 KB lang. Die Grundidee ist einem Buch entnommen und wurde von mir erweitert und an den MZ-80 K angepaßt.

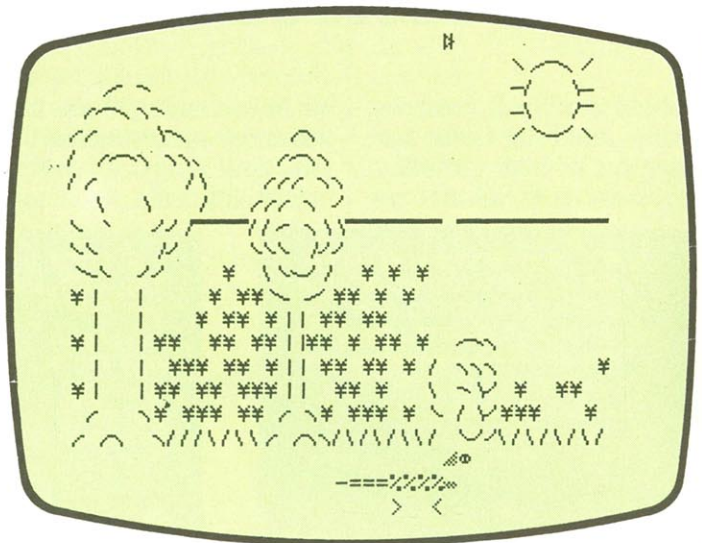
Zum Verlauf des Spieles sei gesagt, es ist ein sehr einfaches Spiel.

Der Spieler soll versuchen, mittels einer beliebigen

Taste die Fliege zu fangen. Auf Tastendruck schnellt die Zunge der Eidechse nach oben und mit etwas Glück oder Geschick trifft er die Fliege.

Nach einer gewissen Trefferquote gibt es einen Zeitbonus. Die Zeit, Treffer und Durchgänge werden dem Spieler auf dem Bildschirm angezeigt.

Egon Kwapil



Die Eidechse mit der Fliege links von der Sonne

```

10 PRINT"EGON KWAPIL   HAMBURG 03.1983"
20 MUSIC"R7R7R7"
30 PRINT"
40 PRINT"
50 PRINT"
60 PRINT"
70 PRINT"
80 PRINT"
90 PRINT"
100 PRINT"
110 PRINT"
120 USR(68):NEXTA:PRINT"
130 PRINT"
140 PRINT"
150 PRINT"
160 PRINT"
170 PRINT"
180 PRINT"
190 PRINT"
200 PRINT"
210 PRINT"
220 PRINT"
230 PRINT"
240 PRINT"
250 PRINT"
260 A$=" CR3 DR3 ER3 DR3 DR3 CR3 DR3 ER3 CR3 DR3 ER3 ER3 DR3 DR3 C5"
270 B$=" CR3 DR3 ER3 DR3 DR3 CR3 DR3 ER3 CR3 DR3 ER3 ER3 DR3 DR3 C5"
280 C$=" CR3 DR3 ER3 DR3 DR3 CR3 DR3 ER3 CR3 DR3 ER3 ER3 DR3 DR3 C5"
290 TEMPO7:MUSICA$,B$,A$,C$
300 PRINT"
310 PRINT"

```


Chiffrier/ Dechiffrier- Programm

TI 99/4A + Drucker

Eingabe

Das Pgm fragt zunächst, ob dechiffriert werden soll, - dann nur DE eingeben (wird im Pgm auch gesagt) Ansonsten verlangt es die Eingabe einer vierstelligen Codezahl (z.B. Datum in der Form 3103 = 31.03.) Diese Codezahl muß auch dem Entschlüssler bekannt sein.

Das Pgm nimmt nur Zahlen an und zwar nur eine! Es sind also alle vier ein-

zeln einzugeben. Danach wird die Eingabe einer zu verschlüsselnden Zeile verlangt. Das Pgm nimmt nur 28 Zeichen an und sagt an, daß der Zeilenrest ggfls mit Punkten zu füllen ist. Die Eingabe geschieht ganz normal und der Drucker druckt sie verschlüsselt aus. So kann Zeile für Zeile eingegeben werden. Die letzte Eingabe soll eine 0 sein, damit das Pgm erkennt, daß die Ein-

gabe abgeschlossen ist.

Das Dechiffrieren geschieht genauso, nur daß hier eben der chiffrierte Text eingegeben wird, der dann Zeile für Zeile dechiffriert herauskommt.

Interessant und daher kaum zu „knacken“ ist der Algorithmus des Pgms hinsichtlich der Verschlüsselung. Manchmal ist der gleiche Buchstabe von gleicher Bedeutung, manchmal haben 2 gleiche Buchstaben eine völlig andere Bedeutung und manchmal werden für einen Buchstaben im Laufe eines längeren Textes bis zu 50 andere eingesetzt.

Programmbeispiel:

Eingegeben wurde zeilenweise

DIES IST EIN CHIFFRIERTEST..
AUFGENOMMEN AM

31.03.83 FÜR DIE ZEITSCHRIFT HOMECOMPUTER

Da die Eingabe automatisch auf 28 Zeichen beschränkt ist, können Sie am Zeilenende jeweils auf dem . stehenbleiben, der Computer meldet sich mit einem BEEP.

So sah der Ausdruck aus:

```
AHEP HSO DIK BHFCERFBQTBPS.+  
>TFDBMOJJDN >L 0.-00+73 CTED  
AHE WDIQPBHOFET ENMB@NMRSED
```

und als der vorige Ausdruck Zeichen für Zeichen, wiederum zeilenweise nach vorheriger Eingabe der Codeziffern eingegeben wurde, kam folgender Ausdruck zustande:

```
DIES IST EIN CHIFFRIERTEST..  
AUFGENOMMEN AM 31.03.83 FUER  
DIE ZEITSCHRIFT HOMECOMPUTER
```

```
100 CALL CLEAR :: CALL SCREEN(16  
)  
110 DIM B$(30)  
120 DIM B(30)  
130 INPUT "ZUM DECHIFFRIEREN DE  
EINGEBEN":DE$ :: IF DE$="DE" THE  
N 370  
140 PRINT "GIB JETZT EINZELN DIE  
4 CODEZIFFERN EIN:"  
150 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):W  
160 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):X  
170 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):Y  
180 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):Z  
190 PRINT "TEXTZEILE EINGEBEN (<br>  
BIS 28. STELLE MIT PUNKTEN FUELLE  
N):"  
200 ACCEPT BEEP SIZE(28):A$  
210 IF A$="0" THEN END  
220 FOR I=1 TO 28  
230 B(I)=ASC(SEG$(A$,I,1))  
240 NEXT I  
250 FOR I=1 TO 28 STEP 4 :: B(I)  
=B(I)-W :: NEXT I  
260 FOR I=2 TO 28 STEP 4 :: B(I)  
=B(I)-X :: NEXT I  
270 FOR I=3 TO 28 STEP 4 :: B(I)  
=B(I)-Y :: NEXT I  
280 FOR I=4 TO 28 STEP 4 :: B(I)
```

```
=B(I)-Z :: NEXT I  
290 FOR I=1 TO 28  
300 IF B(I)<=32 THEN B(I)=32  
310 NEXT I  
320 FOR I=1 TO 28 :: B$(I)=CHR$(  
B(I)):: NEXT I  
330 OPEN #1:"TP.E",OUTPUT  
340 FOR I=1 TO 28 :: PRINT #1:B$(  
I):: NEXT I  
350 CLOSE #1  
360 GOTO 200  
370 PRINT "GIB JETZT EINZELN DIE  
4 CODEZIFFERN EIN:"  
380 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):W  
390 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):X  
400 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):Y  
410 ACCEPT BEEP VALIDATE (DIGIT)S  
IZE(1):Z  
420 PRINT "GIB DEN ZEILENTEXT EI  
N":A$  
430 ACCEPT BEEP SIZE(28):A$  
440 IF A$="0" THEN END  
450 FOR I=1 TO 28  
460 B(I)=ASC(SEG$(A$,I,1))  
470 NEXT I  
480 FOR I=1 TO 28 STEP 4 :: B(I)  
=B(I)+W :: NEXT I  
490 FOR I=2 TO 28 STEP 4 :: B(I)  
=B(I)+X :: NEXT I
```

```

500 FOR I=3 TO 28 STEP 4 :: B(I)
=B(I)+Y :: NEXT I
510 FOR I=4 TO 28 STEP 4 :: B(I)
=B(I)+Z :: NEXT I
520 FOR I=1 TO 28
530 IF B(I)<=41 THEN B(I)=32
540 NEXT I
550 FOR I=1 TO 28 :: B$(I)=CHR$(

```

```

B(I)):: NEXT I
560 OPEN #1:"TP.E",OUTPUT
570 FOR I=1 TO 28 :: PRINT #1:B$(
I);:: NEXT I
580 CLOSE #1
590 GOTO 430
600 END

```

Rangliste

TI 99/4A + Drucker

Aufgrund einer Wahl oder eines Wettbewerbes soll im Verein z. B. eine Rangliste der Ergebnisse aufgestellt werden.

Die Verarbeitungszeit für 100 Teilnehmer beträgt ca. 5 min.

Das Programm fragt in ei-

ner Dreiergruppe für jeden Teilnehmer:
 LISTENPLATZ
 NAME
 ERGEBNIS (in Punkten oder Stimmen)
 Sind die Angaben für alle Teilnehmer eingegeben, erwartet das Pgm zum Abschluß unter LISTENPLATZ die Eingabe einer 0

und beginnt die Verarbeitung.

Ausgabeform:

1. (5.) Mayerhofer = 189
 dabei ist die erste Zahl der Ranglistenplatz die in () stehende der bisherige Listenplatz und nach dem Namen folgt das Ergebnis aus den Punkten/Stimmen.
 Das Pgm stoppt - zur Verlesung der Ergebnisse - automatisch nach der Ausgabe von jeweils 5 Ergebnissen und kann dann durch Drücken irgendeiner Taste

fortgesetzt werden. Es erfolgt die Ausgabe eines kompletten Druckprotokolls.

Falls mehr als 99 Teilnehmer erfaßt werden sollen, sind die DIM-Statements in den Zeilen 110-120-130 entsprechend der Teilnehmerzahl + 1 zu erhöhen. Das Programm läuft auf dem TI 99/4 A mit Extended-Basic-Modul + Drucker.

Herbert May

```

1 CALL SCREEN(16)
2 DIM LP(100)
100 DIM CY$(100)
110 DIM A(100)
120 FOR I=1 TO 100
121 PRINT
122 INPUT "LISTENPLATZ      ":LP(I)
)
123 IF LP(I)=0 THEN 220
124 INPUT "NAME              ":CY$(
I)
125 INPUT "ERGEBNIS        ":A(I)
126 N=N+1
127 LP(I)=LP(I)/100
128 A(I)=A(I)+LP(I)
129 NEXT I
220 GOSUB 320
221 CALL CLEAR
230 FOR I=1 TO 100
240 AA=ABS(A(I)-INT(A(I)))
250 A(I)=INT(A(I))
260 AA=AA*100
270 B=B+1
275 PRINT : :
280 PRINT B;" (.":AA;")";TAB(12)
;CY$(AA);":":TAB(25);A(I)
285 OPEN #1:"TP.E",OUTPUT :: PRI
NT #1:B;" (.":AA;")";TAB(12);CY$(
AA);":":TAB(25);A(I):: CLOSE #1
290 Z=Z+1
300 IF Z=5 THEN 590
301 IF AA=0 THEN 311
310 NEXT I
311 END
320 HL=N

```

```

330 FOR HK=INT(N/2) TO 1 STEP -1
340 HI=HK
350 GOSUB 460
360 NEXT HK
370 FOR HK=N TO 2 STEP -1
380 HH=A(1)
390 A(1)=A(HK)
400 A(HK)=HH
410 HI=1
420 HL=HK-1
430 GOSUB 460
440 NEXT HK
450 RETURN
460 HJ=2*HI
470 IF HJ>HL THEN 570
480 IF HJ=HL THEN 510
490 IF A(HJ)<A(HJ+1) THEN 510
500 HJ=HJ+1
510 IF A(HJ)>A(HI) THEN 570
520 HH=A(HI)
530 A(HI)=A(HJ)
540 A(HJ)=HH
550 HI=HJ
560 GOTO 460
570 RETURN
580 Z=Z+1
590 Z=0
600 GOSUB 620
610 GOTO 310
620 CALL KEY(0,J,ST)
630 IF ST=0 THEN 620
640 IF J=74 THEN 650
650 RETURN
660 Z=0

```

Starship VC-20



HC-Invaders

Zum ersten populären Computer-Videospiel wurden weltweit die Space-Invaders, kleine Monster aus dem All, die versuchen, die Erde zu erobern. Sie haben es geschafft; keine Frage!

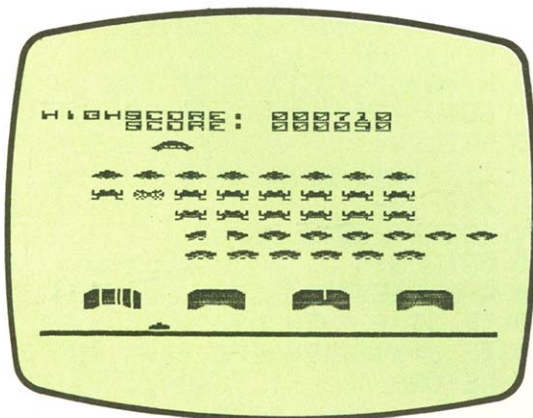
Homecomputer bringt Ihnen für Ihren VC-20, damit Sie Gelegenheit erhalten, den Editor/Assembler zu testen, eine eigene Version der bekannten Invaders ins Haus.

Instruktionen:

1. Editor eingeladen
2. INVADERS (Quellprogramm) eingeben
3. Assembler einladen
4. POKE 55,254: POKE 56,55: RUN
5. POKE 43,254: POKE 44,55: POKE 45,64: POKE 46,63: POKE 56,64: POKE 55,0
6. SAVE „INVADERS“ (MASCHINENPROGRAMM)
7. POKE 43,1: POKE 44,28: POKE 168,0:
8. LOAD „ZEICHENDATAS“
9. RUN
10. LOAD „INVADERS“ (BASIC-PGM)
11. RUN

Wenn alles auf Kassette gespeichert ist, geht man vor wie folgt: POKE 43,254: POKE 44,55: „LOAD“ (Maschinenprogramm).

LOAD „INVADERS“ (Maschinenprogramm) fahre fort wie oben ab 7.



ZEICHENDATAS

```

10 FORI=5128076511:REARDE:POKEI,E:NEXT
1000 DATA124,64,124,68,124,12,12,12,124,60,36,126,98,98,98,98,0
1010 DATA124,36,62,50,50,50,126,0,126,66,64,95,95,98,126,0
1020 DATA126,34,34,50,50,50,126,0,126,64,64,120,96,95,126,0
1030 DATA126,64,64,120,96,96,96,0,126,66,64,110,98,98,126,0
1040 DATA66,66,66,126,98,98,98,0,16,16,16,24,24,24,24,0
1050 DATA2,2,2,6,70,60,0,0,66,68,72,126,98,98,98,0
1060 DATA64,64,64,96,96,126,0,102,90,66,98,98,98,98,0
1070 DATA14,74,74,106,106,102,0,126,66,66,98,98,98,126,0
1080 DATA126,66,66,126,96,96,96,0,126,66,66,98,98,98,126,0
1090 DATA126,66,66,126,100,98,98,0,126,66,64,126,6,70,126,0
1100 DATA124,16,16,24,24,24,24,0,66,66,66,98,98,98,126,0
1110 DATA66,66,66,36,24,24,0,98,98,98,98,66,90,102,0
1120 DATA66,36,24,124,98,98,98,0,66,66,66,60,24,24,24,0
1130 DATA126,2,28,96,36,126,0,60,32,32,48,48,48,60,0
1140 DATA60,126,255,255,255,255,126,60,60,4,4,12,12,12,60,0
1150 DATA0,8,28,42,8,0,0,16,32,127,32,16,0
1160 DATA0,0,0,0,0,0,24,24,28,28,28,0,28,0
1170 DATA108,108,36,36,0,0,0,36,36,126,52,126,52,0
1180 DATA8,62,40,62,26,62,24,0,112,82,116,8,38,42,78,0
1190 DATA12,16,32,48,48,16,12,0,48,0,4,12,12,8,48,0
1200 DATA12,16,32,48,48,16,12,0,48,0,4,12,12,8,48,0
1210 DATA8,42,28,62,42,8,0,0,8,62,62,12,12,0
1220 DATA0,0,0,0,56,56,8,0,0,126,126,0,0,0
1230 DATA0,0,0,0,24,24,0,0,0,2,4,8,16,32,64,0
1240 DATA126,66,70,106,114,98,126,0,56,8,62,24,24,24,60,0
1250 DATA126,66,72,126,96,96,126,0,62,2,2,30,6,6,126,0
1260 DATA64,64,68,68,126,12,12,0,62,32,62,6,70,126,0
1270 DATA126,64,64,126,98,98,126,0,126,66,4,8,24,24,24,0
1280 DATA60,36,36,126,98,98,126,0,126,66,66,126,6,6,126,0
1290 DATA0,0,24,0,24,0,0,0,0,56,0,56,0,0,0
1300 DATA14,24,48,36,48,24,14,0,0,128,128,252,252,4,7,0
1310 DATA1,1,1,7,15,15,15,0,0,128,128,252,252,4,7,0
1320 DATA15,63,255,230,121,16,162,0,0,192,240,112,224,128,96,0
1330 DATA0,3,15,14,7,3,1,0,240,252,255,103,158,112,152,0
1340 DATA0,0,24,24,126,255,255,255,0,0,0,0,0,0,0
1350 DATA0,0,6,31,53,63,0,0,0,128,192,192,192,0
1400 DATA0,0,1,7,15,15,15,0,0,128,128,252,252,4,7,0
1410 DATA0,0,0,1,3,3,0,0,96,96,248,252,252,252,252,14,4
1420 DATA3,15,31,63,106,235,112,32,192,240,248,252,86,255,14,4
1430 DATA0,0,0,0,0,0,0,0,1,7,15,31,127,56,16
1440 DATA24,248,252,254,43,255,7,2,0,0,0,0,128,0,0
1450 DATA0,3,7,15,26,63,28,8,240,252,254,255,149,255,3,1
1460 DATA0,0,0,0,128,128,0,0,1,3,7,13,31,14,4
1470 DATA120,254,255,255,74,255,1,0,0,0,128,192,224,192,128
1480 DATA0,0,1,3,6,15,7,2,60,255,255,255,165,255,0,0
1490 DATA0,0,128,192,96,240,224,64,0,0,0,1,3,7,3,1
1500 DATA0,127,255,255,92,255,128,0,0,128,192,224,176,248,112,32
1510 DATA0,0,0,1,3,1,0,15,63,127,255,169,255,192,128
1520 DATA0,192,224,240,96,252,56,16,0,0,0,0,1,0,0
1530 DATA7,31,63,127,255,224,64,128,224,240,248,172,254,28,8
1540 DATA0,41,150,64,150,41,80,0,160,64,144,32,144,64,160,0
1550 DATA2,9,4,9,2,5,0,10,148,105,2,105,148,10,0
1560 DATA124,254,127,181,33,0,0,0,0,0,0,0,0,0,0
1570 DATA31,63,31,45,0,0,0,0,128,192,64,64,0,0,0
1580 DATA7,15,7,1,2,0,0,0,192,224,240,80,16,0,0,0
1590 DATA1,3,1,2,0,0,0,0,240,248,252,212,132,0,0,0
1600 DATA164,85,171,106,60,126,126,60,0,0,0,0,0,0,0
1610 DATA10,5,10,5,3,7,3,64,80,176,160,192,224,224,192
1620 DATA144,34,85,40,44,58,127,255,0,0,0,0,0,0,0
1630 DATA35,8,21,10,1,14,31,63,0,128,64,0,0,0,192,192
1640 DATA9,2,5,2,3,7,15,0,32,80,128,192,160,240,240
1650 DATA2,0,1,0,0,0,1,3,64,136,84,160,176,232,252,252
1660 DATA9,68,170,20,52,126,254,255,0,0,0,0,0,0,0
1670 DATA2,17,42,5,13,23,63,63,64,0,128,0,0,0,128,192
1680 DATA0,4,10,1,3,5,15,15,144,64,160,64,64,112,224,240
1690 DATA0,1,2,0,0,1,3,3,65,16,168,80,208,112,248,252
1700 DATA15,31,63,127,127,127,127,255,255,255,255,255,255,255,255
1710 DATA240,248,252,254,254,254,254,15,31,63,127,127,127,127
1720 DATA255,255,255,255,255,255,240,248,252,254,254,254,254

```

READY.

INVADERS (BASIC-PGM)

```

30 POKE55,254:POKE56,55
100 POKE5120,0:POKE5121,0:FORI=5122T05127:POKEI,255:NEXT:REM BODEN IM CHR-RAM
110 POKE15841,0:REM NEUER BODEN
120 POKE15752,2:REM HIMMELBLINKGESCHWINDIGKEIT
130 POKE15805,3:REM BODENBLINKGESCHWINDIGKEIT
140 POKE14812,15:REM BODENBLINKANZAHL
150 POKE14916,4:REM HIMMELBLINKANZAHL
160 POKE16190,201:POKE16191,64:POKE16192,144:POKE16193,5:POKE16194,201:POKE16195,76
170 POKE16196,176:POKE16197,1:POKE16198,96:POKE16199,104:POKE16200,24:POKE16201,180
180 POKE16202,4:POKE16203,72:POKE16204,96
190 POKE15429,234:POKE15430,32:POKE15431,62:POKE15432,63
200 POKE15513,234:POKE15514,32:POKE15515,62:POKE15516,63
210 POKE15351,2:REM INVADERSGESCHWINDIGKEIT
220 POKE16018,66:REM BLAUE FARBE
230 POKE15031,5:REM INVADERBLINKANZAHL
240 GOTO270 SCHUSSKORREKTUR
250 S=0:FORI=4136T01-5STEP-1:S=10*S+(15*ANDPEEK(I)):NEXT
260 IFS>HITENHI=S
270 FORI=37888+66T01+21:POKEI,5:NEXT
280 FORI=37954T01+21:POKEI,53:NEXT
290 POKE36869,205
300 PRINT"HIHGHSCORE: "RIGHT$(STR$(165+HI),6)
310 PRINT" SCORE: 000000"
320 PRINT"#####"
330 POKE912,1:REM ANZAHL BÄSEN
340 FORI=37953T01+19:POKEI,4:NEXT
350 S1=15339:REM INVADER-MOVE
360 S2=15239:REM UFO
370 S3=15103:REM TASTATURABFRAGE
380 S4=14853:REM SPIELERSCHUSS WEITERBEBEIDEN
390 S5=15740:REM HIMMELBLINKEN
400 S6=15794:REM BODENBLINKEN
410 S7=15848:REM INVADERBLINKEN
420 S8=14826:REM INVADERSCHUSS INITIALISIEREN UND WEITERBEBEIDEN
430 S9=15659:REM BASISBLINKEN
440 FORI=864T01+44:POKEI,0:NEXT
450 FORI=864T01+44:POKEI,256*AND(I):NEXT
460 POKE862,88:POKE863,16:POKE839,88:POKE840,16:POKE36869,205
470 POKE851,16:POKE852,4:POKE853,1:POKE854,64
480 SY315936:POKE3,0
490 FORI=37888+21*22T01+21:POKEI,2:NEXT
500 IFFPEEK(3)THEN250
510 IFFPEEK(661)>ATTHENROSUBS50
520 AI=PEEK(861)
530 SY31:SY355:SY354:SY355:SY356:SY357:SY358:SY359
540 GOTO500
550 RESTORE:FORI=6240T01+23:REARDE:POKEI,R:POKEI+24,R:POKEI+48,R:POKEI+72,R:NEXT
560 FORI=6416T01+23:REARDE:POKEI,R:POKEI+24,R:POKEI+48,R:POKEI+72,R:NEXT
570 PRINT"#####"
580 PRINT"CHR$(34);:POKE212,0:PRINT"#####"
590 RETURN
600 DATA15,31,63,127,127,127,127,255,255,255,255,255,255,255,255
610 DATA240,248,252,254,254,254,254,15,31,63,127,127,127,127,127
620 DATA127,127,127,127,127,127,126,126,124,120,255,255,255,0,0,0,0
630 DATA254,254,254,254,254,254,254,254,126,62,30

