

monatlich

# CompuTe mit

## COMMODORE & SCHNEIDER

10/86

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

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

3,80 DM

33 öS

3,80 sFr

**Aktuell +++ Aktuell +++ Aktuell +++ Aktuell**

**Jetzt neu: größerer  
Schneider-Teil:  
4 Topprogramme**

**C 16:**

wie wär's mit einer Runde

**Domino ?**

**VC 20**

**Laufschrift**

**C 64**

**Sonderteil Hilfsprogramme**



**Software:  
Jungunternehmer  
auf dem Weg nach oben**

**Leser- + Meckerecke – Report – Software-Abo – Software-Service  
Bücher – Werkstatt – Assembler-Kurs – Software-Reviews – Tips & Tricks  
Großer Kleinanzeigenmarkt – Panic-Car – Airfighter – u.v.m.**



## EDITORIAL

Lieber Leser!

Können Sie sich vorstellen, wie in einer Computerredaktion gearbeitet wird? Ehrlich gesagt, auch wir, die wir täglich an dem Produkt für unserer Leser arbeiten, wissen manchmal nicht mehr, wie wir unsere Termine einhalten können, da unsere Redaktion mit einer Reihe von mehr oder weniger kniffliger Fragen unserer Leser vor allem auf telefonischem Wege konfrontiert wird. Als Leserzeitung sehen wir unsere Aufgabe auch darin, bei Problemen, z.B. mit Listings, Hilfestellung - auch telefonisch. Aber: darunter darf natürlich die normale Arbeit an den Zeitungen des Verlages natürlich nicht leiden. Deswegen haben Sie bitte Verständnis, daß die Programmierabteilung zukünftig nur noch werktags von 14

Uhr bis 16 Uhr für telefonische Anfragen zur Verfügung steht. Um eine möglichst große Zahl von Leseranfragen beantworten zu können, bitten wir Sie außerdem um folgendes: bitte wählen Sie bei Anfragen an die Programmierabteilung ausschließlich die Nummer 0561/30013. Nennen Sie bitte als erstes Ihr Computersystem und den Namen und die Ausgabennummer der Zeitung, zu der Sie Fragen haben. Auf diese Weise helfen Sie uns, Ihre Frage zu beantworten und den anderen Lesern, die wie Sie Fragen haben.



Uwe Knierim  
Chefredakteur

## Report

Nachrichten vom Zubehörmarkt .....	4
Jugend - auf dem Weg nach oben .....	4
Software-Champion '86 .....	5
Buchbesprechung .....	8
Geos intern (2) .....	11

## Tips & Tricks

Laufschrift mal anders (VC-20) .....	16
Refusal-Basiclader (C-16) .....	22
Hardcopy-Blitz (C-16) .....	23
Autostart (C-16) .....	23

## IMPRESSUM

"Compute mit"  
erscheint monatlich im  
Tronic-Verlag, Am Stad 35, 3440 Eschwege  
Tel.: 05651/3 00 11

### Redaktion:

Axel Credé (verantwortlich)  
Chefredakteur: Uwe Knierim  
Redakteure: Manfred Kleimann, Bernd Zimmermann, Frank Brall, Otfried Schmidt, Thomas Brandt

### Gesamtherstellung:

Druckhaus Dierichs Kassel, Frankfurter Str. 168,  
3500 Kassel

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

### Vertrieb

Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz:  
Verlagsunion  
Friedrich-Bergius-Str. 20  
Telefon: 06121/2660

### Erscheinungsweise:

Erstverkaufstag von "Compute mit" jeweils Mitte des Monats.

### Urheberrecht:

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

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

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

Fill-Utility (C16) .....	27
Sprite Turn/Charakter Switch (C-64) .....	44

## Checksummer

VC-20 .....	24
Steurcodes im Klartext .....	25
CPC .....	26
Version 1.0 C-64 .....	31

## Rubriken

Korrekturen .....	8
Leser & Meckerecke .....	11
Assemblerkurs .....	28
Softwareservice .....	33
Kleinanzeigen .....	60

## Software

### Commodore

Treasure Robber (VC 20) .....	13
Scrumble (VC 20) .....	15
Railway (C 16) .....	17
Domino (C 16) .....	20
Horse Games (C 64) .....	34
Patrouille (C 64) .....	41

### Schneider

Panic-Car (CPC) .....	49
Scurfy Joe (CPC) .....	51
Drawgenerator (CPC) .....	53
Airfighter (CPC) .....	56

### Bezugspreis:

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

### Autoren, Manuskripte:

Der Verlag nimmt Manuskripte zur Veröffentlichung gerne entgegen.