```

Starship VC-20

INVADERS (Quellprogramm)

1	ORG \$37FE	62	CMP #17
2PTR1	EQU \$33C	63	BNE #+3
3PTR2	EQU PTR1+1	64	STY PTR3
4PTR3	EQU PTR2+1	65	INC PTR4
5PTR4	EQU PTR3+1	66	LDA PTR4
6UFF	EQU \$340	67	CMP #45
7UP	EQU \$341	68	BNE #+3
8BOBZ	EQU \$342	69	STY PTR4
9BBZ	EQU \$343	70	PLA
10ISF	EQU \$344	71	RTS
11ZUISCH	EQU \$345	72CHRRADR	LDY #0
12ZUSSCH	EQU \$346	73	STY \$FF
13INVI	EQU \$347	74	ASL
14ERG	EQU \$351	75	ROL \$FF
15BP	EQU \$352	76	ASL
16TAB	EQU \$353	77	ROL \$FF
17SSF	EQU \$357	78	ASL
18ISCHADR	EQU 0	79	ROL \$FF
19SSCHADR	EQU 193	80	STA \$FE
20HIM	EQU \$358	81	CLC
21HIMZ	EQU \$359	82	LDA \$FF
22SBDZ	EQU \$35A	83	ADC #20
23BI	EQU \$35B	84	STA \$FF
24IBZ	EQU \$35C	85	RTS
25AZI	EQU \$35D	86SC0	STA 87
26INVA	EQU \$35E	87	SED
27AZB	EQU \$390	88	CLC
28LIZ	EQU \$391	89	LDX #4
29MR	EQU \$34C	90	LDY #0
30BDZ	EQU \$34D	91SC1	ADC 87
31MDZ	EQU \$34E	92	BCC #+2
32SC	EQU 4128	93	INY
33ULA	EQU 4096+66	94	CLC
34BLA	EQU 4558	95	DEX
35F1	EQU 5	96	BNE SC1
36F2	EQU 2	97	STA 87
37F3	EQU 3	98	CLD
38	BRK	99	AND #15
39	BRK	100	ADC SC+6
40RND	LDX PTR1	101	STA SC+6
41	LDA \$360,X	102	CMP #3A
42	LDX PTR2	103	BCC #+5
43	ADC \$360,X	104	SBC #10
44	LDX PTR3	105	STA SC+6
45	ADC \$370,X	106	PHP
46	LDX PTR4	107	LDA 87
47	STA \$360,X	108	LSR
48	PHA	109	LSR
49	LDY #0	110	LSR
50	INC PTR1	111	LSR
51	LDA PTR1	112	PLP
52	CMP #13	113	ADC SC+5
53	BNE #+3	114	STA SC+5
54	STY PTR1	115	CMP #3A
55	INC PTR2	116	BCC #+5
56	LDA PTR2	117	SBC #10
57	CMP #15	118	STA SC+5
58	BNE #+3	119	TYA
59	STY PTR2	120	LDX #4
60	INC PTR3	121SC2	ADC #0
61	LDA PTR3	122	ADC SC,X

Starship VC-20

123	STA SC,X	184Z57	STA 6528,X
124	CMP ##3A	185	DEX
125	BCC #+5	186	BPL Z57
126	SBC #10	187	STX ISF
127	STA SC,X	188	TYA
128	LDA #0	189	CLC
129	DEX	190	ADC #22
130	BNE SC2	191	ADC ISCHADR
131	RTS	192	STA ISCHADR
132INWADER	LDA BOBZ	193	BCC #+2
133	ORA BBZ	194	INC ISCHADR+1
134	BEQ #+1	195	LDA #32
135RTS	RTS	196	STA ZUISCH
136	LDA ISF	197	LDY #0
137	BNE ISCH	198	LDA #176
138	JSR RND	199	STA (ISCHADR),Y
139	CMP #8	200	RTS
140	BCS RTS	201ISCH	LDY #0
141	LDX INVA+1	202	LDA ZUISCH
142	LDA INVA	203	STA (ISCHADR),Y
143	CLC	204	CLC
144	ADC #176	205	LDA ISCHADR
145	STA \$FE	206	ADC #22
146	BCC #+1	207	STA ISCHADR
147	INX	208	BCC #+2
148	STX \$FF	209	INC ISCHADR+1
149	JSR RND	210	LDA (ISCHADR),Y
150	AND ##1F	211	STA ZUISCH
151	CMP #22	212	CMP #32
152	BCS #-9	213	BNE ISS1
153	TYA	214	LDA #176
154IS1	LDA \$FE	215	STA (ISCHADR),Y
155	STA ISCHADR	216	RTS
156	LDA \$FF	217ISS1	CMP #76
157	STA ISCHADR+1	218	BCC P3
158	LDX #5	219	CMP #140
159IS2	LDA (0),Y	220	BCS #+4
160	CMP #64	221	CMP #84
161	BCS IS3	222	BCS P3
162	SEC	223	CMP #174
163	LDA ISCHADR	224	BCS P3
164	SBC #44	225	JSR CHRRADR
165	STA ISCHADR	226	LDY #7
166	BCS #+2	227X20	LDA (\$FE),Y
167	DEC ISCHADR+1	228	STA 6536,Y
168	DEX	229	AND 6528
169	BNE IS2	230	BNE X21
170	BEQ IS4	231	DEY
171IS3	AND #3	232	BPL X20
172	CLC	233	LDY #7
173	BEQ IS5	234X22	LDA (\$FE),Y
174	CMP #3	235	ORA 6528
175	BEQ IS5	236	STA 6536,Y
176IS4	DEY	237	DEY
177	BPL IS1	238	BPL X22
178	LDY #21	239	INY
179	BNE IS1	240	LDA #177
180IS5	BCC #+3	241	STA (ISCHADR),Y
181	LDA ##40	242	RTS
182	DFB \$2C,\$A9,4	243X21	LDY #0 ;GETROFF
183	LDX #7	244	STY ISF

Starship VC-20

245	LDA ZUISCH	306	SBC #22
246	CMP #84	307	STA SSCHADR
247	BCS X23	308	BCS #+2
248	ADC #48	309	DEC SSCHADR+1
249	TAY	310	CMP #66
250	LDA BP	311	LDA SSCHADR+1
251	LSR	312	SBC ##10
252	LSR	313	BCS SF2
253	TAX	314	LDX HIM
254	TYA	315	CPX #3
255	STA BLA,X	316	BCC #+8
256	CLC	317	LDA SSCHADR
257	ADC #1	318	BNE #+2
258	STA BLA+1,X	319	DEC SSCHADR+1
259	LDA #10	320	DEC SSCHADR
260	STA BBZ	321	TXA
261	STA BDZ	322	ASL
262	RTS	323	CLC
263X23	LDA 6528	324	STY SSF
264	EOR ##FF	325	ADC #112
265	LDY #7	326	STA (SSCHADR),Y
266	PHA	327	INY
267X51	PLA	328	ADC #1
268	PHA	329	STA (SSCHADR),Y
269	AND (#FE),Y	330	LDX #20
270	STA (#FE),Y	331	STX HIMZ
271	DEY	332	STX SBDZ
272	BPL X51	333	RTS
273	PLA	334SF2	LDY #0
274	INY	335	LDA (SSCHADR),Y
275	STY ISF	336	STA ZUSSCH
276	RTS	337	CMP #32
277P3	LDX #20 ;BODEN	338	BNE #+5
278	STX BOBZ	339	LDA #178
279	STX BDZ	340	STA (SSCHADR),Y
280	LDY #0	341	RTS
281	STY ISF	342	JSR CHRRADR
282	LDA 6528	343	LDY #7
283	CMP #64	344SF3	LDA (#FE),Y
284	BCC BOD1	345	STA 6552,Y
285	LDX ISCHADR	346	AND 6544
286	BNE #+2	347	BNE SFG
287	DEC ISCHADR+1	348	DEY
288	DEC ISCHADR	349	BPL SF3
289BOD1	LDA #0	350	LDY #7
290	ROL	351SF4	LDA (#FE),Y
291	ASL	352	ORA 6544
292	ADC #120	353	STA 6552,Y
293	STA (ISCHADR),Y	354	DEY
294	INY	355	BPL SF4
295	ADC #1	356	INY
296	STA (ISCHADR),Y	357	LDA #179
297	RTS	358	STA (SSCHADR),Y
298SSCHINC	LDA SSF	359	RTS
299	BNE #+1	360SFG	LDA ZUSSCH ;GETROFFEN
300	RTS	361	LDY #0
301	LDY #0	362	STY SSF
302	LDA ZUSSCH	363	CMP #140
303	STA (SSCHADR),Y	364	BCC SF5
304	SEC	365	CMP #174
305	LDA SSCHADR	366	BCS SF5

Starship VC-20

367	LDA 6544	428	TAX
368	EOR #FF	429	LDA #32
369	LDY #7	430	STA BLA,X
370	PHA	431	STA BLA+1,X
371	PLA	432	LDA 197
372	PHA	433	CMP #30
373	AND (\$FE),Y	434	BNE *+10
374	STA (\$FE),Y	435	LDY BP
375	DEY	436	CPY #83
376	BPL #-9	437	BEQ *+3
377	PLA	438	INC BP
378	RTS	439	CMP #33
379SF5	CMP #76	440	BNE *+8
380	BCS SF6	441	LDY BP
381	STA BI	442	BEQ *+3
382	LSR	443	DEC BP
383	BCC SF7	444	LDA BP
384	LDA SSCHADR	445	LSR
385	BNE *+2	446	LSR
386	DEC SSCHADR+1	447	TAX
387	DEC SSCHADR	448	LDA BP
388SF7	LDX #10	449	AND #3
389	STX IBZ	450	ASL
390	INC SBDZ	451	ADC #76
391	RTS	452	STA BLA,X
392SF6	LDX #21	453	ADC #1
393	LDA #32	454	STA BLA+1,X
394	STA ULA,X	455T1	RTS
395	DEX	456SS	LDA SSF
396	BNE *-6	457	ORA IBZ
397	STX SSF	458	ORA HIMZ
398	STX UFF	459	BNE T1
399	JSR RND	460	LDA BP
400	CMP #254	461	AND #3
401	BCC *+4	462	TAX
402	LDA ##60	463	STX HIM
403	BNE JSC	464	LDA TAB,X
404	CMP #252	465	LDY #7
405	BCC *+4	466Z34T	STA 6544,Y
406	LDA ##50	467	DEY
407	BNE JSC	468	BPL Z34T
408	CMP #248	469	LDA BP
409	BCC *+4	470	LSR
410	LDA ##40	471	LSR
411	BNE JSC	472	CPX #3
412	CMP #240	473	SBC #21
413	BCC *+4	474	ADC #>BLA
414	LDA ##30	475	STA SSCHADR
415	BNE JSC	476	LDA #<BLA
416	CMP #224	477	SBC #0
417	BCC *+4	478	STA SSCHADR+1
418	LDA ##20	479	STA SSF
419	BNE JSC	480	LDA #32
420	LDA ##15	481	STA ZUSSCH
421JSC	JMP SCO	482	LDY #0
422TAST	LDA 197	483	LDA #178
423	CMP #32	484	STA (SSCHADR),Y
424	BEQ SS	485Z449	RTS
425	LDA BP	486UFO	LDA UFF
426	LSR	487	BNE INCU
427	LSR	488	JSR RND

Starship VC-20

489	CMP #254	550	DEX
490	BCC Z449	551	BMI IM2
491	STA UFF	552	ADC #44
492	LDA #0	553	BCC IM1
493	STA UP	554	INC #FF
494	LDA #84	555	BCC IM1
495	STA ULA	556IM2	STA #FE
496	LDA #85	557	LDY #21
497	STA ULA+1	558	LDA #0
498	LDA #86	559	ORA (#FE),Y
499	STA ULA+2	560	DEY
500	RTS	561	BPL *-5
501INCU	LDA UP	562	AND #2
502	LSR	563	CMP #2
503	LSR	564	LDX MR
504	LSR	565	BNE IM3
505	TAX	566	LDA #FE
506	LDA #32	567	ADC #0
507	STA ULA,X	568	STA 88
508	STA ULA+1,X	569	LDA #FF
509	STA ULA+2,X	570	ADC #0
510	INC UP	571	STA 89
511	LDA UP	572IM4	LDY #21
512	EOR #160	573	LDA (#FE),Y
513	BNE #+4	574	PHA
514	STA UFF	575	LDA #32
515	RTS	576	STA (#FE),Y
516	EOR #160	577	PLA
517	LSR	578	CMP #64
518	LSR	579	BCC IM41
519	LSR	580	EOR #2
520	TAX	581	STA (88),Y
521	LDA UP	582IM41	DEY
522	AND #7	583	BPL IM4+2
523	STA #FE	584	PLP
524	ASL	585	BEQ #+1
525	ADC #FE	586	RTS
526	ADC #84	587	LDX INVA+1
527	STA ULA,X	588	LDA INVA
528	ADC #1	589	STX #FF
529	STA ULA+1,X	590	STA #FE
530	ADC #1	591	LDA #0
531	STA ULA+2,X	592	LDY #21
532	RTS	593	LDX #5
533INVM	DEC MDZ	594IM5	ORA (#FE),Y
534	BMI #+1	595	PHA
535	RTS	596	LDA #FE
536	LDA IBZ	597	CLC
537	BNE *-6	598	ADC #44
538	LDX #4	599	STA #FE
539	STX MDZ	600	BCC #+2
540	LDX LIZ	601	INC #FF
541	DEX	602	PLA
542	BPL #+2	603	DEX
543	LDX #4	604	BNE IM5
544	STX LIZ	605	AND #3
545	PHP	606	CMP #3
546	LDA INVA+1	607	BEQ #+1
547	STA #FF	608	RTS
548	LDA INVA	609	DEC MR
549IM1	CLC	610	JMP MD

Starship VC-20

611IM3	LDY #0	672	LDA (88),Y
612	LDA #FE	673	STA (90),Y
613	SBC #0	674	TYA
614	STA 88	675	BNE MD1
615	LDA #FF	676	LDA #32
616	SBC #0	677	LDY #21
617	STA 89	678	STA (88),Y
618IM6	LDA (%FE),Y	679	DEY
619	PHA	680	BPL *-5
620	LDA #32	681	LDA 89
621	STA (%FE),Y	682	CLC
622	PLA	683	ADC #132
623	CMP #64	684	STA 89
624	BCC IM61	685	LDA 91
625	EOR #2	686	ADC #132
626	STA (88),Y	687	STA 91
627IM61	INY	688	LDY #198
628	OPY #22	689MD2	DEY
629	BNE IM6	690	LDA (88),Y
630	PLP	691	STA (90),Y
631	BEQ #+1	692	TYA
632	RTS	693	BNE MD2
633	LDY #0	694	LDX #22
634	STY 87	695MD3	LDA 4513,X
635	LDX INVA+1	696	CMP #64
636	LDA INVA	697	BCC MD4
637	STX #FF	698	CMP #76
638	STA #FE	699	BCC MD5
639	LDX #5	700MD4	DEX
640IM7	LDA (%FE),Y	701	BNE MD3
641	CMP #64	702	RTS
642	BCC IM8	703MD5	INC 3
643	EOR #2	704	RTS
644	ORA 87	705BBL	LDA BBZ
645	STA 87	706	BEQ BB1
646IM8	LDA #FE	707	DEC BDZ
647	CLC	708	BEQ #+1
648	ADC #44	709BB1	RTS
649	STA #FE	710	LDA #8
650	BCC #+2	711	STA BDZ
651	INC #FF	712	DEC BBZ
652	DEX	713	BNE BB2
653	BNE IM7	714	LDX #22
654	LDA 87	715	LDA #32
655	AND #2	716BB3	STA BLA-1,X
656	BNE #+1	717	DEX
657	RTS	718	BNE BB3
658	INC NR	719	STX ISF
659MD	LDA INVA	720	DEC A2B
660	STA 88	721	BNE #+2
661	CLC	722	INC 3
662	ADC #22	723	RTS
663	STA 90	724BB2	LDA BF
664	STA INVA	725	LSR
665	LDA INVA+1	726	LSR
666	STA 89	727	TAX
667	ADC #0	728	LDA BLA,X
668	STA INVA+1	729	CMP #132
669	STA 91	730	BCC #+4
670	LDY #198	731	SBC #8
671MD1	DEY	732	BOS #+2

Starship VC-20

733	ADC #8	794	RTS
734	STA BLA,X	795IBL	LDA IBZ
735	CLC	796	BEQ #+5
736	ADC #1	797	DEC SBDZ
737	STA BLA+1,X	798	BEQ #+1
738	RTS	799	RTS
739HIBL	LDA HIMZ	800	LDA #5
740	BEQ HI1	801	STA SBDZ
741	DEC SBDZ	802	DEC IBZ
742	BEQ #+1	803	BEQ IB1
743HI1	RTS	804	LDY #0
744	LDA #8	805	LDA (SSCHADR),Y
745	STA SBDZ	806	CMP #108
746	DEC HIMZ	807	LDA BI
747	BEQ HI2	808	BCS IB2
748	LDA SSCHADR+1	809	AND #2
749	CLC	810	ADC #108
750	ADC #132	811IB3	STA (SSCHADR),Y
751	STA \$FF	812	ADC #1
752	LDA SSCHADR	813	INY
753	STA \$FE	814	STA (SSCHADR),Y
754	LDY #0	815	RTS
755	LDA (\$FE),Y	816IB2	AND #\$7E
756	EOR #22	817	CLC
757	STA (\$FE),Y	818	BNE IB3
758	INY	819IB1	LDY #0
759	STA (\$FE),Y	820	LDA #32
760	RTS	821	STA (SSCHADR),Y
761HI2	LDY #0	822	INY
762	LDA #32	823	STA (SSCHADR),Y
763	STA (SSCHADR),Y	824	LDA BI
764	INY	825	CMP #68
765	STA (SSCHADR),Y	826	BCS #+4
766	RTS	827	LDA ##10
767BOBL	LDA BOBZ	828	BNE JSC1
768	BEQ BO1	829	CMP #72
769	DEC BDZ	830	BCS #+4
770	BEQ #+1	831	LDA ##6
771BO1	RTS	832	BNE JSC1
772	LDA #8	833	LDA #2
773	STA BDZ	834JSC1	JSR SCO
774	DEC BOBZ	835	DEC AZI
775	BEQ BO2	836	BEQ #+1
776	LDA ISCHADR+1	837	RTS
777	CLC	838	LDA #40
778	ADC #132	839	STA AZI
779	STA \$FF	840	LDA INVI
780	LDA ISCHADR	841	CLC
781	STA \$FE	842	ADC #22
782	LDY #0	843	STA INVI
783	LDA (\$FE),Y	844	STA INVRA
784	EOR #22	845	BCC #+6
785	STA (\$FE),Y	846	INC INVI+1
786	INY	847	INC INVRA+1
787	STA (\$FE),Y	848	INC AZB
788	RTS	849	LDA INVI
789BO2	LDY #0	850	STA \$FE
790	LDA #32	851	LDA INVI+1
791	STA (ISCHADR),Y	852	CLC
792	INY	853	ADC #132
793	STA (ISCHADR),Y	854	STA \$FF

Starship VC-20

```
855 LDA #F1
856 LDY #43
857 STA ($FE),Y
858 DEY
859 BPL *-5
860 LDA $FE
861 CLC
862 ADC #44
863 STA $FE
864 BCC *+2
865 INC $FF
866 LDA #F2
867 LDY #87
868 STA ($FE),Y
869 DEY
870 BPL *-5
871 LDA $FE
872 CLC
873 ADC #88
874 STA $FE
875 BCC *+2
876 INC $FF
877 LDY #220
878 LDA #F3
879 DEY
880 STA ($FE),Y
881 TYA
882 BNE *-8
883 LDA #0
884 STA LIZ
885 STA MR
886 LDA INVA
887 STA $FE
888 LDA INVA+1
889 STA $FF
890 LDA #64
891 LDY #14
892 STA ($FE),Y
893 DEY
894 DEY
895 BPL *-6
896 LDA #65
897 LDY #15
898 STA ($FE),Y
899 DEY
900 DEY
901 BPL *-6
902 LDA $FE
903 CLC
904 ADC #44
905 STA $FE
906 BCC *+2
907 INC $FF
908 LDA #68
909 LDY #14
910 STA ($FE),Y
911 DEY
912 DEY
913 BPL *-6
914 LDA #69
915 LDY #15
```

```
916 STA ($FE),Y
917 DEY
918 DEY
919 BPL *-6
920 LDA $FE
921 CLC
922 ADC #44
923 STA $FE
924 BCC *+2
925 INC $FF
926 LDA #68
927 LDY #14
928 STA ($FE),Y
929 DEY
930 DEY
931 BPL *-6
932 LDA #69
933 LDY #15
934 STA ($FE),Y
935 DEY
936 DEY
937 BPL *-6
938 LDA $FE
939 CLC
940 ADC #44
941 STA $FE
942 BCC *+2
943 INC $FF
944 LDA #72
945 LDY #14
946 STA ($FE),Y
947 DEY
948 DEY
949 BPL *-6
950 LDA #73
951 LDY #15
952 STA ($FE),Y
953 DEY
954 DEY
955 BPL *-6
956 LDA $FE
957 CLC
958 ADC #44
959 STA $FE
960 BCC *+2
961 INC $FF
962 LDA #72
963 LDY #14
964 STA ($FE),Y
965 DEY
966 DEY
967 BPL *-6
968 LDA #73
969 LDY #15
970 STA ($FE),Y
971 DEY
972 DEY
973 BPL *-6
974 RTS
```

Commodore & Sinclair

VC-20 · VC-64 · 4032 · 8032

ZX SPECTRUM

HARDWARE

VC-20 16K-RAM-Modul **149.-**
32K-RAM-Modul schaltbar **199.-**
MODULBOX mit 3 Steckplätzen **99.-**
EPROM-Modul inkl. Prommer **149.-**
IEC-BUS-Modul inkl. Kabel **249.-**

SPECTRUM JOYSTIX **69.-**
ermöglicht den Anschluß von
2 Atari- oder Commodore-Joysticks
an Ihren Spectrum. Einfache Abfrage
mit dem IN-Befehl.

SOFTWARE

GRANDMASTER

Das stärkste Schachprogramm der Welt
für Homecomputer!

Mit hochauflösender Grafik, viel
Bedienungskomfort und
unübertroffener Spielstärke.

NEU: ab sofort auch für CBM 4032 & 8032
erhältlich!

VC-20 (+8K), VC-64 **79.-**
4032 (mit 12"-Monitor) und
8032 inkl. HiRes-Eprom **99.-**

SPIELE für VC-20, VC-64 und
Spectrum in Maschinenspr. ab **19.-**

KINGSOFT

Alle Preise inkl. MwSt. 6 Monate Garantie.

Fordern Sie unseren Gesamtkatalog (2 DM) an!

Schnackebusch 4 · D-5106 Roetgen · Telefon 0 24 08/83 19

Editor für 6502-Systeme

Seine Reihe „Programmieren in Maschinensprache“ setzt Homecomputer diesmal fort mit dem Editor.

Sie haben nun eine Programmserie, mit der Sie Programme und Unterprogramme in der Assemblersprache für alle Microcomputer mit dem 6502-Prozessor schreiben können.

Obwohl die Reihe speziell für den VC-20 und VC-64 begonnen wurde, sind die Programme fast ohne Umschreiben auch für die CBMs, den Apple, den Atari, usw. verwendbar. Lediglich die Bildschirm-Befehle und Tastatur-POKES müssen angepaßt werden bzw. können einfach ignoriert werden.

Der Assembler, der in der April-Ausgabe abgedruckt wurde, hat früher als vorgesehen schon weitere Zusatzfunktionen erhalten, denen in den nächsten Monaten von Zeit zu Zeit weitere folgen werden.

Auch der Disassembler aus Heft 3, wo übrigens für den VC-20 zwei Zeilen beim Druck verlorengegangen sind, wurde entsprechend angepaßt. Man beginnt mit dem Einladen des Editors. Nach dem Starten erhält man eine Tabelle mit den Befehlen, die der Editor kennt und den möglichen Parametern.

Es genügt, wenn die inversen Buchstaben eingegeben werden und auch Blanks sind optional. Man kann aber auch den gesamten Befehl eingeben oder Teile davon, sofern die signifikanten Buchstaben am Anfang enthalten sind.

```

*50 PRINT"HOME COMPUTER-ASSEMBLER"
*100 READO$:READA$:O$=O$+A$:READA$:O$=O$+A$:DIMA$(500),A(500)
*110 FORI=0TO7:READAI$(I):NEXT
*120 DIMO$(58):FORI=0TO55:READO$(I):NEXT
*130 FORI=0TO7:READS$(I):NEXT
*140 DIMAD$(21):FORI=0TO21:READAD$(I):NEXT
*200 PRINT"CASSETTE / DISK 800/800":WAIT198,1:GETG$:IFG$<"C"ORG$>"D"THEN200
*210 DI=G$="D":IFNOTDITHENGOSUB50000
*220 INPUT"FILENAME";F$:IFDITHENF$=F$+",S,R"
*500 OPEN1,1+(7ANDDI),2ANDDI,F$:Z=0
*510 Z=Z+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<>" "THENNEXT
*520 IFI=1THENA$(LC)=LEFT$(L$,I-1):A(LC)=PC:LC=LC+1
*525 PRINTRIGHT$( " "+STR$(PC),5)LEFT$(L$,I)TAB(10)MID$(L$,I+1)
*530 GOSUB4000
*540 PC=PC+L
*900 IF(64ANDST)=0THEN510
*910 CLOSE1
*1000 IFNOTDITHENGOSUB50000
*1010 OPEN1,1+(7ANDDI),2ANDDI,F$:F=-1:Z=0
*1020 Z=Z+1:INPUT#1,L$:FORI=1TO9:IFMID$(L$,I,1)<>" "THENNEXT
*1030 GOSUB4000:IFOI>55THEN1150
*1040 O=O$(OI):IFOI>21THEN1070
*1050 IFAI=1ANDOAND1ORL=3THENO=O+S
*1060 O=O+S$(AI)
*1070 IFL>1THENGOSUB5000:IFOI>46ANDOI<55THENS=S-PC-2:S=S+(256ANDS<0)
*1080 S$(S/256):S=S-256*S$:PRINTPCTAB(7)OTAB(12)STAB(17)S$
*1090 POKEPC,O:POKEPC+1,S:IFL=3THENPOKEPC+2,S$
*1100 PC=PC+L:IF(64ANDST)=0THEN1020
*1110 CLOSE1
*1120 GOTO20000
*1150 IFOI=57THEN1100
*1160 K=0:FORP=PCTOP+L-1:A$="":FORK=K+1TOLEN(L$):J#=MID$(L$,K,1)
*1165 IFJ#<>"",THENA$=A#+J$:NEXT
*1170 GOSUB5000:POKEP,S:NEXTP:GOTO1100
*4000 A#=MID$(L$,I+1,3):L=0
*4010 FORJ=1TO175STEP3:IFR#<MID$(O$,J,3)THENNEXT:PRINT"OPCODE ERROR IN"Z:GOTO20000
*4020 OI=(J-1)/3:L#=MID$(L$,I+5)
*4030 IFOI=56ANDNOTFTHENA$=L$:GOSUB5000:A(LC-1)=S
*4032 IFOI=57THENL=1:FORJ=1TOLEN(L$):L=L-(MID$(L$,J,1)=""):NEXT
*4035 IFOI=58THENA$=L$:GOSUB5000:PC=S
*4040 IFOI>55THENRETURN
*4045 IFOI>21THENA$=L$:GOTO4135
*4050 IFLN(L$)=0ORL$="A"THENB$="A":GOTO4140
*4060 A$="":B$="":FORI=1TOLEN(L$):J#=MID$(L$,I,1)
*4070 IFJ#=#"ORJ#=#("THENB$=B#+J$:GOTO4130
*4090 IFJ#=#)"ORJ#=#"THEN4130
*4100 IFJ#=#",THENI=I+1:B$=B#+MID$(L$,I,1):GOTO4130
*4110 IFJ#=#";THEN4135
*4120 A$=A#+J$
*4130 NEXT
*4135 L#=(OI>21)-(OI>46)-(OI>54):IFLTHENRETURN
*4140 FORAI=0TO7:IFB#<>AI$(AI)THENNEXT:GOTO19990
*4150 L=1-(AI>0)-(AI>3):IF(OI=12OROI=15)ANDAI=4THENAI=7
*4152 IF(21ANDAD$(OI))=0THEN19990
*4155 IFAI<6THENRETURN
*4160 GOSUB5000:IFE=10ROI=7THENL=3:RETURN
*4170 L=2-(S>255)
*4175 IFAI=7THENIFL=3AND(OI=10OROI=12)THEN19990
*4180 RETURN

```

Starship VC-20

```
*5000 FORI=1TOLEN(A#):J=ASC(MID*(A#,I,1))
*5010 IFJ=43THENS#MID*(A#,I+1):A#=LEFT*(A#,I-1):GOSUB5040:SA=S:A#=SA#:GOSUB5040
:S=S+SA:RETURN
*5020 IFJ=45THENS#MID*(A#,I+1):A#=LEFT*(A#,I-1):GOSUB5040:SA=S:A#=SA#:GOSUB5040
:S=SA-S:RETURN
*5030 NEXT
*5040 IFASC(A#)=62THENS#MID*(A#,2):GOSUB6000:S=S/256:S=S-256*S%:RETURN
*5050 IFASC(A#)=60THENS#MID*(A#,2):GOSUB6000:S=INT(S/256):RETURN
*6000 E=0:S=VAL(A#):IF#>0DRASC(A#)=48THENRETURN
*6010 J=ASC(A#):IFJ-36THEN6030
*6020 FORJ=2TOLEN(A#):J#=MID*(A#,J,1):S=16*S+VAL(J#)+(ASC(J#)-55ANDJ#>"@"):NEXT:R
ETURN
*6030 IFJ=37THENFORJ=2TOLEN(A#):S=S+S+VAL(MID*(A#,J,1)):NEXT:RETURN
*6035 IFJ=42THENS=PC+L:RETURN
*6040 FORJ=0TOLC-1:IFA#=#(J)THENS=A(J):RETURN
*6050 NEXT:E=1:RETURN
10000 DATADRASLBITANDROLEORLSRJMPCDRORSTYSTASTXLDYLDALDXPCYMPDECOPXSBINC
10010 DATABRKPPLCLPLPSECRTIPHACLIRTSPLASEIDEYTKATYATXSTAYTAKCLVTSXINYDEXCLDINXN
OP
10020 DATASEDBFLBMBVQVSBCCBCSBNEBEQJSREQUDFBORG
10030 DATA#, (X, Y, Z, " ", X
10040 DATA5,6,36,37,38,69,70,68,101,102,132,133,134,164,165,166,196,197,198,228,
229
10050 DATA230,0,8,24,40,56,64,72,88,96,104,120,136,138,152,154,168,170,184,186,2
00
10060 DATA202,216,232,234,248,16,48,80,112,144,170,208,240,32
10070 DATA4,-4,-4,12,12,32,0,16
*10080 DATA222,193,64,222,193,222,193,96,222,193,192,222,194,194,222,194,66,222,2
04,66
*10090 DATA222,204
*19990 PRINT"ADR ERROR IN"Z
*20000 PRINT"MEINLADEN VOM:M
*20005 PRINT"EDITOR ?":POKE198,0:WAIT198,1:GETG#:IFG#="J"THENCLR:LOAD"EDITOR",8
*20010 PRINT"DISASSEMBLER ?":WAIT198,1:GETG#:IFG#="J"THENCLR:LOAD"DISASSEMBLER",
8
*20020 END
50000 PRINT"MCASSETTE ZURUECKSPULEN":PRINT"DANN P DRUECKEN!"
50005 GETWA#:IFWA#<"P"THEN50005
50010 PRINT:RETURN
*60000 INPUTA#:A#="Z"+A#:GOSUB6000:PRINTS:GOTO60000
```

READY.

Die Zeilennummern mit Punkt wurden gegenüber der Heft 4/83-Version geändert. Für Kassettenbetrieb entfallen in den Zeilen 20005 und 20010 die letzten beiden Zeichen.

ADD

Um ein Assembler-Programm einzugeben, benutzt man den Befehl A(DD) (Anm.: die Buchstaben in Klammern sind optional).

Dann erscheint eine "1" mit einem inversen Blank dahinter. Dies ist der Cursor, während die "1" die laufende Zeilennummer des Befehls ist. Wenn man einmal einen Befehl unter einer bestimmten Zeilennummer eingegeben hat, so heißt dies nicht, daß zu dieser Zeilennummer immer der eingegebene Befehl gehört, denn man kann auch noch nachträglich Befehle einfügen, und dabei ändern sich dann entsprechend die Nummern der folgenden Befehle.

ADD heißt immer Anhängen von weiteren Befehlen. Wenn der Speicher leer ist, dann lautet die „nächste“ Zeile 1, wenn bereits 43 Zeilen vorhanden sind, dann ist 44 die Zeile, in die nach ADD geschrieben wird.

INSERT

Hat man irgendwo im Assembler-Listing einen oder mehrere Befehle vergessen und nachzutragen, so benutzt man den Befehl I(NSERT). Und zwar gibt man nach dem Befehlswort die Zeilennummer an, hinter der die Befehle eingeschoben werden sollen. Fehlen Befehle direkt am Anfang, so wird nur der Befehl ohne Parameter eingegeben und die Zeilen werden solange eingeschoben,

bis eine neue Editoranweisung dieses beendet.

DELETE

Sind Assembler-Zeilen überflüssig geworden, so kann man sie mit D(DELETE) entfernen. Wie bei dem BASIC-Befehl LIST kann man hinter dem Delete-Befehl entweder nur eine Zeilennummer eingeben, wenn man nur eine Zeile löschen will; oder z. B. "D-5" eingeben, wenn man die ersten 5 Zeilen löschen möchte; oder "D5-", wenn alle Zeilen ab Nummer 5 gelöscht werden sollen; oder einfach "D", um sämtliche Zeilen zu löschen.

Damit man nicht aus Versehen Zeilen entfernt, die man noch benötigt, wird zur Sicherheit noch einmal

gefragt, ob man auch sicher ist. Die Frage (in Englisch) taucht immer dann auf, wenn mehr als 10 Zeilen gelöscht werden sollen.

LIST

Auch bei L(IST) ist die Parameterübergabe wie im BASIC. Wird beim Listen die Taste "S" gedrückt, so stoppt der Editor und erwartet zur Fortsetzung einen weiteren Tastendruck. Drücken der Taste "Q" beendet das Listen bevor die letzte Zeilennummer erreicht ist.

EDIT

Stellt man fest, daß eine Zeile falsch eingegeben wurde, so kann sie mit dem E(DIT)-Befehl korrigiert werden. Die Parameterübergabe ist wie unter DELETE beschrieben. Der Editor listet die zu korrigierende Zeile und setzt den Cursor an den Zeilenanfang.




Diesmal ist es ein „echter Cursor“ und es zählt, wie vom CBM gewohnt, alles, was in der oder den entsprechenden Bildschirmzeilen steht. Deshalb kann man nach Durchführen der Korrektur sofort "RETURN" drücken, und auch der Rest nach dem Cursor wird übernommen. Oder nur "RETURN", wenn in der Zeile nichts geändert werden soll.

Auch bei dem EDIT-Befehl kann man einfach eine neue Editor-Anweisung eingeben, wenn man aufhören will, sofern man den Rest der Zeile (oder Zeilen) mit Spaces löscht.

COPY

Wenn man einen Block, den man schon einmal eingegeben hat, an einer anderen Stelle haben möchte, so verwendet man den Befehl C(OPY). Direkt hinter dem Befehl gibt man aber noch keine Parameter an. Der Editor fragt diese näm-

Starship VC-20

```
80 POKE45,32:POKE46,31
90 CLR
100 DIMA$(500)
110 L$=" ":FORI=1TO5:L$=LEFT$(L$+L$,21):NEXT:L$=L$+" "
120 PRINTTAB(6)"HOME COMPUTER
200 PRINT"HOME COMPUTER EDITOR
210 PRINT"APPEND [A]
215 PRINT"SAVE [A][~][E]
220 PRINT"VERIFY [A][~][E]
225 PRINT"ADD
230 PRINT"INSERT [A]
235 PRINT"EDIT [A][~][E]
240 PRINT"DELETE [A][~][E]
245 PRINT"LIST [A][~][E]
250 PRINT"COPY
255 PRINT"QUIT -> ASS + DISASS
280 IFB=1ORB=5ORB=9THENA$="L":GOTO300
290 INPUT"BEFEHL";A$
300 PRINT:RESTORE:FORB=1TO10:READB$:IFLEFT$(A$,LEN(B$))<>
B$THENNEXT
310 IFB<>2ANDB<9THENGOSUB5000
330 ONBGO TO400,600,500,700,800,1000,1200,900,1100,540,200
400 REM APPEND
410 GOSUB6000
420 INPUT#1,A$
430 IFN2<NTHENFORC=NTO N2+1STEP-1:A$(C+1)=A$(C):NEXT
440 N2=N2+1:N=N+1:A$(N2)=A$:IF(64ANDST)=0THEN420
450 CLOSE1:GOTO280
500 REM SAVE
502 INPUT"FILENAME";F$:PRINT"CASSETTE / DISK  :WAIT198,1:GETG$:IFG$="C"THEN510
505 IFG$<>"D"THEN502
507 OPEN1,8,2,F$+ ".S.W":GOTO520
510 OPEN1,1,1,F$
520 FORC=N1TON2:PRINT#1,CHR$(34)A$(C):NEXT:CLOSE1
540 PRINT"ASSEMBLER EINLADEN ?
550 WAIT198,3:GETG$
560 IFG$="J"THENCLR:LOAD"ASSEMBLER",8
570 PRINT"DISASSEMBLER ?
580 WAIT198,3:GETG$
590 IFG$="J"THENCLR:LOAD"DISASSEMBLER",8
595 GOTO200
600 REM ADD
610 C=N:PRINT"
620 GOSUB4000:IFBFTHEN300
630 PRINTL$:GOSUB4500
640 A$(C)=A$:N=C:GOTO620
700 REM LIST
710 PRINT"":FORC=N1TON2:A$=A$(C):GOSUB4500
720 GETG$:IFG$="S"THENWAIT198,1:GETG$
730 IFG$="Q"THEN280
740 NEXT:GOTO280
800 REM DELETE
810 D=N2-N1+1:IFD>10THENPRINT"ARE YOU SURE ? :WAIT198,7:GETG$:IFG$="N"THEN280
820 N=N-D
830 FORC=N1TON:A$(C)=A$(C+D):NEXT:GOTO280
900 REM VERIFY
910 GOSUB6000
920 FORC=N1TON2:INPUT#1,A$
930 IFA$<>A$(C)OR64ANDSTANDC<N2THENPRINT"?VERIFY ERROR IN
"C:CLOSE1:GOTO280
940 NEXT:CLOSE1:PRINT"OK":GOTO280
```

lich der Reihe nach ab. Er beginnt mit dem Bereich der zu kopierenden Zeilen. Die Zeilennummern werden wie bei DELETE, LIST oder EDIT behandelt.

Danach erfolgt nach Anweisung die Eingabe der Zeilennummer, hinter der die Zeilen eingeschoben werden sollen (fast wie bei INSERT).

Man kann den Block nämlich nicht nur anhängen, sondern an jede Stelle innerhalb des Assembler-Listings einfügen, sogar innerhalb des Blocks selbst. Lediglich wenn man den Block direkt an den Anfang kopieren möchte muß man sich eines kleinen Tricks behelfen. Man fügt dann nämlich vorher einfach eine neue Zeile 1 ein, kopiert den Block (Vorsicht! Die zu kopierenden Zeilennummern haben sich jetzt um 1 verschoben!) hinter die Zeile 1 und löscht dieselbe mit D1. Der zu kopierende Block steht jetzt am Anfang.

APPEND SAVE VERIFY

Drei Befehle dienen dem Laden, Retten und Verifizieren von Assembler-Files auf Diskette oder Kassette.

Der Befehl LOAD, den man in diesem Zusammenhang erwartet gibt es nicht. Ihn ersetzt der Befehl AP(PEND), der einen Parameter haben kann; die Zeilennummer, hinter der das File vom Datenträger in das im RAM befindliche Programm geladen werden soll.

Ist der Speicher leer, so wird sich nach APPEND nur das Programm, das zugelesen wurde, im Speicher befinden.

APPEND ohne Parameter hängt das Programm an das im Speicher befindliche an!

Um das Assembler-Listing auf Kassette oder Disk auf-

Starship VC-20

zunehmen, wird der Befehl S(AVE) benutzt. Ohne Parameter speichert man das gesamte, im Speicher befindliche Programm; und mit entsprechenden Parametern (wie oben unter DELETE beschrieben) können Teile des Programms übertragen werden.

Um sich zu vergewissern, daß alles richtig gespeichert wurde, kann mit V(ERIFY) die Korrektheit der Aufnahme überprüft werden. Dabei ist zu beachten, daß wenn Parameter übergeben wurden, diese identisch sind mit denen, die beim Kommando SAVE verwendet wurden.

GOTO 280

Hat man aus irgend einem Grunde Stop oder Break gedrückt, so kann mit GOTO 280 der Programmablauf fortgesetzt werden, ohne daß das Assembler-Listing gelöscht wird (was ja nach RUN der Fall wäre).

QUIT

QU(IT) sorgt als Befehls-eingabe dafür, daß Assembler oder Disassembler ausgeführt werden.

Achtung! Zuerst muß das Assembler-Programm im Speicher auf einen Datenträger (Kassette oder Diskette) aufgenommen werden! Erst dann kann das Programm assembliert werden.

Ist der Assembler nachgeladen, dann werden wie vom Bildschirm her befohlen, die Kassette zurückgespult und die weiteren Befehle befolgt. Dies entfällt bei Diskettenbetrieb, wo lediglich darauf zu achten ist, daß sich Assembler und Assembler-Programm auf der selben Diskette befinden.

```
1000 REM EDIT
1010 FORC=N1TO N2:PRINT"XXXXTTT":S=PEEK(214):A#=A#(C):GOSUB4500
1015 PRINT"J":IFPEEK(214)>STHEN1015
1020 A#="" :SYS50528
1030 FORI=515TO600:IFPEEK(I)THEN A#=A#+CHR$(PEEK(I)):NEXT
1040 GOSUB4100:IFBFTHEN300
1050 A#(C)=A#:NEXTC:GOTO280
1100 REM COPY
1110 INPUT"COPY-BEREICH":A#
1120 GOSUB5000:INPUT"WOHIN (ZEILENNUMMER)":C
1130 IFC>NTHENC=N
1140 D=N2-N1+1
1150 IFC<NTHENFORI=NTOC+1STEP-1:A#(I+D)=A#(I):NEXT
1160 N=N+D:IFN1>CTHENN1=N1+D
1170 FORI=C+1TOC+D:A#(I)=A#(N1):IFN1=CTHENN1=N1+D
1180 N1=N1+1:NEXT
1190 GOTO280
1200 REM INSERT
1210 C=N1
1220 GOSUB4000:IFBFTHEN300
1230 PRINTL$:GOSUB4500
1240 FORI=NTOCSTEP-1:A#(I+1)=A#(I):NEXT:N=N+1:A#(C)=A#
1250 GOTO1220
4000 A#="" :I=1:C=C+I:C$=RIGHT$(" "+STR$(C),3)
4005 PRINT"XXXXTTT"
4010 S=PEEK(214):PRINTC$LEFT$(A#,I-1)" "LEFT$(MID$(A#,I,1)+" ",1)"
" "MID$(A#,I+1)" " J
4015 IFPEEK(214)>STHENPRINT"J":GOTO4015
4020 WAIT198,15:GETG$:J=ASC(G$)
4030 IFJ=29THEN4090
4040 IFJ=157THENI=I+(I>1):GOTO4010
4050 IFJ=148THENA#=LEFT$(A#,I-1)+" "+MID$(A#,I):GOTO4010
4060 IFJ=20THENA#=LEFT$(A#,I-1+(I>1))+MID$(A#,I):I=I+(I>1):GOTO4010
4070 IFJ=13THEN4100
4080 A#=LEFT$(A#,I-1)+G$+MID$(A#,I+1)
4090 I=I+1+(I>LEN(A#)):GOTO4010
4100 FORI=1TOLEN(A#):IFMID$(A#,I,1)<>" "THENNEXT:BF=-1:RETURN
4110 IFMID$(A#,I+1,1)=" "THENA#=LEFT$(A#,I)+MID$(A#,I+2):GOTO4110
4120 BF=0:RETURN
4500 PRINTRIGHT$(" "+STR$(C),3)
4510 FORI=1TO8:I$=MID$(A#,I,1):PRINTI$:IFI$<>" "THENNEXT:I=I-1
4520 PRINTTAB(10)MID$(A#,I+1)" ":RETURN
5000 D=0:FORI=1TO9:J=ASC(MID$(A#,I,1)+" "):IFJ=45THEND=1
5010 IFJ<48ORJ>57THENNEXT
5020 N1=VAL(MID$(A#,I))
5030 N2=VAL(MID$(A#,I+LEN(STR$(N1))))
5040 IFN1=0THENN1=1ANDI<>7:N2=N
5050 IFTHENN2=N1:N1=1
5060 IFRIGHT$(A#,1)="-"THENN2=N
5070 IFN2<N1THENN2=N1
5080 IFN2>NTHENN2=N
5090 IFN1>NTHENN1=N
5100 RETURN
6000 INPUT"FILENAME":F$:PRINT"CASSETTE / DISK 000/000":WAIT198,1:GETG$:IFG$="D"THEN6100
6002 IFG$<>"C"THEN6000
6005 PRINT"CASSETTE ZURUECKSPULEN"
6010 PRINT"BIS ZUM ANFANG DES":PRINT"ASSEMBLER-FILES!"
6020 PRINT"0DANN P DRUECKEN!"
6030 WAIT198,1:GETG$:PRINT"OK":OPEN1,1,0,F$:RETURN
6100 OPEN1,8,2,F$+",S,R":RETURN
10000 DATA P,A,S,L,D,E,I,V,C,0
```

READY.

Für Kassettenbetrieb entfallen in den Zeilen 560 und 590 die letzten beiden Zeichen.

SYNTAX und Features

Gibt man direkt hinter einer Zeilennummer einen Editorbefehl ein, so wird dieser ausgeführt - und nicht in die Zeile übernommen!

Alles andere, was direkt hinter einer Zeilennummer steht und nach einem Blank mit weiteren Eingaben versehen ist, wird in die entsprechende Zeile geschrieben.

Labels werden direkt hinter die Zeilennummer geschrieben und dürfen keine Blanks enthalten. Ein Blank, das einem Label folgt, bedeutet dem Editor, daß nun ein Mnemonic-Befehl folgt. (Mnemonics, soviel sei für Anfänger gesagt, nennt man die Codewörter wie z. B. LDA (Load Akku), STA (Store Akku), TAX (Tranfer Akku to X-Register), usw. - unser 6502-Lehrgang, der in Kürze beginnt und zu dessen

Zweck die Programme vorgestellt wurden, wird darüber Auskunft geben)

Folgt direkt hinter der Zeilennummer ein Blank dann gehört hinter dieses Blank ein Mnemonic.

Dahinter gehört, wenn es sich um einen Mehrbyte-Befehl handelt, wieder ein Blank und dann ein Operand.

Ein Semikolon wird Kommentaren, die zur besseren Verständlichkeit des Programms einer Programmzeile beigefügt werden können, vorangestellt.

Der Assembler versteht Hex- und Dezimalzahlen. Hexzahlen haben ein Dollarzeichen vorangestellt.

Direkte und relative Sprünge zu Labels und zu labelabhängigen Punkten sind möglich. Z. B.:

```
JSR LABEL
oder
JSR LABEL+86
BNE MARKES8-9
```

Auch Sprünge relativ zum Programm-Counter sind gestattet. Das Sternchen "*" ist in diesem Fall das Symbol für den Programm-Counter. Beispiel: BNE. -3 Die bisher zusätzlich vom Assembler verstandenen Mnemonics sind:

```
ORG Programmbeginn
EQU (Equate)
DFB (Define Byte)
ORG hat kein Label und muß eine Hex- oder Dezimalzahl im Operandenfeld haben.
```

Vor EQU muß sich ein Label befinden und dahinter eine entsprechende Hex-, Dual- oder Dezimalzahl.

Vor DFB kann ein Label stehen. Dem Mnemonic folgen ein oder mehrere Hex-Dual- oder Dezimalzahlen. Mehrere Zahlen werden durch Kommata getrennt.

Eine weitere nachträglich eingeführte Bequemlichkeit ist, daß mit den Größer-als-und Kleiner-als-Zeichen Labels in High- und Low-Byte zerlegt werden können.