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

Bei Zusendung von Manuskripten und Software erteilt der Autor dem Verlag die Genehmigung zum Abdruck und Versand der veröffentlichten Programme auf Datenträger. Alle Einsendungen müssen frei von Rechten Dritter sein. Dies muß mit der Unterschrift des Einsenders bestätigt werden.

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

Kopierfähige Kassette oder Diskette mit dem Programm (Computerbezeichnung nicht vergessen!), von Drucker erstelltes Listing (keine Schreibmaschinenlistings!), evtl. Bildschirmfotos oder Hardcopies mit Demonstrationsbeispielen und ausführliche Programmbeschreibung (Erklärung der programmtechnischen Besonderheiten, Spielverlaufbeschreibung). Für eingesandte Programmunterlagen kann keinerlei Haftung übernommen werden.

### Anzeigenpreise:

Bitte Mediaunterlagen anfordern.

### Anzeigenverwaltung:

Anzeigenleiter: Hartmut Wendi

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

**Bei Ihrem Zeitschriftenhändler  
erhältlich!**

SA 3/86

65 55 sfr 6,50 DM 6,50

# Compute mit

**COMODORE**  
**C16**  
Heimcomputer

## SONDER HEFT 3

**C16/plus 4**

**Jetzt neu:  
Kleinanzeigen**

Software  
Reviews

**Hardware  
TIPS**

Whitch

Indiana  
JOE

Musik  
Compiler

Bücher

Tips  
und Tricks

Spionage-  
Lehrling

Gewinnspiel:  
Tolle Preise!

Kampf  
um Rom

**GENESIS —  
der Super-  
ASSEMBLER**

MasterMind

u. a.

**STOP + + + Neue Ausgabe + + + STOP**

# Jugend – auf dem Weg nach oben



Dieter Eckhardt & Holger Gehrman = GOLDEN GAMES GbR. Zwei Namen, die man sich für die Zukunft merken sollte. Beide garantieren: Qualitative Software made in Germany! Erst vor kurzem haben die Beiden ihren ersten Knüller gelandet. Mit OPERATION HONGKONG lieferten sie ein Programm, welches in dem führenden Software-Magazin ASM geradezu grandios bewertet wurde. Also eine echte Rarität, wenn man die deutsche Marktszene in Sachen Adventures betrachtet.

Dabei fing es allerdings gar nicht so gut an. Mit MAGIC BALL hatten sie ein Programm vermarktet, welches beinahe zu einer Katastrophe geführt hätte. Der freie Mitarbeiter hatte einen entscheidenden Fehler in seinem Programm, der die Jungs veranlaßte, das Programm im letzten Moment aus den Kaufhäusern zu holen. Die Konsequenz war eindeutig: Der gute Mann arbeitet nicht mehr für Golden Games. Das sind halt Erfahrungen, das es solche Problemchen bei einer eigenen Firma geben kann. Trotz dem haben Dieter und Holger mit frischem Mut weiter gemacht,

obwohl das größte Problem zumindest bei Holger bestehen bleibt: Die Schule. Das es nicht ganz einfach ist eine eigene Firma neben der Schule zu betreiben, dürfte auf der Hand liegen. Gelegentlich gibt es dann Phasen, wo man in Zweifel gerät oder auch die Lust verliert. Heute scheint sich das aber endgültig gelegt zu haben. Für Holger Gehrman steht mittlerweile fest, daß die Firma an erster Stelle steht, selbst wenn sein Abitur dadurch gefährdet ist. Gerade das Projekt Operation Hongkong dürfte ausschlaggebend für diese Aussage gewesen sein. Bei so einem Erfolg kann man nur auf dem richtigen Weg sein. Das Geld fließt zwar noch nicht in Strömen, jedoch sammelt sich schon einiges an. Im Vergleich zu seinen Klassenkameraden hat Holger natürlich traumhaft viel Geld zur Verfügung. Die Devise heißt trotzdem bei den Beiden: Sparen! Man will ja auch irgendwann einmal gezielt für die Firma investieren.

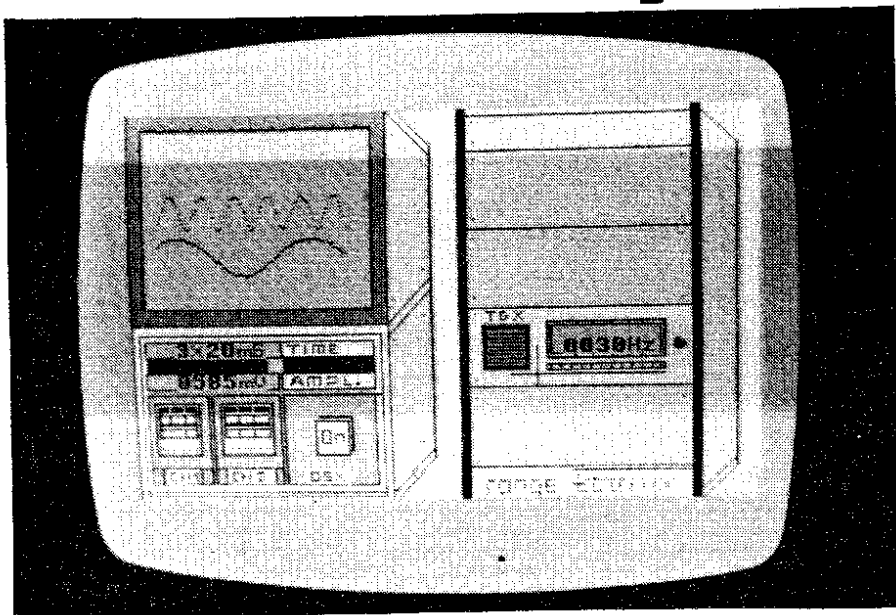
Jedenfalls macht es Dieter und Holger einen riesigen Spaß tolle Programme zu schreiben. Wenn sie sich dann auch noch gut verkaufen, gibt das immer wieder neuen Mut für weitere Programme.

Logical Systems stellt "E-Lab", ein erweiterungsfähiges Meß- und Steuersystem vor. Hobbyelektroniker und Einsteiger bekommen ein Meßsystem zur Hand, das in der Grundversion aus einem Zweikanaloszilloskop und einem Tongenerator besteht. Elektronische Schaltungen können ab sofort mit "E-Lab" ausgiebig studiert werden. An einem Elektronikübungsbuch, das spielerisch in eine komplexe Technik einführt, wird gearbeitet. Zusammen mit dem "E-Lab-System" wird eine sinnvolle und interessante Freizeit gestaltet, die fundierte Kenntnisse nebenbei vermittelt.

PREIS des Systems "E-Lab":  
Grundversion + 2 Meßköpfe....  
98,- DM  
Experimentierhandbuch ca. ...  
60,- DM

Zu beziehen per NN bei:  
Logical Systems, Postfach 72,  
6501 Nieder-Olm

## Meß- und Steuersystem



# Software-Champion 1986!

## Die besten Autoren gesucht!

**1. Preis**

**5000,- DM  
in bar**

\* Schwerpunkt des Wettbewerbs ist der C16 mit seinen Brüdern C116 und plus4, besonders die 64k-Varianten. Aber auch alle anderen Systeme werden berücksichtigt! Die Entscheidung fällt die Redaktion. Anfang 87 werden dann die Gewinner vorgestellt!

**EXTRA**

**200,- DM**

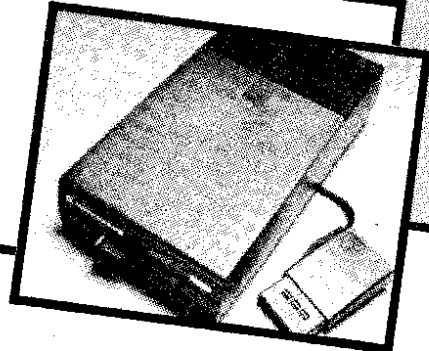
für die 5 besten Programme,  
die für die 64K-Versionen  
von C16/116  
geschrieben sind!

Einsendungen an:  
Tronic-Verlag  
Postfach  
3440 Eschwege  
Kennwort: Software-Champion '86

\* Der Tronic-Verlag sucht die besten Softwareautoren 1986! Die besten Programme werden abgedruckt und mit 120,- DM pro Seite honoriert und kommen in die engere Wahl für unseren Wettbewerb!