Z. B.

```
1 ADRESSE EQU 34356
2 LDA #<ADRESSE
3 STA HIGHBYTE
4 LDA #>ADRESSE
5 STA LOWBYTE
```

Wie bereits angesprochen, werden weitere „Bequemlichkeiten“, nützliche Befehle und Hilfsprogramme

zum Programmieren in Assembler (und somit in Maschinencode) folgen.

Änderungen gegenüber Disassembler Programm aus Heft 3/83

```
600 INPUT "■":A$: IFA$="E" THEN CLR: LOAD "EDITOR": S
605 IFA$="L" THEN 1000
610 IFRIGHT$(A$,1)="L" THEN GOSUB 800: GOTO 1000
630 GOTO 600
680 ILEFT$(A$,1)="$" THEN 810
805 PC=VAL(A$): RETURN
810 PC=0: FOR I=2 TO LEN(A$)-1
820 A=ASC(MID$(A$,I,1))-48: IFA>9 THEN A=A-7
830 PC=PC*16+A: NEXT I: RETURN
900 FOR J=3 TO 0 STEP -1: A=INT(DZ/16↑J): DZ=DZ-A*16↑J: A=A+48: IFA>57 THEN A=A+7
910 PRINT CHR$(A): NEXT J: RETURN
950 FOR J=1 TO 0 STEP -1: A=INT(DZ/16↑J): DZ=DZ-A*16↑J: A=A+48: IFA>57 THEN A=A+7
960 PRINT CHR$(A): NEXT J: RETURN
1000 FOR I=1 TO 11: DZ=PC: PRINT "■": GOSUB 900: PRINT "■": P1=PEEK(PC)
1005 ONLY(P1) GOSUB 1100, 1200, 1300
1010 NEXT I: GOTO 600
1100 DZ=P1: GOSUB 950: PRINT "■": TAB(10): M$(P1): PC=PC+1: RETURN
1200 DZ=P1: GOSUB 950: PRINT " ": P2=PEEK(PC+1): DZ=P2: GOSUB 950
1205 PRINT "■": TAB(10): M$(P1):
1210 IFF$(P1)=9 THEN P2=PEEK(PC+1): IFF2<128 THEN P2=PC+P2+2: GOTO 1220
1215 IFF$(P1)=9 THEN P2=PC-(255-P2)+1
1220 PRINT " ": F1$(FX(P1)): DZ=P2: IFF$(P1)=9 THEN GOSUB 900: GOTO 1240
1230 GOSUB 950
1240 PRINT F2$(FX(P1)): PC=PC+2: RETURN
1300 DZ=P1: GOSUB 950: PRINT " ": P2=PEEK(PC+1): DZ=P2: GOSUB 950: PRINT " "
1310 P3=PEEK(PC+2): DZ=P3: GOSUB 950: PRINT "■": TAB(10): M$(P1):
1320 PRINT " ": F1$(FX(P1)): DZ=P3*256+P2: GOSUB 900: PRINT F2$(FX(P1)): PC=PC+3: RETURN

READY.
```

Zeile 602 entfällt

Zeile 1020 entfällt

Für Kassettenbetrieb entfallen in Zeile 600 die letzten zwei Zeichen.

Starship VC-20

Labyrinth in 3,5 K!

Wieviele Leser von Homecomputer uns angerufen haben, haben wir nicht gezählt, aber wenn es um das Labyrinth für den VC-20 ging, dann lag die Antwort auf der Hand, keine Speichererweiterung vorhan-

den, Programm also nicht lauffähig. Wir haben uns deshalb bemüht (vielen Anrufern habe ich es versprochen), falls es überhaupt möglich ist, das Programm für den VC-20 ohne Erweiterung umzuschrei-

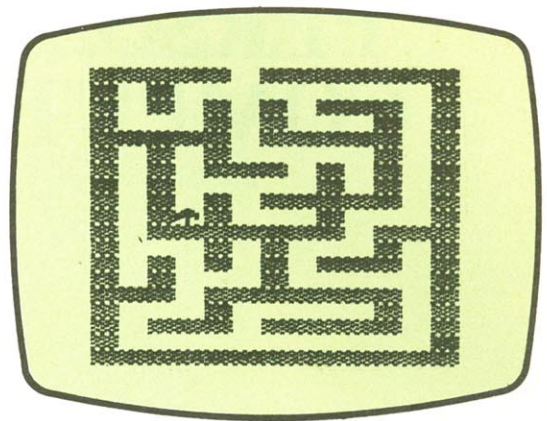
ben, es zum Laufen zu bringen.

Am Ende konnte durch Renummerierung, Dimensionierung, Streichen der Texte und Poken eines neuen RAMTOPs das Programm so weit gekürzt wer-

den, daß es inclusive Variablen fast genau soviel Platz verbraucht, wie ohne Speichererweiterung zur Verfügung stehen.

```
1 DIMM(7),W(3)
2 POKE52,29:POKE54,29:POKE56,29
3 VP=30720:CO=5
4 GOTO79
5 Y=PEEK(TC-U):Z=PEEK(TC-U+LL):IFABS(U)=1ORDC=0THEN7
6 IFY=32ANDZ=32THENFORA=TC-U+LLTOBC-V-LLSTEPLL:POKEA,W3:NEXT
7 POKETC,TW:POKEBC,BW:POKETC+VP,CO:POKEBC+VP,CO
8 B=B+1:IFB<DWTHENTC=TC+U:BC=BC+V:GOTO7
9 IFABS(U)=1THENPOKETC,T1:POKEBC,B1:POKETC+VP,CO:POKEBC+VP,CO
10 IFTC=BC-LLTHENRETURN
11 FORB=TC+LLTOBC-LLSTEPLL:POKEB,SW:POKEB+VP,5:NEXT:RETURN
12 F=ML:U=LL+1:V=1-LL:T1=80:B1=122:SW=103
13 FORX=4TO6STEP2:TW=77:BW=76:B=0:IFX=4THEN15
14 F=MR:U=LL-1:V=-LL-1:SW=101:T1=79:B1=76:TW=78:BW=77
15 LM=M(F):LD=M(F)
16 IFF=3ANDOP=ECTHEN22
17 IFLD=0THENLD=2
18 IFQ-LD*INT(Q/LD)=0THENTC=M(X):BC=M(X+1):GOSUB5:GOTO21
19 TW=99:BW=100:U=U-LL:V=V+LL:N=PEEK(OP+LM)
20 IFN-D*INT(N/D)=0THENTC=M(X)+DW*LL:BC=M(X+1)+DW*-LL:GOSUB5
21 IFABS(U)=1THENU=U+LL:V=V-LL
22 M(X)=M(X)+DW*U:M(X+1)=M(X+1)+DW*V:NEXT
23 IFM=3ANDOP=ECTHENRETURN
24 IFQ-D*INT(Q/D)=0THEN27
25 OP=OP+M(M):Q=PEEK(OP):DC=DC+1:IFDC<CDTHEN12
26 IFDC=CDTHENRETURN
27 R=PEEK(M(4)-1)
28 IFR=32THENM(4)=M(4)-1:M(5)=M(5)-1:IFM(4)<>TL+DW*LLTHEN27
29 R=PEEK(M(6)+1)
30 IFR=32THENM(6)=M(6)+1:M(7)=M(7)+1:IFM(6)<>TR+DW*LLTHEN29
31 FORTC=M(4)TOM(6):POKETC,W1:POKETC+VP,CO:NEXT
32 FORBC=M(5)TOM(7):POKEBC,W2:POKEBC+VP,CO:NEXT:RETURN
33 PRINT"O="OX"LE";
34 GETA$:IFA$=""THEN34
35 PRINT
36 NP=P:Q=PEEK(P):IFA$="0"THENNP=P+M(M)
37 IFNP=PTHEN42
38 IFNP=ECTHENPRINT"ENDE!":GOSUB92:RUN
39 IFQ-D*INT(Q/D)<>0THENP=NP:PRINT"SIE GEHEN":GOTO48
40 IFRND(1)>.9THENPRINT"RA":OL=OL+1
41 PRINT"WAND GERAMMT":GOSUB91:GOTO51
```


Starship VC-20

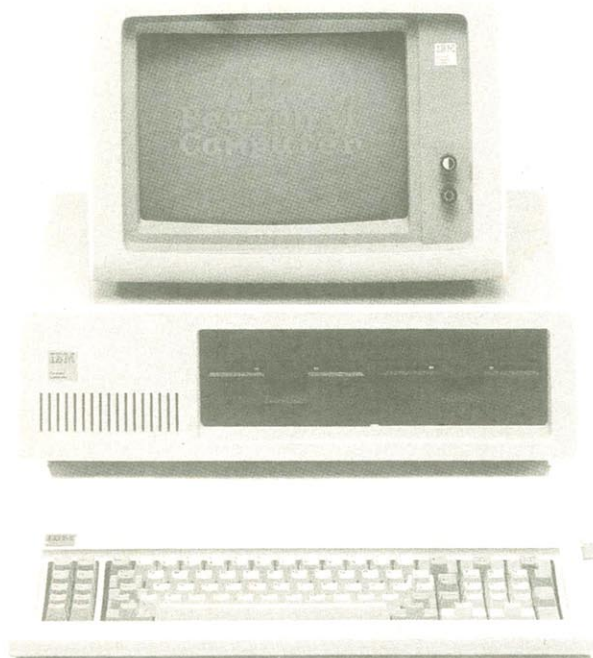


```
42 NM=M: IFA$="R" THEN NM=M+1
43 IFA$="M" THEN GOSUB 67
44 IFA$="L" THEN NM=M+3
45 IF NM=M THEN 33
46 IF NM>3 THEN NM=NM-4*INT(NM/4)
47 M=NM: PRINT "DREHEN"
48 D=W(M): Q=PEEK(P): OP=P: ML=M-1: MR=M+1
49 IF ML<0 THEN ML=3
50 IF MR>3 THEN MMR=0
51 OX=OX-OL: IF OX<0 THEN PRINT "LUFT=0": GOSUB 92: RUN
52 T=T+1: IFT=TT THEN TT=1: PRINT "LICHT": GOSUB 91
53 IFTT=1 THEN DC=0: GOSUB 90: GOSUB 12
54 IF P=MC AND RM=0 THEN RM=1: GOSUB 67
55 M(4)=TL: M(5)=BL: M(6)=TR: M(7)=BR: GOTO 33
56 P=P+1: IF P>EM THEN P=SM
57 A=INT(RND(1)*4): DC=0
58 A=A+1: DC=DC+1: IF DC>3 THEN 56
59 IFA>3 THEN A=0
60 M=P+M(A): IF M<SM OR M>EM THEN 58
61 CP=PEEK(P): CM=PEEK(M): IFC>0 AND CP=210 THEN P=M: GOTO 57
62 TM=M-SM: IF (CP=CM OR CM<210) AND DC>0 THEN 58
63 ME=TM-L*INT(TM/L): IF (ME=0 AND M(A)=1) OR (ME=0 AND M(A)=-1) THEN 58
64 OD=INT(15/W(A)): CP=CP/W(A): POKE P, CP: CM=CM/OD: POKE M, CM
65 P=M: C=C+1: IFC<H THEN 57
66 RETURN
67 GOSUB 90: PRINT "KARTE"
68 A=TL+LL+1: MS=A: DC=SM: DD=SM+0
69 FOR B=DC TO DD: FOR C=0 TO 3: CB=PEEK(B): CB=CB-W(C)*INT(CB/W(C))
70 AA=1: IFC=1 OR C=3 THEN AA=LL
71 IFC>1 THEN AA=-AA
72 BB=LL/AA: P2=A+AA: P1=P2+BB: P3=P2-BB
73 IF B=P THEN POKE A, W4: POKE A+VP, C0
74 IFCB=0 THEN POKE P1, W3: POKE P2, W3: POKE P3, W3
75 IFCB=0 THEN POKE P1+VP, C0: POKE P2+VP, C0: POKE P3+VP, C0
76 NEXT A: A=A+2: NEXT C: DC=DC+L: DD=DD+L: A=MS+(2*LL): MS=A
77 IF DD<=EM THEN 69
78 GOSUB 91: RETURN
79 REM
80 TL=7680: BL=8164: TR=TL+21: BR=BL+21: LL=22: DW=3: CD=3
81 W1=99: W2=100: W3=160: W4=88
82 W(0)=5: W(1)=7: W(2)=3: W(3)=2: L=INT(RND(1)*5+6)
83 M(0)=1: M(1)=L: M(2)=-1: M(3)=-L: M(4)=TL: M(5)=BL: M(6)=TR: M(7)=BR
84 W=INT(RND(1)*5+6): H=L*W-1: O=L-1: C=0: DC=0: T=0: RM=0
85 SM=7480: EM=SM+H: FOR A=SM TO EM: POKE A, 210: NEXT A: P=SM+INT(RND(1)*H)
86 EC=SM+INT(RND(1)*L): MC=INT(RND(1)*H)+SM
87 OX=INT(H/3)+1: IF P<OX THEN OX=-OX
88 OX=OX+H: OL=1: TT=INT(RND(1)*H/8)+1: GOSUB 56: GOSUB 90
89 M=0: NP=P: CE=PEEK(EC): CE=CE/W(3): POKE EC, CE: GOTO 48
90 PRINT "J": RETURN
91 FOR X=0 TO TL: NEXT X: RETURN
92 GET A$: IFA$="" THEN 92
93 RETURN
```

READY.

IBM PERSONAL COMPUTER

6 Monate Garantie



Systemeinheit 64 KB + Disklaufwerk 320 KB	6590,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
Grafik Drucker	1980,--
Druckerkabel	185,--
2. Laufwerk 320 KB	1495,--
DOS	137,50
Speichererweiterung	950,--
Komplettes System (wie vor)	13300,--

Systemeinheit 64 KB + Disklaufwerk 160 KB	5950,--
Tastatur	870,--
Bildschirm	1110,--
Adapter für Drucker und Bildschirm	1090,--
DOS	137,50
Komplettes System (wie vor)	8690,--



Computerversand

Albert-Schweitzer

3436 Hessisch L.

Alle genannten Preise incl. Mehrwertsteuer

H. Schädel

r-Straße 1

ichtenau

rtsteuer + Versandkosten

olivetti - M 20 -

Der eingeführte bewährte Personal Computer mit den
perfekten Problemlösungen



olivetti-Computer M 20
mit 160 KB
+ 2 Disklaufwerken à 320 KB **9950,--**



TI 99/4A

6 Monate Garantie

TI 99/4A 16 K RAM	699,--
Recorder	135,--
Kabel für Recorderanschluß	47,75
Sprachsynthesizer	349,--
Joysticks	95,--

Schach- programm Grandmaster – ein Meister seines Fachs

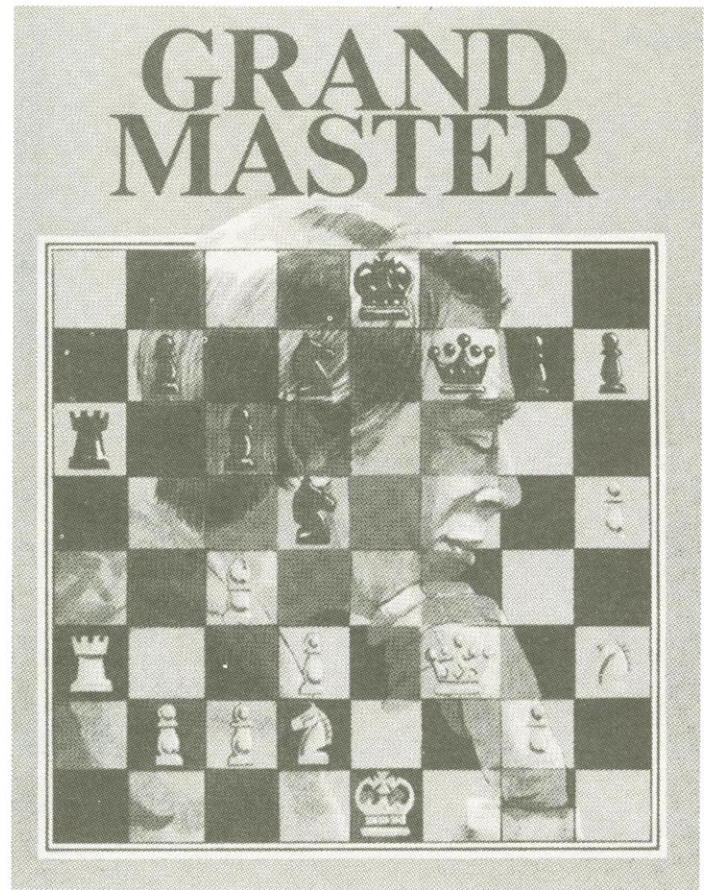
Seit es Computer gibt, übte das „Königliche Spiel“ eine starke Faszination und Herausforderung auf viele Leute aus, die sich mit dem Programmieren beschäftigen. War es anfangs selbst auf großen Anlagen kaum möglich, den Computer zu einem guten und für den geübten Spieler ernsthaften Gegner zu entwickeln, so ist dies heute dank der rasanten Entwicklung der Mikroprozessortechnik selbst bei den Kleinen möglich geworden.

Aus der Fülle der angebotenen Schachprogramme, die inzwischen auf dem Markt sind, fiel uns das 1982 von der Firma „Kingsoft“ entwickelte Programm GRANDMASTER auf. Grandmaster gibt es als Cassette für den VC-20 und VC-64. Für die VC-20 Version ist eine Speichererweiterung um mindestens 8 K notwendig.

Kingsoft selbst bezeichnet sein Programm als das „stärkste Schachprogramm für Heimcomputer“. Diese Behauptung möchten wir an dieser Stelle weder be-

stätigen noch in Frage stellen. Nachdem wir Grandmaster jedoch ausführlich getestet haben, können wir sagen, daß dieses Programm in punkto Handling, Grafik und Spielstärke sehr ansprechend und last not least auch vom Preis (79,-- DM unverbindl.) recht interessant ist.

Die gutverständliche deutsche Bedienungsanleitung rundet das Programm ab. Nach dem Laden des Programms waren wir von der Grafik und der übersichtlichen Anordnung der Bildschirmanzeigen äußerst



angetan. Für die Ästhetiker unter den Schachspielern wurden einige reizvolle Möglichkeiten, den Bildschirm nach persönlichem Geschmack einzurichten, eingebaut. So kann die Grafik mittels der Cursor-tasten in horizontaler und vertikaler Richtung exakt auf die Bildschirmmitte zentriert werden. Über die Funktionstasten können das Spielbrett sowie Vorder- und Hintergrund nach Lust und Laune jede mögliche Farbkombination erhalten. Zugegeben, eine Spielerei, aber dennoch sehr schön!

Doch nun zum eigentlichen Spiel: Es stehen 10 Spielstufen mit Rechenzeiten zwi-

schen 5 Sekunden und 2 Stunden zur Verfügung. Für Freunde des Briefschachs ist Stufe 0 vorgesehen, bei welcher der Computer Stunden oder unter Umständen sogar Tage für seine Zugentscheidung benötigt.

Eine Änderung der Spielstärke ist zu jeder Zeit über die Taste „S“ in Verbindung mit der entsprechenden Zahl möglich.

Während Grandmaster überlegt, zeigt er den momentan favorisierten Zug und wieviel Halbzüge er im Voraus analysiert hat. Betätigen der „Stop-Taste“ unterbricht seine Überlegungen und veranlaßt ihn, diesen Zug auszuführen. Diese Funktion ist sehr praktisch, wenn auf einer

CAR-CRASH

höheren Spielstufe gespielt wird und der Spieler nicht die endgültige Entscheidung des Computers abwarten möchte.

Die folgenden Möglichkeiten ergänzen die praktische Handhabung. Mit der Funktionstaste F2 kann auf automatisches Spiel umgeschaltet werden. Grandmaster spielt nun so lange gegen sich selbst, bis er ein Matt oder Patt findet. Für einen Anfänger zum Erlernen von Strategien ideal geeignet! Einfaches Betätigen der STOP-Taste führt wieder zum normalen Spiel zurück.

Um eine Figur zu setzen, wird lediglich die Start- und Zieladresse eingegeben (z. B. E2 E4) und RETURN gedrückt. Mittels der DEL-Taste können falsch eingegebene Zeichen gelöscht werden, wenn RETURN noch nicht gedrückt war. Auch nach dem RETURN-Befehl ist eine Zugrücknahme noch möglich und zwar über die ←-Taste. Für den ernsthaften Spieler ist dies natürlich Mogelei und daher nicht akzeptabel; für Anfänger jedoch unter Umständen lehrreich.

Grandmaster verfügt selbstverständlich über eine Eröffnungsbibliothek, die alle gängigen Varianten der Spieleröffnung enthält. CTRL Ø dreht das Spielbrett um; der Spieler kann mit Schwarz gegen den Computer spielen.

Alle Sonderzüge, Rochade, enPassant-Schlagen sowie die Bauernumwandlung sind möglich (Bauern werden grundsätzlich in eine Dame umgewandelt).

Unser Urteil: Ein sehr gutes, empfehlenswertes Programm, den meisten reinen Schachcomputern zumindest ebenbürtig, jedoch wesentlich preiswerter, als alles bisher angebotene.

Ein Spielprogramm für den VC-20 mit mindestens 8K Speichererweiterung.

Spielen können zwei Spieler gegeneinander oder ein Spieler allein gegen den Computer.

Ziel des Spieles ist es, das gegen den Uhrzeigersinn fahrende Auto mit den Tasten „“ und „.“ (innen,

außen) möglichst lange unfallfrei zu fahren. Wenn fast alle Punkte verbraucht sind, erhält man ein „Bonusauto“.

Beantwortet man die anfangs gestellte Frage nach der Spielerzahl mit „2“, so kann ein zweiter Spieler das im Uhrzeigersinn fahrende Crashauto gleichzei-

tig mit „C=“ (das ist die Commodore-Taste) und „Shift“ (innen, außen) steuern.

Vor dem Laden oder Eintippen muß

POKE44,32:POKE8182,0:
NEW eingegeben werden.

Georg Foltin

```
1 REM
2 REM CAR CRASH
3 REM
4 REM BY GEORG FOLTIN
5 REM
6 REM 1983
7 REM
8 REM LAENGE: 4.2K
9 REM
10 PRINT"#####"CHR$(8):POKE36879,25
20 FORI=1TO21
25 POKE646,RND(1)*6+2:PRINT" ";
30 NEXT
40 PRINT"###"
41 PRINTTAB(5)"###"
42 PRINTTAB(5)"# VC=20 "
43 PRINTTAB(5)"# "
44 PRINTTAB(5)"# CAR CRASH "
45 PRINTTAB(5)"# "
50 PRINT"#####":FORI=1TO21
60 POKE646,RND(1)*6+2:PRINT" ";:NEXT
70 GOSUB7000
80 PRINT"#####":PRINT"##### (1) := 1 SPIELER"
85 PRINT:PRINT"##### (2) := 2 SPIELER"
86 GETA$:IF A$<>"1"AND A$<>"2"THEN 85
87 Y=VAL(A$+"")
88 PRINT"#####BITTE WARTEN"
100 GOSUB5000
110 REM BEWEGUNG MENSCH
120 POKEPM,32
130 AC=PEEK(PM+V(IM))
140 IF AC=81 THEN GOSUB3000:ONS GOTO110,200
150 IF AC=32 THEN 200
160 IF PC=PM+V(IM) THEN 4000
170 IM=(IM+1)AND3
200 POKET,0:
205 PM=PM+V(IM)
210 POKEPM,C(IM)
215 IF PC=PM+V(IM) THEN 4000
220 POKEPC,0C
230 AC=PEEK(PC+V(IC))
240 IF AC=81 OR AC=32 THEN 300
250 IF PM=PC+V(IC) THEN 4000
260 IC=(IC-1)AND3
300 PC=PC+V(IC)
```

Starship VC-20

```

310 OC=PEEK(PC):POKEPC,C(1C)
315 IFPM=PC+V(1C)THEN4000
320 V=PEEK(197):V=(V=37)-(V=29)
330 IFV=0THEN450
340 V=V*(1+1)AND3
350 IFPEEK(PM+V)<32THEN450
355 POKET+1,190:POKET,245
360 POKERM,32:POKERM+V,C(1M)
370 IFPM+2*V=POTHEN4000
380 POKERM+V,32:PM=PM+2*V
390 POKERM,C(1M)
440 POKET+1,0:POKET,0
450 ONY60TO451,460
451 IFRND(1)>.6THEN550
455 V=SGN(RND(1)-.5)
456 GOTO470
460 V=PEEK(653):V=(V=2)-(V=1)
465 IFV=0THEN550
470 V=V*(1+1)AND3
480 IFPEEK(V+PC)<32THEN550
490 POKEPC,32:POKEPC+V,C(1C)
500 IFPC+2*V=POTHEN4000
510 POKEPC+V,32:PC=PC+2*V
520 POKEPC,C(1M)
550 GOTO110
3000 S=2
3010 SC=6C+1:POKET,230:POKET-1,0
3020 PRINTP$;"MID$(STR$(SC),2)
3030 ZZ=ZZ+1
3040 IFZZ<140THENRETURN
3044 FORI=1TO9:POKET,220:POKET+3,76
3046 POKET,0:POKET+3,25:FORO=0TO99:NEXT:NEXT
3050 ZZ=0:S=1:XX=XX+1
3055 POKET,0
3060 GOSUB3050
3070 PRINTP$;"M":XX-1
3080 RETURN
4000 POKEPC,32:POKERM,32:POKET+1,0
4010 FORI=10TO100STEP10
4015 POKET+1,10R128
4020 FORJ=0TO3
4030 POKEPC,C(J):POKEPC+33792,RND(1)*6+2
4035 FORTT=0TO1:NEXT
4040 NEXT:NEXT
4045 POKET+1,0:POKEPC+33792,6
4050 POKET+1,0:IFXX>0THENPM=4503:IM=0:PC=4295:IC=1:OC=32:PRINTP$;"M":XX-1:GOTO11
0
4060 POKE198,0
4065 PRINT "#####M INT#####AT52"
4070 P=4502:I=0:F=0
4080 IFC=64ORC=93THENI=(1+1)AND3:F=(F+1)AND7:IFFC>6THENPOKE36879,17*F+8
4085 FORTT=0TO40:NEXT
4090 POKEP,32:P=P+V*(I)
4095 GETR$:IFR$=""THEN4075
4100 GOSUB5045:POKE36879,45
4110 GOTO110
5000 FORI=0TO511:POKEI+6144,PEEK(I+32768):NEXT
5020 READC:IFC=-1THEN5045
5030 A=6144+8*C
5040 FORI=0TO7:READC:POKEI+A,C:NEXT:GOTO5020
5045 XX=3:SC=0:ZZ=0
5050 POK36865,206:POKE36879,25:POKE36878,15:T=36876
5055 PM=4503:IM=0:PC=4295:IC=1:OC=32:P$="#####"
5060 PRINT "###"
5065 C(0)=107:C(1)=113:C(2)=115:C(3)=114:V(0)=1:V(1)=-22:V(2)=-1:V(3)=22
5070 POK650,128
5080 PRINT "#####
5090 PRINT "#####
5100 PRINT "#####
5110 PRINT "#####
5120 PRINT "#####
5130 PRINT "#####
5140 PRINT "#####
5150 PRINT "#####
5160 PRINT "#####
5170 PRINT "#####
5180 PRINT "#####
5190 PRINT "#####
5200 PRINT "#####
5210 PRINT "#####
5220 PRINT "#####
5230 PRINT "#####
5240 PRINT "#####
5250 PRINT "#####
5260 PRINT "#####
5270 PRINT "#####
5280 PRINT"### @CRASH!"
5300 POKERM,C(1M):POKEPC,C(1C)
5310 RETURN
7000 POKE36878,15:FORI=0TO1
7010 RESTORE
7020 FORJ=1TO22
7030 READH,L
7040 POKE36875+I,H
7050 FORK=1TO300/L
7060 NEXTK
7070 POKE36875+I,0
7080 FORK=1TO10
7090 NEXTK
7100 NEXTJ
7110 NEXTI
7999 RETURN
9046 POKET,0:POKET+3,25:FORO=0TO99:NEXT:NEXT
10000 DATA201,2,191,2,201,4,201,4,191,2,183,4,183,4,175,2,201,
2
10010 DATA191,2,201,4,201,4,191,2,183,4,183,4,175,2,175,2
20000 DATA64,0,0,255,0,0,255,0,0
20010 DATA93,36,36,36,36,36,36,36,36
20020 DATA75,36,36,36,196,4,0,240,0,0
20030 DATA100,0,0,240,0,0,240,0,0
20040 DATA74,36,36,35,32,16,15,0,0
20070 DATA73,0,0,240,8,4,196,36,36
20080 DATA85,0,0,15,16,32,55,36,36
20100 DATA123,0,0,15,16,16,15,0,0
20110 DATA126,36,36,36,36,24,0,0,0
20120 DATA124,0,0,0,24,36,36,36,36
20130 DATA113,24,90,136,90,24,219,255,219
20140 DATA114,219,255,219,24,90,126,90,24
20150 DATA107,224,238,68,255,255,68,238,224
20160 DATA115,7,119,34,255,255,34,119,7
20170 DATA81,0,0,0,24,24,0,0,0
20180 DATA48,28,51,51,0,51,51,28,0
20190 DATA49,0,3,0,3,0,3,0,0
20200 DATA50,28,3,3,28,48,48,28,0
20210 DATA51,28,3,3,28,3,3,28,0
20220 DATA52,0,51,51,28,3,3,0,0
20230 DATA53,28,48,48,28,3,3,28,0
20240 DATA54,28,48,48,28,51,51,28,0
20250 DATA55,28,3,3,0,3,0,0
20260 DATA56,28,51,51,28,51,51,28,0
20270 DATA57,28,51,51,28,3,3,28,0
20280 DATA84,255,255,24,24,24,24,24
20290 DATA69,255,255,192,255,255,192,255,255
20300 DATA72,195,195,195,255,255,195,195,195
20310 DATA68,252,252,195,195,195,252,252
20320 DATA78,195,195,243,243,207,207,195,195
20330 DATA96,170,85,170,85,170,85,170,85,-1
30000 REM (C) BY
30001 REM GEORG FOLTIN
30002 REM &
30003 REM CORD KIELHORN
30004 REM 1983
READY.

```



Ritter

Ihre Aufgabe ist es eine Burg einzunehmen. Sie können sich mit den Tasten:

- I ← links
- E → rechts
- P  Schild
- L  Schuß bewegen.

Sinn des Spieles ist es, die Kette der Zugbrücke durchzuschießen, indem Sie ganz nach vorne zum Graben gehen und dann auf „Teufel komm raus“ losfeuern.

Es ist natürlich klar, daß sich die Burgbewohner so etwas nicht gefallen lassen. Sie werden sich mit Bogenschützen und Wunderwolken verteidigen.

Wenn ein Pfeil auf Sie zufliegt, müssen Sie sich duk-

ken „P“. Bei der Wunderwolke schützen Sie sich entweder mit dem Schutzschild „P“ vor dem Pechregen, aber besser ist es ihm auszuweichen.

Sie sollen die Kettenbefestigung 16 mal treffen. Wenn Sie sich ducken oder das Schutzschild benutzen, wird Ihnen jeweils 1-20 Treffer abgezogen.

Sie fangen mit 0 Treffern an.

Sie sollen 16 Treffer erreichen und können maximal -6 Treffer durch ducken oder Schutzschildbetätigung bekommen. Jedemal wenn Sie sich ducken wird nämlich die Kette erneuert bzw. repariert.

Zum Programm:

Als erstes geben Sie ein:
1 Rem... (19. Punkte. ein).
N/L

Danach schreiben Sie
Poke 16510,0 N/L
und dann geben Sie das

Listing wie gewohnt der Reihe nach ein.

Bemerkungen:

Bevor Sie Ihren ersten Probelauf machen, save Sie lieber das Programm, denn wenn in den ersten 7 Zeilen ein Fehler ist, kann dann das Programm gelöscht werden, wenn das MC-Programm in Zeile 560 und 4010 abgerufen wird.

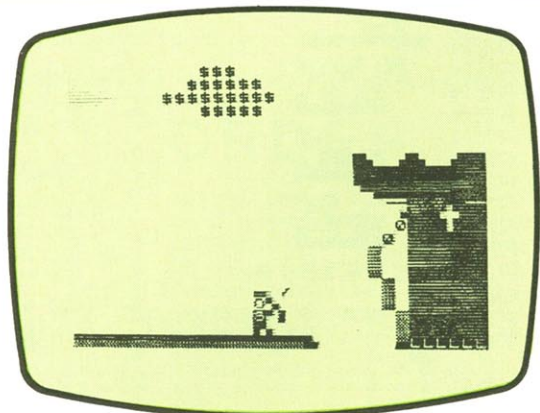
In Zeile 510 wird man in den Gefahrenabschnitt ge-

bracht. Durch verändern des Wertes hinter dem „RND“ beeinflusst man die Gefahren.

In Zeile 570 wird bestimmt, ob man dem Bogenschützen oder der Wolke ausgesetzt wird.

In der Zeile 420 werden die Inputzeilen zu den Ausdruckszeilen dazugegeben (Bildschirm) (24-Zeilen-Bildschirm)

Stefan Benk



Die ZX-Seiten

```

0 REM EERNND: 477$4 NEXT TAN
1 PRINT AT 0,0;" EROBERN SI
E EINE BURG... GE
STEUERT WIRD MIT:
1.....L
E.....RE
CHTS.....3C
HUSS.....5C
HUTZ" ,TAB 3;"UIEL GLUECK"
2 RAND
3 LET A$="A20004E081600232E7E
E0877019F32D0023F9C"
4 FOR A=16514 TO 16532
5 POKE A,CODE A$+16*CODE A$(2
)-476
6 LET A$=A$(3 TO )
7 NEXT A
8 FOR A=1 TO 50
9 NEXT A
10 LET A$=""
11 LET A$=A$+" "
20 LET A$=A$+" "
30 LET A$=A$+" "
40 LET A$=A$+" "
50 LET A$=A$+" "
60 LET A$=A$+" "
70 LET A$=A$+" "
80 LET A$=A$+" "
90 LET A$=A$+" "
100 LET A$=A$+" "
110 LET A$=A$+" "
120 LET A$=A$+" "
130 LET A$=A$+" "
140 LET A$=A$+" "
150 LET A$=A$+" "
160 LET A$=A$+" "
170 LET A$=A$+" "
180 REM " "
190 REM " "
200 REM " "
210 REM " "
220 REM " "
230 REM " "
240 REM " "
250 REM " "
260 REM " "
270 REM " "
280 REM " "
290 REM " "
300 REM " "
310 REM " "
320 REM " "
330 REM " "
340 REM " "
350 REM " "
360 REM " "
370 REM " "
380 REM " "
390 REM " "
400 REM " "
410 REM " "
420 POKE 16418,0
425 CLS
430 PRINT AT 7,0;A$;AT 17,3;"
"/;TAB 3;" 0";TAB 3;" " ";
TAB 3;" "
440 LET A=4
450 LET L=0

```

```

460 LET S=0
470 LET D=0
480 LET B=1
490 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
500 PRINT AT 17,A;" / ";TAB A
;" 0";TAB A;" " ";TAB A;" "
510 IF RAND<.15 THEN GOSUB 550
520 LET D=INKEY$="L"
530 IF D OR S THEN GOSUB 1080
540 GOTO 490
550 REM "ACHTUNG"
560 PRINT TAB USR 16514;AT 22,2
;"ACHTUNG...";TAB USR 16514
570 IF RAND>.5 THEN GOTO 730
580 REM "GEGENSCHUETZE"
590 PRINT AT 16,24;" ";TAB
24;" <-";TAB 22;" ";TAB
22;" "
600 FOR C=31 TO 0 STEP -2
610 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
620 IF B THEN PRINT AT 17,A;"
/ ";TAB A;" 0";TAB A;" " ";
TAB A;" "
630 IF NOT B THEN PRINT AT 17,A
;" ";TAB A;" " ";TAB A;" "
640 LET B=INKEY$<>"P"
650 IF NOT B THEN LET L=L-(L>-6
)
655 LET D=INKEY$="L"
660 IF D OR S THEN GOSUB 1080
670 IF C<25 THEN PRINT AT 17,C;
"<- "
680 IF ABS (A-C+1)<2 AND B THEN
GOTO 2000
690 NEXT C
700 PRINT AT 17,0;" "
710 PRINT AT 22,2;" ";
AT 16,24;" ";TAB 24;" ";
";TAB 22;" ";TAB 22;" "
720 RETURN
730 REM "DECH"
740 FOR C=22 TO A-2 AND A>3 STE
P -1
750 PRINT AT 0,C+3;"$$$ ";TAB C
+2;" $$$$ ";TAB C;" $$$$$$$$ ";
TAB C+3;" $$$$ "
760 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
770 LET B=INKEY$<>"P"
775 IF NOT B THEN LET L=L-(L>-6
)
780 IF B THEN PRINT AT 17,A;"
/ ";TAB A;" 0";TAB A;" " ";
TAB A;" "
790 IF NOT B THEN PRINT AT 17,A
-1;" ";TAB A-1;" " ";TAB
A;" " ";TAB A;" "
795 LET D=INKEY$="L"
800 IF D OR S THEN GOSUB 1080
810 NEXT C
820 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
830 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" "
";TAB 18;" ";TAB 17;" "
840 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
850 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" "
";TAB 18;" ";TAB 17;" "
860 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" ";TAB
18;" ";TAB 17;" "
870 PRINT AT 6,20;" ";TAB 19;"
";TAB 18;" ";TAB 17;" "
";TAB 18;" ";TAB 17;" "
880 PRINT AT 0,C+3;" " ";TAB
C+2;" " ";TAB C+1;" "

```


ALLES
FÜR

ZX81 UND SPECTRUM

VON
F+K



TUNNEL

Reaktionsspiel mit superschneller 3D-Farbgraphik. Für SPECTRUM 16K/48K. Was liegt im TUNNEL verborgen? DM 29,-

ESCAPE

3D-Adventure. Bis zu 5 Monster sind hinter Ihnen her, während Sie den Ausgang des Irrgartens suchen. Tolle Graphik! Für SPECTRUM 16K/48K.

nur DM 27,-



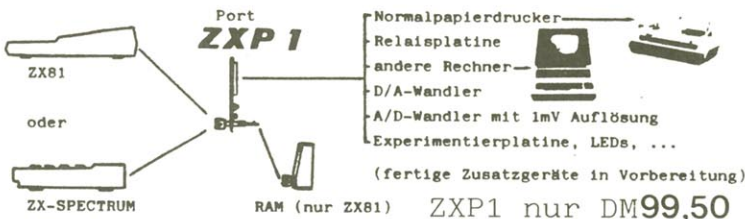
3D DEFENDER für ZX81/16K BestNr 152
Vergessen Sie alles, was Sie bisher an Weltraumspielen gesehen haben. Bei 3D DEFENDER schauen Sie aus Ihrem Cockpit und sehen -dank der superschnellen Graphik- alles wie bei einem Flugsimulator vor sich. Und das in dreidimensionaler Darstellung. Mit 8 Flugrichtungen, Radar, UFOs, Plasmaangriffen, ... Das müssen Sie sehen!
"Another 3D winner" schreibt SINCLAIR USER.

3D MONSTER MAZE für ZX81/16K BestNr 151
Unglaubliche Graphik! COMPUTER & VIDEO GAMES schreibt: "3D MONSTER MAZE is the best game I have seen for the ZX81". Finden Sie Ihren Weg heraus aus dem Irrgarten, während REX Ihnen nachjagt? Wände, Gänge und REX selbst sind dreidimensional dargestellt, sowas ist noch nie dagewesen. Nur bei uns, nur DM 29,-



"If I had to choose just one programme to impress an audience with the capabilities of the ZX81, then 3D MONSTER MAZE would be the one without doubt" schreibt ZX COMPUTING.

Port ZXP1 für ZX81, ZX-SPECTRUM BestNr 400
Technische Daten: Eingabe 8 Bit, Ausgabe 8 Bit, zusätzlich 5 STROBES. Hochstromausgänge (low=24mA), Schmitt-Trigger-Eingänge. Anschlüsse steckbar. Fertiger mit ohne Gehäuse mit ausführlicher Anleitung.
Besonderheiten: für den Druckeranschluß (GPI00A) vorbereitet (CENTRONICS)



Zusätzlich kann auch der SINCLAIR-Drucker angeschlossen werden.

Wir führen auch **HARDWARE**
S. CHIP 4/83 S. 179

Der Knüller APPEND

APPEND ermöglicht das Zusammenladen von verschiedenen BASIC-Programmen. Außerdem ist noch eine RENUMBER-Funktion eingebaut, die Ihre Programme oder Teile davon neu durchnummerieren kann (auch GOTO, GOSUB und RUN).

Unser APPEND ist das einzige derartige Programm, mit dem man z.B. auch 2 Programme á 5 KByte zusammenladen kann. Andere Programme gleichen Namens können das nicht!!

APPEND ist nicht billig, denn Qualität hat ihren Preis, aber es lohnt sich!

APPEND für ZX81/16K DM 45,-

DBMS 81

Programm zur Erzeugung beliebiger Dateien/Kartellen. Unterstützt Sie nach der Erzeugung bei der Verwaltung Ihrer Daten und ermöglicht durch eine spezielle Speicherart eine bessere Ausnutzung des RAMs. Z.B. werden über 15500 Zeichen im 16K-RAM verwaltet, im 64K-RAM sind es über 60000 !!!!

Ähnliche Systeme verwalten sonst nur ca. 10000 Zeichen im 16K-RAM. Weitere Details in unserer Info. DBMS81 für ZX81/mindestens 16K äußerst preiswert nur DM 45,-

ADVENTURE 1: INKA - TEMPEL



Nur bei uns: das erste SPECTRUM-Adventure in deutsch und in MC. Sie bannen sich Ihren Weg durch 53 Räume, um die 7 Schätze des Tempels einzusammeln. Aber das ist schwerer, als es Ihnen jetzt vorkommt!!! Nur für 48K-SPECTRUM.

nur DM
29,-

Alle diese Produkte kommen aus dem Hause Wolfgang Fleischhauer und Nils Körber GbR.

Wollen Sie mehr wissen? Wir schicken Ihnen gerne eine gegen Voreinsendung eines selbstadressierten Freiumschlages DIN C6.

Schicken Sie diesen an:

» F+K-WARE, Abt. H
Rebenacker 1a, 2000 Hamburg 54 «

SPECTRUM-Benutzer fordern bitte 'Liste S' an.

für ZX81/16K:
Spieler-Klassiker

- ASTEROIDEN DM 21,-
- INVADERS DM 21,-
- BREAKOUT* DM 12,95
- FUCKMAN DM 27,-
- CENTIPEDE DM 29,-

*auch für ZX81/1K

ORDER

Bestellungen bitte auch an die vorgenannte Adresse richten. Versandpauschale: DM 6,-. Alle genannten Preise enthalten die ges. MwSt und sind gültig bis 30.9.83.

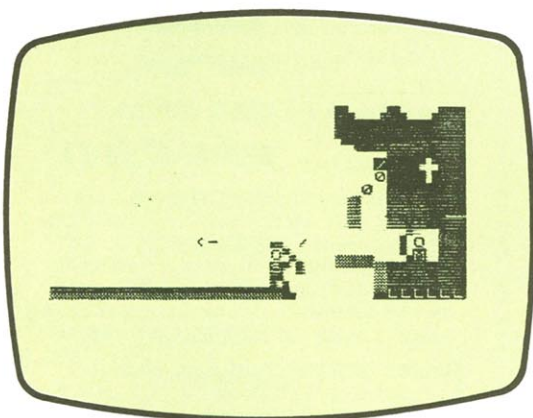
Händleranfragen erwünscht!!

Die ZX-Seiten

```

;TAB C+3;" "
881 PRINT AT 0,C+3;" ";TAB C+2
;" " ;TAB C+1;" " ;T
AB C+3;" "
882 PRINT AT 1,C+2;" ";TAB
C+1;" " ;TAB C+3;" " ;T
883 PRINT AT 1,C+2;" ";TAB
C+1;" " ;TAB C+3;" " ;T
884 PRINT AT 3,C+3;" " ;TAB C
+3;" "
885 PRINT AT 4,C+3;" " ;TAB C
+3;" "
890 FOR Z=3 TO 18 STEP 2
900 LET A=A-(INKEY$="1" AND A>0
)+(INKEY$="E" AND A<16)
910 LET D=INKEY$="L"
920 IF D OR S THEN GOSUB 1080
930 LET B=INKEY$<>"P"
940 IF B THEN PRINT AT 17,A;"
/" ;TAB A;" 0" ;TAB A;" " ;
TAB A;" "
950 IF NOT B THEN PRINT AT 17,A
-1;" " ;TAB A-1;" 0" ;TAB
A;" " ;TAB A;" "
960 IF NOT B THEN LET L=L-(L>-6
)

```



```

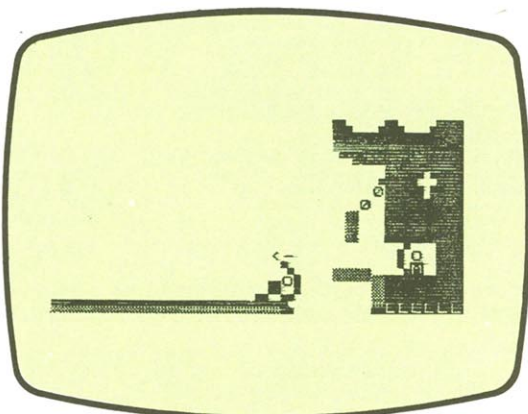
970 PRINT AT Z,C+4;" " ;AT Z-2,C
+4,
980 NEXT Z
990 PRINT AT Z-2,C+4,
995 PRINT AT 22,0;" "
;
1000 IF ABS (A-C-3) > 1.6 OR NOT B
THEN RETURN
1005 LET L=RND**RND**RND
1010 PRINT AT 17,A-1;" " ;TA
B A-1;" " ;TAB A-1;" " ;
TAB A-1;" " ;TAB A;" " ;
1015 LET L=RND**RND**RND
1020 PRINT AT 17,A;" " ;TAB A-
1;" " ;TAB A-1;" " ;TAB
A-1;" " ;TAB A-1;" " ;M ;TA
B A;" "
1025 LET L=RND**RND**RND
1026 PRINT AT 13,A;" " ;TAB A-
1;" " ;TAB A-1;" " ;TAB
A-1;" " ;TAB A-1;" " ;M ;TA
B A;" "
1027 LET L=RND**RND**RND
1030 PRINT AT 0,0;" DER " "RECH-RE
SEN" " HAT SIE "ER
SCHLAGEN" "
1040 GOTO 2050
1080 IF S THEN GOTO 1140
1090 LET H=18
1100 FOR S=A+3 TO A+9
1110 LET H=H-1
1120 PRINT AT H+1,S-1,AT H,S;" /"
;
1130 RETURN
1140 NEXT S
1150 PRINT AT H,S-1,
1160 LET L=L+(S=26)
1170 IF S=26 THEN PRINT AT 11,25
;CHR$ (L+156)

```

```

1180 LET S=0
1190 IF L=16 THEN GOTO 4000
1200 LET D=0
1210 RETURN
2000 PRINT AT 17,A;" " / ;TAB A
;" 0" ;TAB A;" " S " ;TAB A;" "
2005 LET L=RND**RND**RND
2010 PRINT AT 17,A;" " / ;TAB A
;" 0" ;TAB A;" " S " ;TAB A;" "
2015 LET L=RND**RND**RND
2020 PRINT AT 17,A;" " / ;TAB A
;" 0" ;TAB A;" " S " ;TAB A;" "
2025 LET L=RND**RND**RND
2030 PRINT AT 17,A;" " ;TAB A
;" " ;TAB A;" / " ;TAB A-1;" 0
"
2035 LET L=RND**RND**RND
2040 PRINT AT 0,0;" UOM BOGENSCHU
ETZEN ERWISCHT"
2050 PRINT AT 10,0;" NOCHMAL?"
2060 RUN (2060 AND INKEY$<>"J")
4000 FOR Z=0 TO 30
4010 LET L=USR 16514+USR 16514
4020 NEXT Z
4030 PRINT AT 22,1;" GENONNEN."
;TAB C;" "
4040 FOR Z=14 TO 20
4050 PRINT AT Z,23;" "
4060 NEXT Z
4070 PRINT AT 20,19;" "
4080 PRINT AT 21,18;" "
4090 FOR A=A TO 25
4100 PRINT AT 17,A;" " / ;TAB A
;" 0" ;TAB A;" " S " ;TAB A;" "
4110 NEXT A
4115 PRINT AT 1,0;" HERZLICHEN
WELKOMMEN"
4120 PRINT AT 7,0;A$( TO LEN A$-
64)
4130 PRINT AT 10,0;" NOCHMAL?"
4140 RUN (4150 AND INKEY$<>"J")
4150 PRINT AT 6,22-6;" " ;0

```



```

;"TAB 22;" " S "
4155 LET L=RND**RND
4160 PRINT AT 6,22-6;" " 0
;"TAB 22;" " S "
4170 PRINT AT 6,22-6;" " 0
;"TAB 22;" " S "
4180 PRINT AT 6,22-6;" " 0
;"TAB 22;" " S "
4190 PRINT AT 6,22-6;" " 0
;"TAB 22;" " S "
4200 PRINT AT 6,22-6;" " 0
;"TAB 22;" " S "
4210 GOTO 4140
9940 STOP
9950 SAVE "RITTER"
9960 RUN

```


Mondlandung VC-64

Ein Raumschiff soll auf der Oberfläche des Mondes weich gelandet werden. Bei zu hartem Aufsetzen, bei zu starker seitlicher Abdrift oder beim Berühren der Felsen explodiert es. Wenn es zu weit aus dem Bildschirm verschwindet, hat es sich verirrt, was ebenfalls Minuspunkte bringt. Die Punktzahl errechnet sich nach der Schwierigkeit des Landeplatzes, nach der Entfernung des Landeplatzes vom Ausgangspunkt, nach dem noch vorhandenen Treibstoffvorrat und nach der Landegeschwindigkeit. Bei mehreren Spielen erscheint für jeden Teilnehmer im selben Durchgang die gleiche Landschaft, die vom Zufallsgenerator gezeichnet wird.