**2.+3. Preis**

**Floppy  
1551**  
für C16/116/  
plus 4 oder  
anderes Zubehör  
im Wert  
von 400,- DM



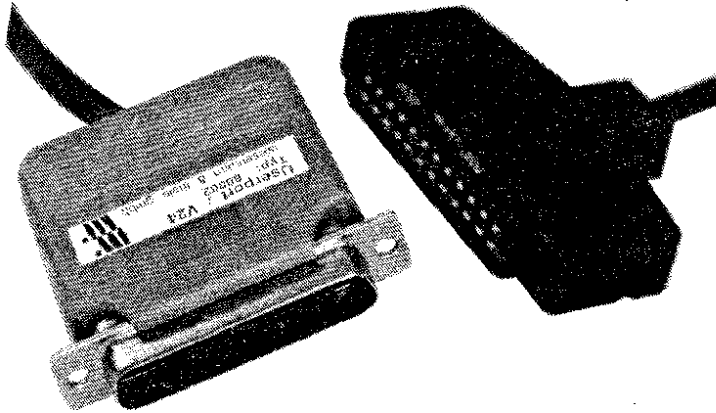
**4.-53. Preis**

**50  
Programm-  
Kassetten  
nach Wahl!**

## V 24-Adapter für C-64/128

C-64/C-128-Computer haben eine eingebaute serielle Schnittstelle, die im TTL-Pegel an den User-Port geführt ist.

schließlich Spannungswandler ist sehr kompakt im Steckergehäuse untergebracht. Eine Stromversorgung ist nicht erforderlich.

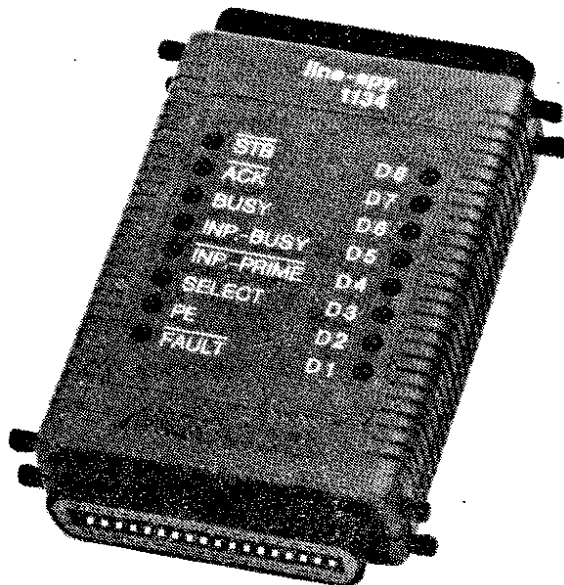


Mit Hilfe des Interfaces Type 88002 der Firma Wiesemann Microcomputertechnik, werden die TTL-Pegel in normgerechte V24-Signale gewandelt (ca. +/- 9V). Die notwendige Elektronik, ein-

PREIS: 98,- DM (inkl. MwSt. ab Wuppertal)

Wiesemann Microcomputertechnik, Winchenbachstr. 3-5, 5600 Wuppertal 2 (Barmen)

## Linespy auf Zack



Fehleranalyse in Sekundenschnelle – das leistet der Linespy 1134, der sich für alle Centronics-Systeme verwenden läßt. Die wichtigsten Leitungen werden von 16 Leuchtdioden überwacht. Besondere Impulsverzögerungsschaltungen ermöglichen die Analyse auch sehr

kurzer Signale, wie zum Beispiel den Strobe. Und: Im Gegensatz zu vielen anderen vergleichbaren Testern zeigt der Misco Linespy die Signale immer in aktivem Zustand – eine enorme Erleichterung für den Anwender.

\*\*\*

# Neuer Katalog – über 500 Artikel



Mehr als 450 Zubehörteile für Home- und Personal-Computer beschreibt der über 70 Din-A-4-Seiten starke Katalog von Lindy. Interfaces, EPROM-Programmer, Trackball, Joystick, Lichtgriffel und Maus, Stecker jeder Art, ein Gerät, das den Homecomputer in ein komfortables Speicheroszilloskop verwandelt und Software sind nur einige Beispiele aus dem Programm. Die besondere Stärke des Lieferanten liegt jedoch bei Computerkabeln. Er beschränkt sich bewußt nicht auf die gängigsten Kabeltypen, sondern versucht für jedes Problem die passende Lösung anzubieten. Die Kabel verbinden fast

jeden Computer mit jedem Monitor, Fernsehgerät, Kassettenscanner, Diskettenlaufwerk, Drucker, Akustikkoppler und Joystick.