Die Steuerung erfolgt wahlweise mittels Joystick (in Control-port 1) rechts und links mit dem Steuerknüppel, die Bremsraketen werden mit der Feuertaste gezündet, oder über die Tastatur. Nach links mit ←, nach rechts mit →. Gebremst wird mit der Space-taste. Zubeachten ist dabei die Trägheit, d.h. das Raumschiff reagiert nicht sofort, sondern mit Verzögerung. Mit S kann der Countdown unterbrochen werden, mit der Spacetaste oder mit dem Feuerknopf wird er abgekürzt.

Die Schwierigkeitsgrade können in folgenden Zeilen verändert werden:

Zeile 2

T% = Treibstoffmenge

Zeile 5106

V = Fallgeschwindigkeit

Zeile 5107

RX = seitliche Abdrift




```

D
4040 IFHI<STHENHI=S
4050 SP=0:S=0:GOTO3
5000 FORA=1TO800:NEXT:POKESI+P(0),0:POKEP(0),87
5010 FORA=1TOL:GETX$:IFSO=1THENNEXT:SO=0:GOTO6000
5015 IFPEEK(J0)<>127ORPEEK(203)<>64THENS0=1
5020 FORB=1TO100:NEXT:POKESI+P(A),0:POKEP(A),91:POKEP(0),87:NEXT
5030 FORA=1TO1000:NEXT:POKE53281,0:PRINTCHR$(14)"TER OORM HAT ";
5040 PRINTP$(1);FORA=1TO1500:NEXT:PRINT"J":FORA=1TO200:PRINTP$(2);:FORB=1TO15
5050 NEXT:NEXT:FORA=1TO1000:NEXT
6000 W=W-1:IFW=0THEN4000
6010 POKE53281,FU:PRINTCHR$(142);"J":L=LW:SP=0:GOTO100
20000 POKE649,9:PRINTCHR$(14)"LITTE WAEHLEN"
20010 PRINT,"PIELBEGINN"
20013 PRINT,"1":PRINT,"2 OECHESEL JOYSTICK/KEYBOARD"
20015 PRINT,"3":PRINT,"4 OERTE AENDERN"
20016 PRINT,"5":PRINT,"6 ENDE"
20019 PRINT,"7 MITTE ZUMMER EINGEBEN"
20020 GETN$:IFPEEK(J0)<>127THEN20031
20021 IFN$=""THEN20020
20025 IFN$="1"ORN$="2"ORN$="3"ORN$="4"THENA=VAL(N$):GOTO20030
20026 GOTO20020
20030 ONAGOTO20031,20032,20040,4035
20031 PRINTCHR$(142);"J":L=LW:F=FS:G=GS:T=TS:GOTO100
20032 IFJ$="J"THENJ$="K":GOTO20000
20033 IFJ$="K"THENJ$="J":GOTO20000
20040 PRINT"J" LITTE WAEHLEN
20050 PRINT,"1 ANZAHL DES LIFTS":PRINT,"2 ANZAHL DES LUTTERS"
20060 PRINT,"3":PRINT,"4 ESCHWINDIGKEIT":PRINT,"5"
20070 PRINT,"6 RICHTUNGSSTEUERUNG":PRINT,"7":PRINT,"8 OORMLAENGE"
20073 PRINT,"9"
20075 PRINT,"10":PRINT,"11 OERTE IN FRONUNG"
20080 GETN$:IFN$=""THEN20080
20090 IFN$="1"ORN$="2"ORN$="3"ORN$="4"ORN$="5"ORN$="6"THEN20100
20091 GOTO20080
20100 A=VAL(N$):ONAGOTO20120,20140,20160,20180,20200,20000
20120 PRINT"ANZAHL DES LIFTS:","ALTER OERT:":GS:PRINT,,"/EUER OERT":
20130 INPUTGS:GOTO20040
20140 PRINT"ANZAHL DES LUTTERS:","ALTER OERT:":FS:PRINT,,"/EUER OERT":
20150 INPUTFS:GOTO20040
20160 PRINT"ESCHWINDIGKEIT:","ALTER OERT:":TS:PRINT,,"/EUER OERT:":
20170 INPUTTS:GOTO20040
20180 PRINT"RICHTUNGSSTEUERUNG: ALTE OERTE:":PRINT,"UFWAERTS",A$
20181 PRINT,"BWAERTS",B$:PRINT,"LINKS",C$:PRINT,"LECHTS",D$
20182 PRINT,"/EUE OERTE:":PRINT,"UFWAERTS",
20183 GETA$:IFA$=""THEN20183
20184 PRINTA$:PRINT,"BWAERTS",
20185 GETB$:IFB$=""THEN20185
20186 PRINTB$:PRINT,"LINKS",
20187 GETC$:IFC$=""THEN20187
20188 PRINTC$:PRINT,"LECHTS",
20189 GETD$:IFD$=""THEN20189
20190 PRINTD$:FORA=1TO1000:NEXT:GOTO20040
20200 PRINT"OORMLAENGE:","ALTER OERT":LW:PRINT,,"/EUER OERT (MAX10)":
20210 INPUTL:GOTO20040
50000 PRINT"J", "LITTE WAEHLEN"
50010 PRINT,"1 ANZAHL DES LIFTS", "2 ANZAHL DES LUTTERS"
50015 B=-2.7:FORA=14TO9STEP-1:B=B+2.79
50020 PRINTTAB(A)"EIN":FORC=1TO8:PRINT" ";:NEXT:PRINT"SPIEL":
50030 FORC=1TO200:NEXT:IFAC14THEN50050
50040 PRINT,"1 VON":PRINT," DENNIS MERBACH:J":FORC=1TO2000:NEXT
50050 NEXT:FORA=1TO1000:NEXT:PRINTTAB(13)"GROSSARTIGES":FORA=1TO2000:NEXT
50060 PRINT"JOYSTICK ODER KEYBOARD (J/K)"
50070 GETJ$:IFPEEK(J0)<>127THENJ$="J":GOTO2
50075 IFJ$=""THEN50070
50080 IFJ$="J"ORJ$="K"THEN2
50090 GOTO50070
READY.

```



```
2502 IF(JAND16)=0ANDSW<=0ANDSY=221THENSW=1
2503 IFSW>0THENSY=SY-16:SW=SW-.06
2504 IFSW<=0THENSW=0:SY=221: SX=PJ
2506 GETA#: IFA#><"V" THEN2512
2510 GOTO2510
2512 IFPJ>2360RPJ<24THEN5000
2513 POKE53249,SY:POKE53278,0:POKE53248,SX:POKE53250,PJ:C=C+.015
2514 SC=PEEK(53278): IF(SCAND8)=8THENP3=40:SW=0
2515 P3=P3+1.6+C:POKE53255,P3:IFP3>=200THENA=3:GOTO6000
2516 IF(SCAND16)=16THENP4=40:SW=0
2517 P4=P4+1.3+C:POKE53257,P4:IFP4>=200THENA=4:GOTO6000
2518 IF(SCAND32)=32THENP5=40:SW=0
2519 P5=P5+2.0+C:POKE53259,P5:IFP5>=200THENA=5:GOTO6000
2520 IF(SCAND64)=64THENP6=40:SW=0
2521 P6=P6+1.6+C:POKE53261,P6:IFP6>=200THENA=6:GOTO6000
2522 IF(SCAND128)=128THENP7=40:SW=0
2523 P7=P7+1.2+C:POKE53263,P7:IFP7>=200THENA=7:GOTO6000
2524 PRINT"♣",MID$(TI$,4,1);": "RIGHT$(TI$,2):GOTO2500
5000 IFPJ<24THENPJ=24:GOTO2513
5010 T$=TI$:PRINT"♣♣♣♣ SIE HABEN SICH ERTRAENKT!"
5020 POKEVC+1,221:FORA=237TO255:POKEVC+2,A:POKEVC,A:NEXT
5030 FORA=221TO255:POKEVC+3,A:POKEVC+1,A:NEXT
5040 PU=VAL(MID$(T$,3,2))*60+VAL(RIGHT$(T$,2))
5050 PRINT,"♣♣♣SIE HABEN SICH":PRINT,PU"SEKUNDEN"
5055 PRINT"
GEHALTEN"
5060 PRINT"
ALTER REKORD:";HI;"SEKUNDEN":IFPU>HITHENHI=PU
5090 WAIT56321,16,127:FORA=3TO7:POKEVC+2*A,A*45-100:POKEVC+2*A+1,28:NEXT
5100 POKEVC,150:POKEVC+1,221:POKEVC+21,255:POKEVC+2,150:POKEVC+3,221:SY=221:SW=0
5110 FORA=1TO1500:NEXT:GOTO2000
6000 T$=TI$:P=PEEK(VC+A*2):POKEVC,PJ:POKEVC+1,221:FORB=202TO221:POKEVC+A*2+1,B
6010 NEXT:RG=-1:IFPJ>PTHENRG=1
6020 POKEVC+A*2,P
6030 IFP<>INT(PJ)THENP=P+RG:GOTO6020
6040 P=P+2:POKEVC+A*2,P:POKEVC+2,P:POKEVC,P
6050 IFP<254THEN6040
6060 PRINT"♣♣♣ DAS GESPENST HAT SIE ERTRAENKT!":GOTO5020
10000 POKEVC+21,15:POKEVC+23,15:POKEVC+29,15
10010 POKEVC,40:POKEVC+1,208:POKEVC+2,40:POKEVC+3,208:POKEVC+6,30:POKEVC+7,50
10020 POKEVC+16,4:POKEVC+4,20:POKEVC+5,50
10030 PRINTCHR$(14),"♣ LEISTERJAEGER"
10040 PRINT"♣♣♣♣ "IE IESPENSTER MOECHTEN ♣IE FANGEN!"
10050 PRINT"
XERSUCHEN ♣IE, SIE DARAN ZU HINDERN,"
10060 PRINT"
INDEM ♣IE SIE MIT DEM ♣PEER TREFFEN."
10070 PRINT"
♣IE KOENNEN SICH NACH LINKS UND NACH "
10090 PRINT"
RECHTS BEWEGEN, ENTWEDER MITTELS"
10100 PRINT"
'1' UND '2' ODER MITTELS 'OYSTICK,"
10110 PRINT"
UND SIE KOENNEN MIT DER '♣PACE'~ "
10120 PRINT"
BZW MIT DER 'EUERTASTE DEN ♣PEER":PRINT"
WERFEN."
10130 PRINT"♣
~ALLEN ♣IE NICHT IN DEN IRABEN!"
10140 PRINT,"
ALERTIG?"
10150 GETA#: IFA#="" THEN10150
10160 PRINTCHR$(142);"J":POKEVC+23,0:POKEVC+29,0:POKEVC+21,0:POKEVC,0:POKEVC+2,0
10170 GOTO113
```

READY.

VC-64-Sprites-Generator

Sprite-Grafik ist eine verhältnismäßig neue Form, um Zeichen und Figuren über den Bildschirm zu bewegen. Beim VC-64 lassen sich Sprites zusätzlich zum normalen Videobild auf dem Bildschirm darstellen.

Das Programm soll das Erstellen von Sprites erleichtern, indem es durch die Cursorsteuertasten und die Space-Taste in einer 24*21-Matrix die Kon-

struktion von Sprites erlaubt.

Nach Erstellung aller Sprites werden die entsprechenden Bytes in Datazeilen übernommen und das Hilfsprogramm gelöscht.

Erklärung des Programms:

Zeile 20 Abfrage der Spriteanzahl (max. 4)
Zeile 60 Abfrage der Speicherbereiche (11,13,14,15)
Zeile 70 Abfrage der Zeilennummer ab der die Datazeilen gespeichert werden sollen (größer als 1000)
Zeile 100 Ausgabe des Zeichenfeldes
Klammeraffe: Löschen des Spritepunktes
F1 übernimmt fertig gezeichnete Bytes in DATA-Zeilen
Zeile 200 Sprites zeichnen und berechnen des Spritepunktes
Zeile 400 Sprites übernehmen
Zeile 650 Unterprogramm Punkte setzen/löschen
Zeile 690 Löschmodul

Einschränkend sei vermerkt, daß maximal 4 verschiedene Sprites erstellt werden können, da mehr als 4 Sprites nicht problemlos abgespeichert werden können. Es dürfte aller-

dings kein Problem sein, das Programm entsprechend abzuändern. Außerdem können die Sprites nicht in Multicolor erstellt werden.

Rolf Werdehausen

```
10 PRINT"ERSTELLEN VON SPRITES!"
20 PRINT"WIEVIEL SPRITES":INPUTSZ$
30 IF VAL(SZ$)<=0ORVAL(SZ$)>4THEN10
40 SZ=INT(VAL(SZ$)):PRINT"MIN WELCHE SPEICHERPLAETZE SOLL GEPOKT WERDEN (<";
50 PRINT"FREI SIND: 11, 13,14 UND 15)?
60 PRINT:FORI=1TOSZ:PRINT"SPRITES NR" I"█":INPUTSB:SB(I)=SB*64:NEXT
70 PRINT"DATA ZEILEN NR.: (>1000) ██████████":INPUTDA$
80 IFVAL(DA$)<=1000ORVAL(DA$)>65000THENPRINT"TT":GOTO70
90 DA=INT(VAL(DA$))
100 DIMBS(21,3):I=0
110 POKE646,14:POKE198,0:I=I+1:IFI>SZTHEN690
120 PRINT"SPRITES NR: " I"█"
130 PRINT:FORZ=1TO21:PRINTTAB(11)ZTAB(15)"*****":NEXT
140 PRINT"ZEICHNEN
150 PRINT"CURSOR/SPACE
160 PRINT"██ LOESCHEN
170 PRINT"███<0>
180 PRINT"███ ENDE
190 PRINT"███<F1>█
200 SP=1:S=1:ZZ=1:BS=7
210 PO=1024+SP+14+(ZZ+2)*40:IFPEEK(PO)=160THENPE=1
220 ZE=170:FA=7:GOSUB650:GETT$
230 ZE=42:FA=14:GOSUB650:IFPE=1THENZE=160:FA=7:PE=0:GOSUB650
240 IFT$=""THEN210
250 REM SPRITES ZEICHNEN
270 IFT$="█"ORT$="" THEN340
280 IFT$="██"THEN360
290 IFT$="███"THEN430
300 IFT$="███"THEN450
310 IFT$="███"THEN470
320 IFASC(T$)=133THEN490
330 GOTO210
```

64' VideoChips

```
340 IFT$=" "THENIFPEEK(PO)<>160THENZE=160:FA=7:GOSUB650:BS(ZZ,S)=BS(ZZ,S)+2↑BS
350 IFSP=24ANDZZ<21THENZZ=ZZ+1:S=1:SP=1:BS=7:GOTO210
360 IFSP=24THEN210
370 SP=SP+1:BS=BS-1:GOTO400
380 IFSP=1THEN210
390 SP=SP-1:BS=BS+1
400 IFBS=-1THENBS=7:S=S+1:GOTO210
410 IFBS=8THENBS=0:S=S-1
420 GOTO210
430 IFZZ<>1THENZZ=ZZ-1
440 GOTO210
450 IFZZ<>21THENZZ=ZZ+1
460 GOTO210
470 IFPEEK(PO)<>42THENZE=42:FA=14:GOSUB650:BS(ZZ,S)=BS(ZZ,S)-2↑BS
480 GOTO210
490 REM SPRITES UEBERNEHMEN
500 FZ=DA:DA=FZ+INT((SZ+10)/10)*10
510 DA$=STR$(DA)+"DATA"
520 POKE646,6:PRINT"J"STR$(FZ)+"FORJ=0TO62:READQ:POKE"SB(I)+"J,Q:NEXT
530 D1=0:FORZZ=1TO21:FORS=1TO3
540 IFLEN(DA$)>70THENDA=DA+10:PRINTLEFT$(DA$,LEN(DA$)-1):DA$=STR$(DA+10)+"DATA"
550 DA$=DA$+RIGHT$(STR$(BS(ZZ,S)),LEN(STR$(BS(ZZ,S)))-1)+", "
560 NEXT:NEXT
570 IFDA$=STR$(DA+10)+"DATA"THEN540
580 PRINTLEFT$(DA$,LEN(DA$)-1):DA=DA+10
590 PRINT"90DIMBS(21,3),SB("SZ"):I="I":SZ="SZ":DA="DA+10
600 PRINT"100";:FORN=1TOSZ:PRINT"SB("N")="SB(N)">";:NEXT:PRINT"|| "
610 PRINT"500FZ="FZ+1
620 PRINT"GOTO90"
630 GOTO730
640 END
650 POKEPO,ZE:POKE54272+PO,FA:RETURN
690 ZA=10:ZE=120
700 POKE646,6:PRINT"J590ZA="ZE+10":ZE="ZE+110"
710 FORI=ZATOZESTEP10:PRINTI:NEXT:PRINT"GOTO690"
720 IFI>700THENPRINT"JPOKE646,14"
730 POKE631,19:FORI=1TO14:POKE631+I,13:NEXT:POKE198,15:END
```

IMTAL
COMPOSING
GMBH

Postfach 1207
3436 Hessisch Lichtenau
☎ Sa-Nr. 05602/4041
☒ 994026 vogt d

Fotosatz-Studio
Belichtungs-Service
Reproduktionen
Druckformherstellung
Plattenkopierstraße
Ganzseitenumbruch
über Bildschirm
Konturenprogramm
Daten-Fern-Übertragung
über Post-Modem
Weiterverarbeitung
Ihrer EDV-Daten zu Fotosatz

Disketten- schutz für Apple II (Teil 2)

Nachdem im letzten Heft ein Programm beschrieben wurde, mit dem Daten in einem nicht standardisierten Code auf Diskette aufgezeichnet wurden, wird hier nun ein Verfahren vorgestellt, das nur befugten Personen die Benutzung der Diskette erlaubt.

Das Wesentliche bei einem solchen Schutzsystem ist, daß der Benutzer keine Möglichkeit besitzen darf, in das Operationssystem (hier Applesoft-Basic mit DOS) zu kommen. Dies wird durch das Programm DOS PROTECTOR (Abb. 1) gewährleistet. Alle Abbruchbefehle wie RESET, CTRL-C, Erzwingen einer Fehlermeldung und auch IRQ und NMI werden von einem mit diesem Programm geänderten DOS mit einem „Aufhängen“ des Systems quittiert.

Um dieses Verhalten zu erreichen, werden im DOS auf Diskette einige Bytes geändert. So wird zum Beispiel der Programmteil „Fehlermeldung ausgeben“ so modifiziert, daß er die gleiche Aktion wie die RESET-Taste ausführt, nämlich den Sprung in eine Endlosschleife. Außerdem wird ein CTRL-C-Filter eingebaut. Wirkung: siehe oben.

Um nur befugtem Bedienungspersonal einen Zu-

griff auf Diskette zu gewährleisten, ist die Eingabe eines Codewortes sinnvoll. Ein konstantes Codewort allerdings schränkt dessen Funktion ein, da die Gefahr des „Abguckens“ doch relativ groß ist. Daher verlangt das Programm SPERRE, das in Abb. 2 zu sehen ist die Errechnung des einzugebenden Codezeichens. Eine 5 Zeichen lange Zeichenfolge, in der genau zwei Zahlen zwischen 0 und 4 vorkommen, wird angezeigt. Diese zwei Zahlen müssen addiert werden und das Ergebnis „geschiftet“ werden. Das heißt, die erhaltene Zahl muß zusammen mit der SHIFT-TASTE gedrückt werden. Eine Ausnahme bildet die Zahl 0: wenn sie als Summe erscheint, muß die Leertaste gedrückt werden. Als benutzerspezifischer Wert steht in Zeile 999 als DATA-Wert der Name der Diskette, der im oberen Bildschirmteil erscheint. Bei einer falschen Eingabe reagiert das Programm mit ... Aber lassen sie sich überraschen.

Das Programm SPERRE ruft seinerseits das Programm HELLO1 auf. Dieses Programm stellt ein einfaches Menüprogramm dar, das sich als Auswahl für bestimmte Programme eignet. Auch hier erscheinen die benutzerabhängigen Werte in DATA-Zeilen. In Zeile 999 wird der Name der Diskette abgelegt. Ab Zeile 1000 sind dann die einzelnen Programmnamen gespeichert. Als Zusatz zum Programmnamen wird noch eine Ziffer abgerufen, die die Art des Programmaufrufs spezifiziert. 1 steht dabei für BRUN, 2 für RUN und 3 für EXEC.

Die Programme sollen nun etwas genauer untersucht werden.

Zuerst DOS PROTECTOR: Diese Routine liest bestimmte Sektoren der Diskette ein, ändert einige Bytes und schreibt dann die geänderten Sektoren wieder auf die Diskette zurück. In den Zeilen 70, 110, 140, 170, 200 und 220 wird der einzulesende Sektor definiert und mit einem GOSUB 1040 gelesen.

Was bewirken jetzt aber die einzelnen POKES? In den Zeilen 70-100 wird der Teil des DOS geändert, der spezifiziert, welche Werte in den Speicherstellen 3D0-3FF stehen. In Zeile 90 werden diese Werte so umgestellt, daß die Speicherstellen von 3EF bis 3F1 und von 3F5 bis 3FF alle den gleichen Wert enthalten, nämlich ein JMP (\$03F2). Dieser Befehl bewirkt einen Sprung zu der Adresse, die in \$3F2 abgelegt ist. Genau das passiert aber auch bei einem RESET. Die Zeile 80 ändert den Teil des DOS um, der die RESET-Adresse er-

rechnet. Normalerweise wird nämlich in \$3F2 und \$3F3 die gleiche Adresse wie in \$3D1 und \$3D2 gespeichert. Anstelle dessen steht dort nun die absolute Adresse \$3F8. Da dort ein JMP (\$03F2) steht, führt ein Reset zur Endlosschleife.

Zeile 120 bewirkt, daß, wenn nach einem LOAD-Befehl, aber vor Rücksprung in das Basic CTRL-C gedrückt wird, dieser Tastendruck ignoriert wird. Er würde nämlich vom DOS nicht verarbeitet werden und somit auch nicht zur Endlosschleife führen. Zeile 180 ändert das Fehlermeldungsmodul, so daß ein Sprung in die Endlosschleife erfolgt.

Zeile 205 schließlich stellt den CTRL-C-Filter und den Unterprogrammteil, der CTRL-C nach einem LOAD-Befehl unterdrückt. Die Zeilen 150 und 230 sorgen dafür, daß dieser CTRL-C-Filter auch aktiviert wird.

Die Zeilen 1000-1035 stellen ein Unterprogramm dar, das den IOB (I/O-Control-Block) lokalisiert. Dieser IOB wird in den Unterprogrammen in den Zeilen 1040-1060 und 1070-1090 benutzt, um die RWTS (Read or Write a Track and Sector) zu steuern.

Nun zur Analyse des Programms SPERRE. In Zeile 1 werden zunächst zwei Zufallsfunktionen definiert. Die Funktion R1 liefert den ASCII-Wert eines zufälligen Zeichens zwischen 0 und 4, generiert also eine Zufallszahl. Die Funktion R2 hingegen generiert ein zufälliges Zeichen zwischen § und /. Zeile 4 ist sehr interessant. Hier wird nämlich in RL und RH der momentane RESET-Vector gespei-

chert. Anstatt dessen wird dort der Wert \$E199 gespeichert. An dieser Stelle steht die Applesoft-Fehlermeldung „Illegal Quantity Error“. Wichtig ist an dieser Stelle der Befehl CALL -1169. Warum, wird an späterer Stelle erläutert. In Zeile 5 steht nun der Befehl ONERR GOTO 210.

Durch die Änderung des RESET-Vektors wird also bei RESET die oben genannte Fehlermeldung ausgegeben. Diese wird aber vom ONERR GOTO-Befehl abgefangen und führt zu einem Sprung nach Zeile 210. Damit wurde also der Befehl ON RESET GOTO 210 simuliert. Die Zeilen 10-40 führen zur Ausgabe des oberen Bildschirmteils. Wichtig sind wieder die Zeilen 50-70. Diese Zeilen entstanden aus der Notwendigkeit, die Zufallsfunktion RND(1) wirklich zufällig zu machen. Denn das ist sie keinesfalls. Dieses kann mit Hilfe eines kleinen Programms leicht bewiesen werden, das als HELLO-Programm auf Diskette abgelegt wird, und Zufallszahlen erzeugt. Nach dem Kaltstart des Systems werden dort immer die gleichen Zahlen erscheinen.

Daher warten die Zeilen 50-70 des Programms SPERRE auf einen Tastendruck. So lange dieser nicht erscheint werden „Dummy-“ Zufallszahlen erzeugt.

Die Zeilen 95-180, die durch die FOR-NEXT Schleife in Zeile 90 insgesamt 5 mal durchlaufen werden, fordern den Benutzer zur Eingabe eines Codezeichens auf. Da diese Zeilen relativ komplex aufgebaut sind, würde eine Analyse hier zu weit führen und bleibt dem Leser selbst überlassen.

Die Zeilen 210-310 werden bei einer Fehleingabe durchlaufen. Das Ergebnis soll hier nicht verraten werden, sondern selbst ausprobiert werden.

In Zeile 500 wird wieder der alte RESET-Vektor zurückgeschrieben. Der CALL 1002 ist zur Reaktivierung des DOS unbedingt notwendig.

Der Befehl CALL -1169, der nach jeder Modifikation des RESET-Vektors auftaucht hat folgenden Sinn: Um unterscheiden zu können, ob der Rechner gerade einen Einschalt-Reset ausführt oder einen Lauf-Reset, wird in der Speicherstelle \$3F4 ein bestimmter Wert abgelegt, der vom Autostart-Monitor-ROM abgefragt wird. Sollte dieser Wert vorhanden sein, so wird ein JMP (\$03F2) ausgeführt, andernfalls ein Cold-Boot. Um diesen Cold-Boot zu verhindern wird der Befehl CALL -1169 verwendet, der diesen bestimmten Wert in \$3F4 schreibt. Da dieser Wert nicht konstant ist, muß er nach jeder Modifikation des RESET-Vektors neu errechnet werden. Näheres dazu findet sich in (1)-(6).

Das Programm HELLO1 ist sehr einfach aufgebaut. Auch hier findet der ONEER GOTO-Befehl Verwendung. Er wird benutzt, um das Ende der Data-Zeilen zu erkennen. In Zeile 10-55 wird der Bildschirm erstellt. Die Zeilen 60-80 geben die in den DATA-Zeilen gespeicherten Programmnamen mit einem vorausgestellten Kennbuchstaben aus. Dabei ist anzumerken, daß eine Leere DATA-Zeile eine Leerzeile auf dem Bildschirm bewirkt (siehe Zeile 70). In Zeile 99 wird der ONEER GOTO-Befehl

wieder gelöscht, und es wird die Eingabe des Kennbuchstabens vom aufzufindenden Programm verlangt. Nach dessen Überprüfung (Zeile 120) wird der entsprechende Programmname auf dem Bildschirm mit blinkender Schrift gekennzeichnet (Zeilen 140-185) und das Programm aufgerufen (Zeilen 190-210).

Eine optimal geschützte Diskette würde also das Bootprogramm das Programm HELLO von der SYSTEM MASTER-Diskette enthalten. Dieses Programm würde dann das Programm SPERRE aufrufen. Von dort aus wird dann das Menüprogramm HELLO1 aktiviert.

Wichtig: Aus den im ersten Absatz geschilderten Bedingungen darf natürlich kein Programm mit dem END-Befehl aufhören. Dieser Befehl bewirkt nämlich genau das, was mit dem Schutzsystem verhindert werden soll: den Ausstieg in die Kommandoebene des BASIC. Der Befehl END sollte ersetzt werden durch CALL 1016. Damit wird ein Sprung in die Endlosschleife erreicht.

Der Autor möchte noch erwähnen, daß dieses System keineswegs absolut knacksicher ist. Es ist aber eine einfache Möglichkeit, nur befugten Personal Zugriff zu bestimmten Disketten zu gewähren. Das typische Vorgehen beim Erstellen einer geschützten Diskette ist wie folgt:

Zuerst, wie im DOS-Manual beschrieben, das Standard-Hello-Programm von der System-Master-Diskette so modifizieren, daß das Programm SPERRE aufgerufen wird, nachdem die Language-Card

geladen wurde. Der verbleibende END-Befehl ist zu ersetzen durch CALL 1016. Mit diesem Programm als Boot-Programm die Disketten nun initialisieren. Jetzt die Programme SPERRE und HELLO1 auf die neu initialisierte Diskette kopieren. Dabei jeweils den in Zeile 999 abgelegten Diskettennamen ändern. Im Programm HELLO1 sind daraufhin in die DATA-Zeilen ab 1000 die wählbaren Programme einzutragen (Bitte nicht die Codeziffer für den Aufruftyp vergessen). Eine leere DATA-Zeile bewirkt dabei eine Leerzeile auf dem Bildschirm. Als letztes werden die Programme auf die Diskette kopiert.

Bei Basic-Programmen ist dabei jeder END-Befehl durch CALL 1016 zu ersetzen. Bei Maschinenprogrammen ist das Vorgehen wesentlich komplizierter. Diese Beschreibung würde den Rahmen dieses Artikels sprengen.

Damit ist die Diskette bootfertig. Um sie nun zu schützen, muß zuerst das Programm DOS PROTECTOR aufgerufen werden, mit dem das DOS modifiziert wird. Danach kann das Programm SPECIAL DOS aktiviert werden, mit dem die Daten im Spezialformat aufgezeichnet werden. Damit ist die Diskette geschützt und fertig für die Verwendung.

Vorschläge und Anregungen zu diesen Programmen richten Sie bitte an die Redaktion.

Jörg Bliesener

```

10 TEXT : HOME :A$ = "
O S P R O T E C T O R
20 VTAB 12: HTAB 1: PRINT "DRIVE ? (1/2) ->";
30 GET A$: IF A$ < > "1" AND A$ < > "2" THEN 30
40 DR = VAL (A$)
50 GOSUB 1000
60 VTAB 12: HTAB 1: PRINT "DOS PROTECTION RUNNING..."
70 T = 0: S = 13: GOSUB 1040
80 POKE 560,169: POKE 561,3: POKE 562,234: POKE 571,169: POKE 572,248: POKE 573,234
90 POKE 625,248: POKE 626,3: POKE 630,108: POKE 631,242: POKE 632,3: POKE 633,108: POKE 634,242
: POKE 635,3: POKE 636,108: POKE 637,242: POKE 638,3: POKE 639,242: POKE 640,3
100 GOSUB 1070
110 T = 1: S = 3: GOSUB 1040
120 POKE 589,76: POKE 590,105: POKE 591,186
130 GOSUB 1070
140 T = 0: S = 0: GOSUB 1040
150 POKE 575,32: POKE 576,208: POKE 577,8: POKE 720,169: POKE 721,111: POKE 722,133: POKE 723,5
6: POKE 724,169: POKE 725,186: POKE 726,133: POKE 727,57: POKE 728,96
160 GOSUB 1070
170 T = 1: S = 5: GOSUB 1040
180 POKE 725,108: POKE 726,242: POKE 727,3
190 GOSUB 1070
200 T = 0: S = 4: GOSUB 1040
205 FOR I = 617 TO 633: READ A: POKE I,A: NEXT
210 GOSUB 1070
220 T = 0: S = 1: GOSUB 1040
230 POKE 580,32: POKE 581,208: POKE 582,8: GOSUB 1070
240 VTAB 12: HTAB 1: PRINT "DOS PROTECTION COMPLETED."
250 END
1000 DATA 32,227,3,133,7,132,6,96,165,7,164,6,32,217,3,96
1010 FOR I = 768 TO 783: READ J: POKE I,J: NEXT
1020 CALL 768
1030 IOB = PEEK (6) + 256 * PEEK (7)
1035 RETURN
1040 POKE IOB + 2,DR: POKE IOB + 3,0: POKE IOB + 4,T: POKE IOB + 5,S: POKE IOB + 8,0: POKE IOB
+ 9,2: POKE IOB + 12,1: POKE IOB + 13,0
1050 CALL 776
1060 RETURN
1070 POKE IOB + 2,DR: POKE IOB + 3,0: POKE IOB + 4,T: POKE IOB + 5,S: POKE IOB + 8,0: POKE IOB
+ 9,2: POKE IOB + 12,2: POKE IOB + 13,0
1080 CALL 776
1090 RETURN
10000 DATA 173,16,192,108,96,157,32,27,253,201,131,208,3,76,16,164,96

```

Abb. 1: DOS PROTECTOR

```

1 DEF FN R1(X) = INT ( RND (1) * 5) + 48: DEF FN R2(X) = INT ( RND (1) * 48) + 64
4 RL = PEEK (1010):RH = PEEK (1011): POKE 1010,153: POKE 1011,225: CALL - 1169
5 ONERR GOTO 210
10 HOME : FOR I = 1 TO 40: A$ = A$ + " ": NEXT
30 READ N$
40 INVERSE : PRINT A$:A$: SPC ( 20 - LEN (N$) / 2):N$: SPC ( 40 - POS (0)):A$:A$: NORMAL
50 PRINT : PRINT : VTAB 14: HTAB 15: PRINT "PRESS ANY KEY"
60 POKE - 16368,0
70 IF PEEK (- 16384) < 128 THEN X = RND (1): GOTO 70
80 POKE - 16368,0: VTAB 6: CALL - 958
90 FOR I = 1 TO 5: VTAB 12: HTAB 18
95 X(1) = FN R1(3):X(2) = FN R1(4):C$(1) = CHR$( FN R2(3)):C$(2) = CHR$( FN R2(12)):C$(3)
= CHR$( FN R2(2))
100 P(1) = INT ( RND (1) * 4) + 1
110 P(2) = FN R1(34) - 47: IF P(2) < = P(1) THEN 110
120 CX = 1:CC = 1
130 FOR J = 1 TO 5
140 IF J = P(CX) THEN PRINT CHR$( X(CX)):CX = CX + 1

```

Abb. 3: HELLO1

```

150 IF J < > P(CX) THEN PRINT C$(CX):CX = CX + 1
160 NEXT
170 X$ = CHR$( X(1) + X(2) - 64)
180 VTAB 15: HTAB 3: PRINT "CODE CHARACTER ->": POKE - 16368,0: WAIT - 16384,128: GET C$
190 IF C$ = X$ THEN NEXT
200 IF I = 6 THEN 500
210 RESTORE : READ N$: FOR J = 778 TO 798: READ I: POKE J,I: NEXT
230 FOR J = 1 TO 4: READ I: POKE 776,I: CALL 777,I: CALL 778: NEXT
270 HOME : FLASH : PRINT A$:A$: SPC ( 17):"+++++": SPC ( 35):"+++++": SPC ( 35):"+++++": SPC ( 25)"*+
+++++: SPC ( 8)
280 S$ = "
+++++
290 PRINT A$: FOR W = 1 TO 5000: NEXT
300 HOME : NDRMAL : VTAB 12: PRINT " E V E R Y T H I N G I S O V E R
310 POKE 1010,0: POKE 1011,3: CALL - 1169: POKE 768,76: POKE 769,0: POKE 770,3: CALL 768
500 POKE 1010,RL: POKE 1011,RH: CALL - 1169: CALL 1002: PRINT : PRINT CHR$( 4)"RUN HELLO1"
999 DATA "GAMES III"
1000 DATA 173,48,192,136,208,5,206,9,3,240,9,202,208,245,174,8,3,76,10
1005 DATA 3,96
1010 DATA 200,150,200,150,200,150,240,255

```

Abb. 2: SPERRE

```

10 HOME
20 FOR I = 1 TO 40:A$ = A$ + " ": NEXT
30 READ N$: INVERSE : PRINT A$" DOS 3.3": SPC ( 29 - LEN (N$)):N$: " :A$
40 FOR I = 1 TO 20: PRINT A$: NEXT
50 VTAB 4: PRINT "WELCHES PROGRAMM ?"
55 PRINT : PRINT
60 I = 1: ONERR GOTO 99
70 READ A$: IF A$ = "" THEN PRINT : GOTO 70
73 READ C
75 PRINT CHR$( I + 64)" "A$
80 I = I + 1: GOTO 70
99 POKE 216,0
100 VTAB 22: PRINT "KENNBUCHSTABE EINGEBEN..-": HTAB 25
105 I = I - 1
110 GET A$
120 IF A$ < "A" OR A$ > CHR$( I + 64) THEN 110
130 HTAB 25: PRINT A$
135 B$ = A$
140 RESTORE : READ A$
150 FOR I = 1 TO ASC (B$) - 64
160 READ A$
165 IF A$ = "" THEN E = E + 1: GOTO 160
167 READ C
170 NEXT
180 VTAB 5 + I + E: FLASH : HTAB 5: PRINT A$
185 NORMAL
190 IF C = 1 THEN PRINT CHR$( 4)"BRUN "A$
200 IF C = 2 THEN PRINT CHR$( 4)"RUN "A$
210 IF C = 3 THEN PRINT CHR$( 4)"EXEC "A$
999 DATA GAMES III
1000 DATA GAME NO. 1,2
1020 DATA GAME NO. 2,2
1030 DATA SUPER GAME NO. 1,1
1040 DATA EXEC FILE NO. 1,3

```


Trollhöhlen

Obwohl für den Apple geschrieben, paßt das Programm in jeden Rechner. Die Befehle HOME und

VTAB bedeuten: 'Lösche Bildschirm' und 'Vertikal-Tabulator'.

```

50 HOME : VTAB 5
52 PRINT "HOEHLE NR. 1": PRINT
55 PRINT "TROLL: 'DU DARFST DIR SOVIELE KUPFER-           MUENZEN NEHMEN WIE DU WILLST.
    DAZU SCHENKE ICH DIR SCHLUESSEL           NR 1.'"
60 PRINT : INPUT "TROLL: 'ALSO WIEVIELE MUENZEN WILLST DU           NEHMEN?' ";A$
65 KM = VAL (A$):BM = 0:S = 1
100 HOME : VTAB 5
102 PRINT "HOEHLE NR. 2": PRINT
110 PRINT "TROLL: 'DU BEKOMMST SOVIELE KUPFERMUEN-           ZEN WIE DU BRONZEMUENZEN HAST
    UND UMGEKEHRT.'"
120 KM = KM + BM:BM = KM
130 GOSUB 5000
200 HOME : VTAB 5
202 PRINT "HOEHLE NR. 3": PRINT
210 PRINT "TROLL: 'WENN DU WENIGER ALS 3 MUENZEN           HAST, DANN SCHMEISSE ICH DICH IN
    EINE BODENLOSE GRUBE'"
220 IF BM + KM < 3 THEN 300
230 GOSUB 5000
290 GOTO 400
300 GOSUB 5000
310 HOME
320 END
400 HOME : VTAB 5
402 PRINT "HOEHLE NR. 4": PRINT
410 PRINT "TROLL: 'ICH NEHME MIR ZWEI BRONZEMUEN-           ZEN VON DIR!'"
420 BM = BM - 2: IF BM < 0 THEN BM = 0
430 GOSUB 5000
500 HOME : VTAB 5
502 PRINT "HOEHLE NR. 5": PRINT
510 PRINT "TROLL: 'ICH JAGE DICH ZURUECK, WENN DU           MEHR ALS EINE BRONZEMUENZE BEI
    DIR HAST.'"
520 GOSUB 5000
530 IF BM > 1 THEN 400
600 HOME : VTAB 5
602 PRINT "HOEHLE NR. 6": PRINT
610 PRINT "TROLL: 'WENN DU KEINE BRONZEMUENZEN HAST           DANN MUSST DU DIE TUER ZU DER
    DEIN SCHLUESSEL PASST OEFFNEN!'"
620 GOSUB 5000
630 IF BM > < 0 THEN 700
640 IF S < 5 THEN 800
650 HOME : VTAB 12
652 PRINT "HOEHLE NR. 7": PRINT
660 PRINT "ZAUBERER: 'WENN DU MEHR ALS ZWEI KUPFER-           MUENZEN HAST VERWANDELE ICH
    DICH IN EINEN WURM!'"
670 GOSUB 5000
680 HOME : VTAB 12: IF KM > 2 THEN PRINT "DU BIST EIN WURM.": END
690 PRINT "DU HAST ES GESCHAFFT...": END
700 HOME : VTAB 5
702 PRINT "HOEHLE NR. 9": PRINT
710 PRINT "TROLL: 'ICH TAUSCHE DEINEN SCHLUESSEL           GEGEN DEN NAECHST HOEHEREN.
    WENN DU MIR SCHLUESSEL NR.5           GIBST, DANN FRESSE ICH DICH.'"
720 IF S < 5 THEN S = S + 1: GOSUB 5000: GOTO 900
730 GOSUB 5000
740 PRINT "TROLL: 'ICH HABE DICH VIERMAL GEWARNT!'"
750 END
800 HOME : VTAB 5
802 PRINT "HOEHLE NR. 8": PRINT
810 PRINT "DU BIST IN EINE HOEHLE VON MENSCHENFRES-           SENDEN BANDWUERMERN GERATEN UND WIRST GERA

```

Apple-Kiste

```

DE VERSPEIST.": END
900 HOME : VTAB 5
902 PRINT "HOEHLE NR. 10": PRINT
910 PRINT "TROLL: 'ICH TAUSCHE DEINE BRONZEMUENZEN          GEGEN KUPFERMUENZEN.'"
920 KM = KM + BM:BM = 0
930 GOSUB 5000
1000 HOME : VTAB 5
1002 PRINT "HOEHLE NR. 11": PRINT
1010 PRINT "TROLL: 'ICH GEBE DIR EINE BRONZEMUENZE.'"
1020 BM = BM + 1
1030 GOSUB 5000
1100 HOME : VTAB 5
1102 PRINT "HOEHLE NR. 12": PRINT
1110 PRINT "TROLL: 'ICH NEHME MIR ZWEI KUPFERMUEN-          ZEN.'"
1120 KM = KM - 2: IF KM < 0 THEN KM = 0
1130 GOSUB 5000
1200 HOME : VTAB 5
1202 PRINT "HOEHLE NR. 13": PRINT
1210 PRINT "TROLL: 'DU DARFST NACH OBEN, WENN DU          KEINE KUPFERMUENZEN MEHR HAST.
ANDERNFALLS GEHT ES DURCH DEN          ZWEITEN AUSGANG ZURUECK!'"
1230 GOSUB 5000
1240 IF KM = 0 THEN 100
1250 GOTO 1000
4999 END
5000 PRINT : PRINT "KUPFERMUENZEN: ";KM
5010 PRINT "BRONZEMUENZEN: ";BM
5020 PRINT "SCHLUESSEL NR.: ";S
5030 PRINT : PRINT "DRUECKE EINE TASTE. ";: GET A$: RETURN
    
```



EXPLORING SPECTRUM BASIC £4.95 An ideal complement to the Sinclair manual. Over 50 games, application & utility programs plus explanations of programming techniques. Mike Lord. 192 pages.

THE EXPLORERS GUIDE TO THE ZX81 £4.95 The book for the ZX81 enthusiast, now in edition 2 with 152 pages of games, engineering & utility programs plus much useful information on machine language and hardware. Mike Lord.

WAKE UP YOUR ATOM £4.95 20 great programs to make the most of your Atom; including the colour board (old or new type) if fitted. Plus copious programming tips. Brian Lloyd.

THE ATOM MAGIC BOOK £5.50 A wealth of games and other programs plus much useful software and hardware information. Mike Lord.

Also available:

MASTERING MACHINE CODE ON YOUR ZX81 Toni Baker 180 pages £7.50

GETTING ACQUAINTED WITH YOUR ZX81 Tim Hartnell £4.95

GETTING ACQUAINTED WITH YOUR ACORN ATOM Tim Hartnell & Trevor Sharples £7.95

PRACTICAL PROGRAMS FOR THE BBC COMPUTER & ACORN ATOM D. Johnson-Davies £5.95

WHAT CAN I DO WITH 1K? Roger Valentine £4.95

ATOM RAM BOARDS S.A.E. for details.

ATOM ROM BOARDS £35.00 inclusive Software switch between 3 utility ROMs and 4K RAM (fitted) to load your own utilities from tape or disc.

All prices include U.K. P&P and VAT where applicable. Overseas customers add £1.50 per item for surface mail.

TIMEDATA Ltd. Dept HC 5
16 Hemmells, Laindon, Basildon, Essex Tel: (0268) 418121



TIMEDATA

Computerhirn = Mastermind

Für den CBM stellt Homecomputer heute 2 Mastermind-Versionen vor. In der einen erhält der Spieler die Option, selbst den versteckten Code zu finden, oder dem Computer die Aufgabe zu stellen! In der anderen spielen zwei Spieler gleichzeitig gegeneinander und jeder kann das Ergebnis des Gegners mit auswerten, um die Lösung zu finden.

Mastermind 1

Nachdem man das Mastermindprogramm eingetastet hat, darf man nicht gleich mit RUN starten! Zuerst wird RUN 2000 ein-

gegeben. Dabei wird erst einmal überprüft, ob man die Zeilen 0 bis 5 exakt eingegeben hat. Das ist nötig, da dort das Maschinenpro-

gramm hingePOKEd wird. Wenn alles in Ordnung ist, dann kann man das Programm starten.

Wenn der Computer die Zahl raten soll, dann wendet er folgende Strategie an:

Zuerst nimmt er Zufallszahlen als Versuch, bis er alle Ziffern einmal probiert hat; höchstens jedoch drei mal. Daß er alle in der Lösung vorkommenden Ziffern einmal probiert hat, merkt er daran, daß die Summe von allen Schwarzen und Weißen gleich Drei ist. Dabei kommt keine Ziffer doppelt vor, weil Ziffernkombinationen mit Ziffern, die schon einmal vorkamen, am Anfang nicht so informativ sind. Für den Menschen ist es vielleicht einfacher, wenn er auch schon am Anfang Ziffern wiederholt verwendet, weil er dann nicht so viel denken muß. Den Computer stört das nicht.

Falls unter den ersten drei Versuchen keine Ziffer doppelt vorkommt, weiß der Computer nach drei Versuchen, wenn man ihm noch nicht drei Schwarze

und Weiße gegeben hat, daß die noch nicht probierte Ziffer in der Lösung vorkommt.

Dann wird erst das Maschinenprogramm benutzt. Und zwar wendet es folgende Strategie an: Es zählt von 000 bis 999. Bei jeder Ziffernkombination geht es alle Versuche durch und berechnet bei jedem Versuch, wieviel Schwarze und Weiße es hätte erhalten müssen, wenn die angenommene Kombination die Lösung wäre.

Wenn es bei allen bisherigen Versuchen genauso viele Schwarze und Weiße berechnet, wie es auch bekommen hat, so verwendet es die laufende Ziffernkombination als nächsten Versuch, d.h., daß es als jeweils nächsten Versuch die kleinste Zahl nimmt, bei der die Annahme, daß es sich um die Lösung handelt, zu keinem Widerspruch zu den bisherigen Versuchen führt.

Die Versuche werden nicht in dimensionierten Variablen, sondern in den zweiten Kassettenbuffer abgelegt.

```

0 REM COPYRIGHT BY CORNELIUS GEPPERT
1 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
2 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
3 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
4 REM HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER-HOMECOMPUTER
5 REMCOMPUTER
70 GOTO300
75 R%=10*RAND(1):IFZ(R%)THEN75
77 Z(R%)=1:RETURN
80 POKE285,N:SYS1068
100 A%=PEEK(269):B%=PEEK(268):C%=PEEK(267)
120 IFPEEK(271)=0THENPRINT:PRINTTAB(7)"ERRARE HUMANUM EST":GOTO1250
240 N=N+5
250 IFETHEN1140
260 GOTO430
280 END
300 PRINT"☐ ☐ ☐ HOMECOMPUTER - M A S T E R - M I N D "
305 PRINT"☐☐☐ WOLLEN SIE DIE ZAHL RATEN? ";
310 GETB$:IFB$=""THEN310
312 GOSUB1400:A$=B$
320 PRINT:PRINT:IFA$="N"THENPRINT" SOLL DER COMPUTER GEGEN":PRINT:PRINT" SICH
SELBER SPIELEN? ";
330 IFA$="N"THENWAIT158,1:GETB$:GOSUB1400
338 N=0:POKE280,1
340 FORM=267TO270:POKEM,N:NEXT
342 FORW=1TO1E3:NEXT:PRINT"☐"
345 IFA$="J"THENE=0:GOTO1020
347 IFB$="J"THENE=1:GOTO1020

```

```

360 GOSUB75:AX=RX:GOSUB75:BX=RX:GOSUB75:CX=RX
425 IFE=1THEN1140
427 IFN=0THENPRINT"NOCHVERSUCH?"
430 PRINT:PRINTAX;BX;CX
440 PRINT:INPUT"WIEVIELE SCHWARZE UND WEISSE ";S;W:SU=SU+S+W
445 IFS+W>3THENPRINT"MAN MUESSTE MASTER-MIND SPIELEN KOENNEN!":GOTO440
447 POKE896+N,AX:POKE897+N,BX:POKE898+N,CX:POKE899+N,S:POKE900+N,W
450 GOSUB1300
480 IFSUC3ANDN<10THENN=N+5:GOTO360
490 IFS<3THEN80
500 PRINT:PRINT"ICH HABE NUR"N/5+1"VERSUCH"LEFT$( "E",1ANDN>0)" GEBRAUCHT":GOTO12
50
1020 A=INT(10*RND(1)):B=INT(10*RND(1)):C=INT(10*RND(1))
1030 IFE=1THEN360
1040 INPUT"WIE LAUTET DER ERSTE VERSUCH "V:GOTO1110
1050 GOSUB1300
1070 IFE=0THENN=N+5
1075 IFSUC3ANDN<10ANDETHENN=N+5:GOTO360
1080 IFS<3ANDETHEN80
1085 IFETHEN500
1086 IFS<3THEN1090
1087 PRINT:PRINT"SIE HABEN NUR"N/5+1"VERSUCH"LEFT$( "E",1ANDN>0)" GEBRAUCHT":GOTO
1250
1090 INPUT"WIE LAUTET DER NAECHSTE VERSUCH"V
1110 AX=V/100+1E-4
1120 BX=10*(V/100-INT(V/100)+1E-3)
1130 CX=10*(V/10-INT(V/10)+1E-3)
1140 S1=1:S2=S1:S3=S1:S4=S1:S5=S1:S6=S1:S=0:W=S
1150 IFA=AXTHENS=S+1:S1=0:S2=S1
1160 IFB=BTHENS=S+1:S3=0:S4=S3
1170 IFC=CTHENS=S+1:S5=0:S6=S5
1180 IFAX=BXANDS2ANDS3THENW=W+1:S2=0:S3=S2
1190 IFAX=CXANDS2ANDS5THENW=W+1:S5=0
1200 IFA=BXANDS1ANDS4THENW=W+1:S1=0:S4=S1
1210 IFB=CXANDS4ANDS5THENW=W+1
1220 IFA=CXANDS1ANDS6THENW=W+1:S6=0
1230 IFCX=BXANDS3ANDS6THENW=W+1
1235 POKE896+N,AX:POKE897+N,BX:POKE898+N,CX:POKE899+N,S:POKE900+N,W:SU=SU+S+W
1240 GOTO1050
1250 PRINT"NOCHMAL SPIELEN (LASSEN) ?
1260 WAIT158,1:GETB$:IFB$="J"THENRUN
1270 PRINT"SCHADE !"
1280 PRINT:PRINT" AUF WIEDERSEHEN"
1290 END
1300 IFEANDN>0THENFORN1=N1+5TON1STEP5:GOTO1330
1305 INPUT:PRINTTAB(22);"ANZAHL DER
1310 PRINT" VERSUCH"SPC(9)"SCHWARZEN UND WEISSEN
1320 FORN1=896TON+N1STEP5
1330 PRINTPEEK(N1)PEEK(1+N1)PEEK(2+N1),,PEEK(3+N1),PEEK(4+N1):NEXT:N1=N1+5
1340 RETURN
1400 PRINTLEFT$( "JA",2ANDB$="J")LEFT$( "NEIN",4ANDB$="N"):RETURN
2000 FORI=0TO5:READA:IFPEEK(A)ORPEEK(A+4)THENPRINT"DIE ZEILE"ISTIMMT NICHT.":EN
D
2010 NEXT
5000 DATA1061,1132,1203,1274,1345,1359
5010 FORI=1062TO1359:READA:POKEI,A:NEXT
5020 PRINT"JETZT KOENNEN DIE ZEILEN AB 2000
5030 PRINT"GELOESCHT WERDEN.
6000 DATA129,4,1,0,143,34,141,15,1,174,41,4,169,1,160,6,153,15,1,136,208,250,140
6010 DATA22,1,140,23,1,189,128,3,77,13,1,208,9,238,22,1,140,16,1,140,17,1,189,12
9
6020 DATA3,77,12,1,208,9,238,22,1,140,18,1,140,19,1,189,130,3,77,11,1,208,9,238
6030 DATA22,1,140,20,1,140,21,1,189,131,3,77,22,1,240,11,76,26,5,0,80,5,2,0,143
6040 DATA34,66,189,128,3,77,12,1,208,17,173,17,1,45,18,1,240,9,238,23,1,140,17,1
6050 DATA140,18,1,189,128,3,77,11,1,208,14,173,17,1,45,20,1,240,6,238,23,1,140,2
0
6060 DATA1,189,129,3,77,13,1,208,17,173,16,1,45,19,1,240,9,238,23,1,140,16,1,140
6070 DATA19,1,189,129,3,77,11,1,208,11,173,19,1,45,20,1,240,3,238,23,1,189,130,3
6080 DATA77,13,1,208,14,173,16,1,45,21,1,240,6,238,23,1,140,21,1,189,130,3,77,12
6090 DATA1,208,11,173,18,1,45,21,1,240,3,238,23,1,189,132,3,77,23,1,208,6,236,29
6100 DATA1,208,46,96,173,11,1,201,9,240,6,238,11,1,76,47,4,205,12,1,240,9,238,12
6110 DATA1,152,141,11,1,240,239,205,13,1,240,9,238,13,1,152,141,12,1,240,237,140
6120 DATA15,1,96,232,232,232,232,232,76,50,4,0

```

Mastermind 2

Bei dieser Version spielen zwei Spieler gegeneinander und versuchen zur gleichen Zeit die versteckte Zeichenkombination zu erraten, wobei jeder die

Versuche des Gegners und die eigenen in seine Strategie zur Lösungsfindung mit einbeziehen kann. Die Spielanleitung am Anfang erklärt, wie das Spiel bedient wird. Hat man sei-

nen Versuch auf dem Bildschirm mittels Zahlen und den 'Größer-als- Kleiner-als-Tasten' plazierte, so schließt man ihn mit der Spacetaste ab.

```

1 H=2:GOSUB1000
5 INPUT"HWIEVIEL ZEICHEN ( 3 - 10 )";WZ:WZ=INT(WZ):IFWZ>10ORWZ<3THEN5
10 DIMTR$(2,12,5),RP(2,12),RZ(2,12),CZ$(5),P$(14)
20 FORI=1TO5:CZ(I)=INT(RND(1)*WZ):NEXT
21 FORI=1TO5:IFCZ(I)=0THENCZ$(I)="#"
22 IFCZ(I)=1THENCZ$(I)="@"
23 IFCZ(I)=2THENCZ$(I)="A"
24 IFCZ(I)=3THENCZ$(I)="*"
25 IFCZ(I)=4THENCZ$(I)="^"
26 IFCZ(I)=5THENCZ$(I)="<"
27 IFCZ(I)=6THENCZ$(I)=">"
28 IFCZ(I)=7THENCZ$(I)="="
29 IFCZ(I)=8THENCZ$(I)=">"
30 IFCZ(I)=9THENCZ$(I)="<"
31 NEXT
32 FORI=1TO14:READP$(I):NEXT
35 PRINT" "
50 F=0:IFH=1THENH=2:GOTO70
60 IFH=2THENH=1
62 DATA" ", " ", " ", "0 = #", "1 = @", "2 = A", "3 = *", "4 = ^", "5 = <", "6 = >"
63 DATA"7 = =", "8 = >", "9 = <"
64 DATA" ", " ", " "
70 PRINT" ** MASTERMIND ** "
75 IFV<1THENFORI=1TO14:PRINT" ";P$(I);" ";NEXT
80 PRINT" "
82 PRINT" RETURN "
84 PRINT" "
95 PRINT" "
100 FORI=1TO12:FORJ=1TO5
110 PRINT" ";LEFT$(TR$(H,I,J),1);IFTR$(H,I,J)=" "THENPRINT" ";";
115 NEXT
120 PRINT" ";IFRP(H,I)=0THENPRINT" ";";LF=5
121 IFRP(H,I)=1THENPRINT" ";";LF=4
122 IFRP(H,I)=2THENPRINT" ";";LF=3
123 IFRP(H,I)=3THENPRINT" ";";LF=2
124 IFRP(H,I)=4THENPRINT" ";";LF=1
125 IFRP(H,I)=5THENPRINT" ";";
126 FORI1=1TOLF:PRINT" ";NEXT
130 IFRZ(H,I)=1THENPRINT" ";";
131 IFRZ(H,I)=2THENPRINT" ";";
132 IFRZ(H,I)=3THENPRINT" ";";
133 IFRZ(H,I)=4THENPRINT" ";";
134 IFRZ(H,I)=5THENPRINT" ";";
140 PRINT
150 NEXT:IFF=1THEN50
200 F=1:L=L+1:V=INT(L/2.000001)+1
210 PRINT" SPIELER";H;" "
215 FORI2=0TO4:POKE32810+V*40+I2*3,191:NEXT
216 GOSUB600
221 GOSUB501
225 FORI=1TO5:CZ$(I)=LEFT$(CZ$(I),1):NEXT
230 FORI=1TO5:IFTR$(H,V,I)=CZ$(I)THENRP(H,V)=RP(H,V)+1:GOSUB900
235 NEXT
240 FORI=1TO5:FORJ=1TO5:IFTR$(H,V,I)=CZ$(J)THENRZ(H,V)=RZ(H,V)+1:GOSUB910
245 NEXTJ,I
250 IFRP(H,V)=5THEN300
260 IFV=12ANDH=2THEN400
270 GOTO50
300 PRINT" SPIELER";H;" GEWONNEN IM";V;" VERSUCH ";
310 GOTO430
400 PRINT" "
401 PRINT" ALLE SPIELER SIND NACH 12 VERSUCHEN GESCHEITERT!";
402 PRINT" "

```


Der VC-64 Videochip Teil 1

In diesem Heft beginnt eine Serie, in der dem Computer etwas näher in die Bytes geschaut werden soll.

Die maximale Anzahl der Zeilen im Tastaturpuffer läßt sich durch POKE 649, N bestimmen, wobei N eine Integerzahl zwischen 0 und 255 ist. Tippt man noch mehr Tastendrucke ein bevor die nächste GET Anweisung kommt, so werden die weiteren Tastendrucke ignoriert. Es ist jedoch nicht empfehlenswert, in 649 eine größere Zahl als 10 zu poken, da der 11te abgespeicherte Tastendruck den Computer aussteigen läßt. Man kann das Basic umändern und sich sozusagen

sein eigenes Basic schreiben. Dazu muß man erst das Basic aus dem ROM in den RAM-Bereich gleicher Adresse kopieren. Das macht man mit:
FOR I = 40960 TO 49151:
POKE I, PEEK (I): NEXT I.
Jetzt muß man dem Computer noch sagen, daß das Basic im RAM steht. Das tut man mit POKE 1,54. Zurückschalten kann man mit POKE 1,55.
Nun kann man nachsehen, an welcher Stelle die einzelnen Befehlswoorte stehen; z. B. indem man die STOP-Taste drückt, wenn

das gesuchte Wort kommt und dann die Variable I drückt. Dann PEEKed man die Bytes in der Nähe von I ab, um den genauen Wert zu bekommen. Per POKE kann man jetzt die Befehlswoorte umändern. Man muß aber darauf achten, daß man bei dem letzten Buchstaben oder Zeichen das höchstwertige Bit setzt, d. h. zu dem ASCII-Code 128 addiert, und bei den anderen nicht, da sich sonst die Interpretercodes der darauffolgenden Befehle und Funktionen ändern. So kann man z. B. mit POKE 41118, ASC ("U") den Befehl END in UND umändern. Dann kann man schreiben: END = 7 : PRINTEND und erhält die 7. Dagegen führt die Anweisung END zu einem ?SYNTAX ERROR. Am Programmende kann man UND statt END schreiben.

Die Befehlswoorte stehen ab 41118 bis 41372 in der Reihenfolge ihres Interpreter-

tercodes. Der erste Befehl, END, hat den Interpretercode 128. Das Ende eines Befehlswoortes ist dadurch gekennzeichnet, daß das höchstwertige Bit gesetzt ist.

Wenn man die Befehlswoorte auslisten will, muß man das höchstwertige Bit löschen, da sonst statt des letzten Buchstabens des Befehlswoortes das Graphikzeichen erscheint, das man erhält, wenn man den letzten Buchstaben mit gedrückter Shift-Taste tippt. Hier ein Programm dazu:

```
10 FOR I = 41118 TO 41372
20 P = PEEK(I):
PRINTCHR$(127 AND
P); IF P > 127 THEN
PRINT
30 NEXT
```

Im nächsten Heft werden wir uns dann näher mit dem VC-64-Zeichensatz und seine Möglichkeiten befassen.

Homecomputer

**Sonderheft
Hannover Messe**



**Hannover
Messe '83**

ab 20. Mai '83

bei Ihrem Zeitschriftenhändler

oder direkt bestellen beim:

Roeske Verlag

Postfach 629

3440 Eschwege

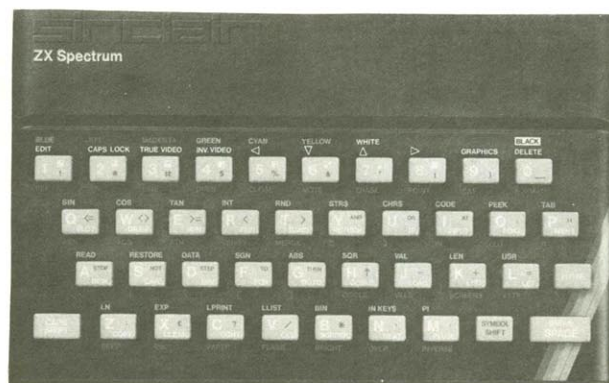
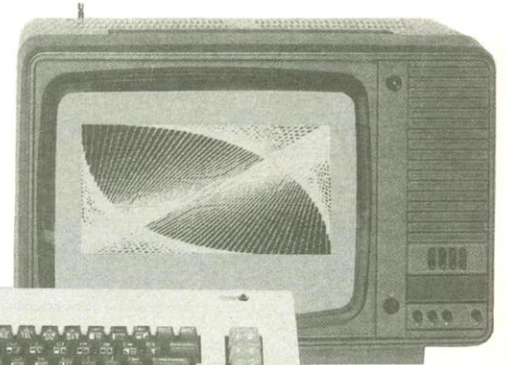
250 Computer zu gewinnen.

Jeder Abonnent, der einen Abonnenten für Homecomputer wirbt, erhält als kleines Dankeschön "Das Microcomputer-Taschenbuch" (Erscheinungstermin Juni 1983) mit viel Wissenswertem über die populärsten Microcomputer. Außerdem verteilen wir noch 250 Computer an diejenigen Leser, die Homecomputer die meisten Abonnenten zuführen.

Die Gewinnverteilung erfolgt nach einem für alle Leser sehr interessanten Schlüssel. Jedesmal, wenn bei uns 1000 neue Abonnenten registriert wurden, dann erhalten diejenigen 10 Leser, die zu diesem Zeitpunkt die meisten Abonnenten geworben haben – also in der Wettbewerbsliste vorne liegen – einen Microcomputer ihrer Wahl und zwar der Erste dieser Wettbewerbsliste einen Microcomputer im Werte von 2500,- DM und die 9 folgenden jeder einen im Werte von 1000,- DM.

Die 10 Gewinner werden aus der Wettbewerbsliste gestrichen (sie können allerdings von neuem beginnen und wieder gewinnen), und die ihnen Folgenden rücken nach.

Sind wieder 1000 Abonnenten erreicht, geht es weiter wie oben beschrieben.



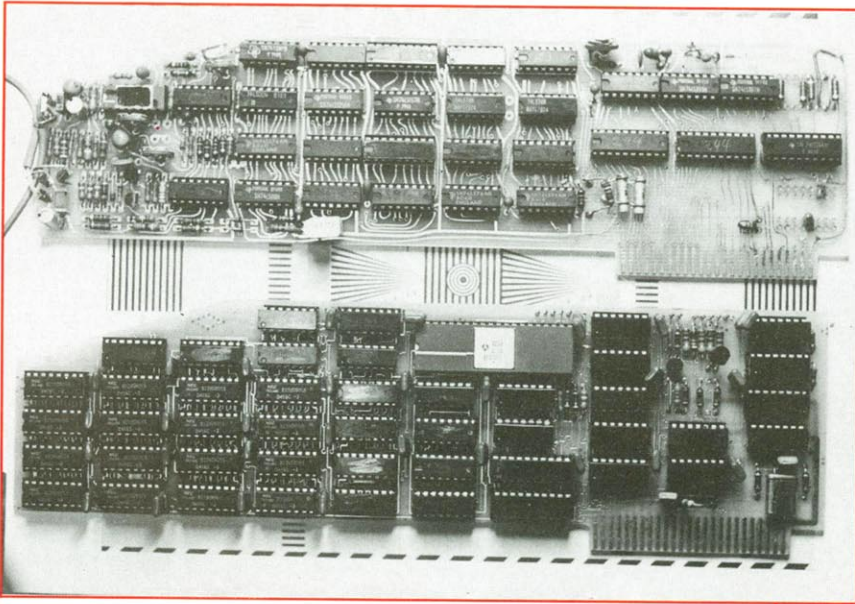
Teilnahmeberechtigt ist jeder, der Homecomputer regelmäßig im Abonnement bezieht.

Die Gewinner werden sofort benachrichtigt und in der jeweils folgenden Ausgabe von Homecomputer bekanntgegeben.

BARTSCHER ELEKTRONIK

Rüsten Sie Ihren Apple auf mit unseren Video-Interfaces!

(auch für Apple kompatible Computer)



AD 16-3

Echtzeitdigitalisierer

1356,- DM

incl. MWSt.



HGR-512

Graphicinterface

1356,- DM

incl. MWSt.

HGR-512 Graphicinterface

- Eigener 32 k Byte Bildspeicher
- Graphic-Controller EF 9365
- Umschaltbar 512 x 512 Punkte in schwarz-weiß oder 256 x 256 in 16 Graustufen bzw. 16 Farben auf jedem dieser Punkte
- Hardcopymöglichkeit von 512er-Modus
- ASCII-Zeichensatz in Größe und Richtung programmierbar
- Vektorplot mit 1 Millionen Punkte pro Sekunde
- Ausgang für RGB- und BAS-Monitore
- Demosoftware auf Diskette

AD 16-3 Echtzeitdigitalisierer

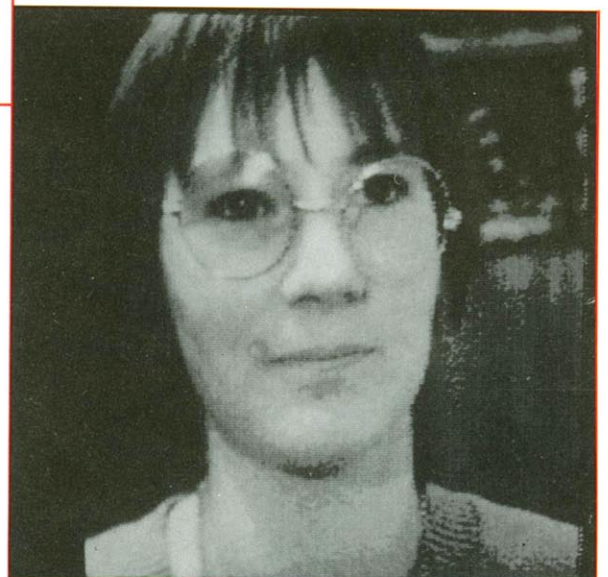
- Echtzeitdigitalisierer in 40 m sec.
- ca. 65 000 Bildpunkte mit 16 Graustufen
- eingebauter DA-Wandler zur Betrachtung des digitalisierten Bildes
- Digitalisierung von Videorecorder, Kamera oder Fernseher
- Keine Software für Bildlesevorgang nötig

NEUANKÜNDIGUNG:

AD 16-4

Echtzeitdigitizer mit
Sofortdisplay und
eigenem Bildspeicher

Digitalisierung und Display geschieht in 1/50 sec.
Karte enthält eigenen Bildspeicher.
Auflösung: 256 x 256 Bildpunkte.
Fremdsynchronisation der Kamera notwendig.



mit AD16-3 digitalisiert durch HGR-512 dargestellt

Bartscher Elektronik · Westring 84 · 3440 Eschwege · Tel.: 0 56 51/17 95

Programm-
Zusendungen
erwünscht!

Neu:

Demnächst
Software für alle
Systeme lieferbar!

Der Fluch des Pharaos

Adventure-Spiel f. VC-20

(Speichererweiterung mind. 16K)

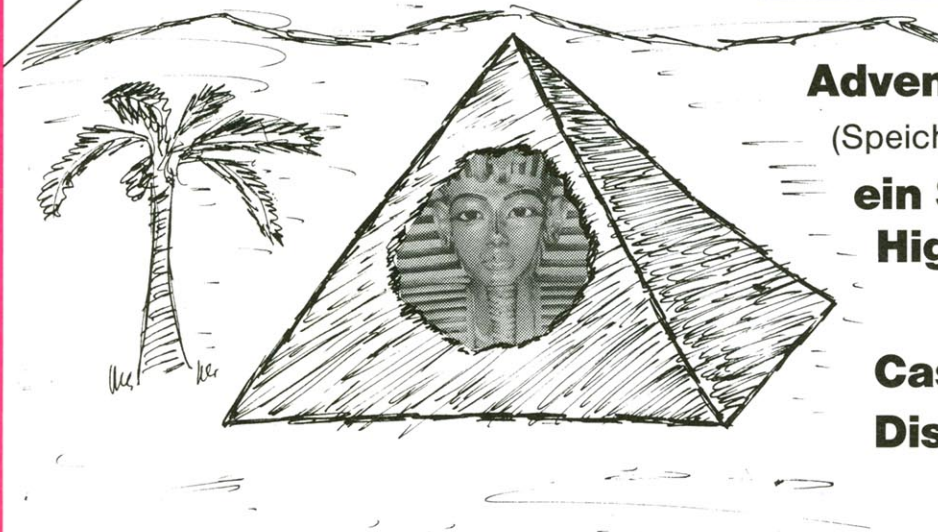
**ein Superspiel m.
High-Res.-Graphik**

Cassette

19.50 DM

Diskette

25.00 DM



5 Programme für den VC-20 auf einer Cassette!
Geschicklichkeits- und Denkspiele f. Joystick und
Tastenbedienung.

- Autorennen
- Rememby
- Roulette
- Mastermind
- Fadenkreuz

alles zusammen nur

19.50 DM

- V-Scheck ist beigelegt
 per Nachnahme
(zzgl. Nachnahmegebühr)

WICOSOFT

Nordstr. 22, 3443 Herleshausen

Ich bestelle:

Der Fluch des Pharaos
VC-Programmkassette

- Cassette 19.50 DM
 Diskette 25.00 DM
 VC-20 o.Erw.
 VC-20 +8K

Name, Anschrift: _____