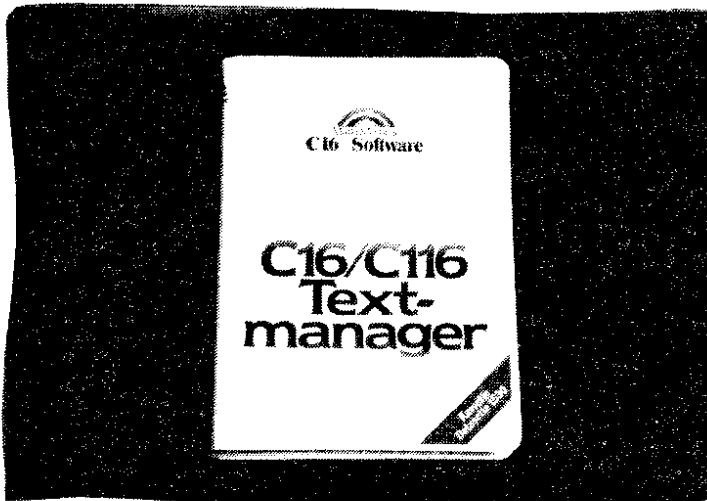
Verlängerungskabel für den IBM-Tastaturanschluß sind beispielsweise ebenso im Lager wie Verbindungskabel zwischen C-64 und Centronics-Drucker.

(Ein Programm als "Software-Interface" sorgt dafür, daß sich diese beiden "unverträglichen" Geräte verstehen.)

LINDY-Elektronik GmbH, Postfach 1428, Böckstr. 21, 6800 Mannheim 1

\*\*\*\*\*

## Jetzt endlich: Auch für den C-16 ein Super-Textprogramm!



**Programm:** Textmanager  
**System:** C16/C116  
**Preis:** Diskette 29.90 DM, Kasset-  
te 29.90 DM  
**Hersteller/Vertrieb:** Markt &  
Technik Verlag

Durch die rasche Verbreitung der C16 Rechner entstand lange Zeit eine Lücke an brauchbaren Anwenderprogrammen. Mittlerweile hat sich das Bild grundsätzlich geändert; von fast allen großen Herstellern wird nun auch der Besitzer eines C16 unterstützt. Das Programm TEXTMANAGER ist eines der ersten, wirklich leistungsfähigen Textprogramme überhaupt. Interessant ist noch dazu, daß das Programm zu einem Schleuderpreis von 30 DM angeboten wird. Damit versucht: Markt & Technik offensichtlich genau in die Zielgruppe der C16 Käufer zu treffen. Trotz dieses enorm günstigen Preises braucht sich das Programm nicht hinter Konkurrenzprodukten zu verstecken. Ganz im Gegenteil, selbst einige C64-Programme können sich an diesem Programm noch ein Beispiel nehmen. Der Text-Manager arbeitet mit "SCROLLING" in allen Richtungen. Der Bildschirm dient als Fenster auf den Text, welches in beliebige Richtungen verschoben werden kann. Auch das heute so beliebte "Wordwrapping" ist in das Programm integriert. Dadurch ist es möglich, den Text, ohne auf den Seitenrand achten zu müssen, fließend einzugeben. Wird der Rand durch ein Wort überschrieben, so wird dieses automatisch in die nächste Zeile gesetzt. Das

Ganze läuft so schnell ab, daß der Anwender diesen Vorgang nicht bemerkt. Die Textbreite kann beliebig im Bereich zwischen 35 und 99 Zeilen variiert werden. Änderungen sind einfach und jederzeit möglich, wobei der Text automatisch der neuen Zeilenbreite angepaßt wird. Der Texteditor arbeitet vollkommen bildschirmorientiert. Das bedeutet, daß mit Hilfe der vier Cursor-Tasten, der Tasten HOME und CLEAR sowie der Funktionstasten, beliebig auf dem Bildschirm umhergewandert und editiert, gelöscht oder eingefügt werden kann. Um das Erstellen eines Textes so weit wie möglich zu erleichtern, bietet der Textmanager eine Reihe leistungsfähiger Kommandos. Diese Kommandos werden einfach durch Drücken der Taste CTRL eingeleitet. Hier stehen folgende Funktionen zur Verfügung:

### FIND

Von der Cursorposition aus, wird nach einem bestimmten Zeichen, Wort oder Satz gesucht

### REPLACE

Eine bestimmte Zeichenkette wird gesucht und durch eine andere ersetzt

### DISK COMMAND

Kommandos können an die Floppy geschickt werden. Beispielsweise kann eine Diskette formatiert werden

### SAVE

Ein Text kann abgespeichert werden. Hier hat man die Wahl zwischen Kassetten oder Diskette

## Multidatei

### ein universelles Dateiverwaltungssystem für C 64

**Programm:** Multidatei  
**System:** C64/C128, Schneider  
**Preis:** (Kass) 59 Mark, (Disc) 69  
Mark  
**Hersteller/Vertrieb:** Ariola Soft

Eine Dateiverwaltung ist mit Sicherheit nichts Neues, wird jedoch trotzdem immer wieder mal benötigt. Nicht zuletzt aus diesem Grund bietet jeder Hersteller, der etwas auf sich hält, mindestens zwei Programme dieser Kategorie an. Auch die Firma ARIOLA-SOFT macht hier keine Ausnahme. In diesem Test habe ich mir das universelle Programm Multidatei genauer angeschaut. Kurz nach der Eingabe des Ladebefehls erscheint das erste Menü. Hier hat man eigentlich die Wahl zwischen vier verschiedenen Dateiprogrammen:

Version 1 mit max. 40 Datensätze bei 40 Feldern  
Version 2 mit max. 80 Datensätze bei 20 Feldern  
Version 3 mit max. 160 Datensätze bei 10 Feldern  
Version 4 mit max. 400 Datensätze bei 4 Feldern

Wie schon diesen Daten zu entnehmen ist, eignet sich das Programm nur für kleinere Anwendungen im Hobbybereich. Ein Beispiel wäre eine Adressdatei, Büchersammlung oder Schallplattenarchiv. Aber auch bei diesen

Anwendungen kann es, bei einer maximalen Datensatzanzahl von 400 bzw. noch weniger, schnell zu Schwierigkeiten kommen. Man sollte sich deshalb vor dem Kauf überlegen, ob sich das Programm für die zu verwaltende Datenmenge eignet. Ich habe mich in meinem ersten Test für die Version 4 entschieden. Nach etwas längerer Wartezeit, immerhin fast 1.5 Minuten, erscheint ein weiteres Menü. An dieser Stelle kann die Rand- sowie die Hintergrundfarbe gewählt werden. Da ich die Voreinstellung, blaue Schrift auf schwarzem Hintergrund, sehr angenehm empfand, übergang ich dieses Menü. Endlich war es soweit, das Hauptmenü erschien:

**DATEI EINRICHTEN & EIN-  
GABE**  
**DATEI LADEN**  
**DATEI ABSPEICHERN**  
**DS ÄNDERN & LÖSCHEN**  
**DATENSATZ / FELD SU-  
CHEN**  
**DATEI SORTIEREN**  
**DATEI LISTEN &  
DRUCKEN**  
**RECHNUNG & GRAFIK &  
CLR/ENDE**  
**DISKETTEN SYSTEMAR-  
BEIT**

Die Bedienung wird ab jetzt vollkommen durch dieses Hauptme-  
▶

### LOAD

Ein Text kann von Diskette oder Kassetten geladen werden

### PRINT

Mit diesem Befehl kann der Text über einen Drucker ausgegeben werden. Der Textmanager druckt im Blocksatz, das heißt, der durch das Wordwrapping erzeugte Flattersatz wird ausgeglichen, indem eine Zeile durch Einfügen von Leerzeichen zwischen den einzelnen Wörtern aufgefüllt wird. Das Ergebnis ist ein vollkommen gleichmäßiger rechter Rand. Das Programm erlaubt den Ausdruck

auf Einzelblatt- oder Endlospapier. Da der Rechnerspeicher mit 16KB relativ klein ist, besteht auch die Möglichkeit, getrennt abgespeicherte Teilstücke als Ganzes auszudrucken.

Insgesamt kann man sagen, das TEXTMANAGER für den Hobbybereich mehr als nur geeignet ist. Es ist fast unglaublich, aber man erhält auf diese Weise ein Textverarbeitungssystem mit Computer, Drucker und Programm schon unter 500 DM. Etwas nachteilig macht sich lediglich der kleine Speicher bemerkbar.  
Frank Brall



nü gesteuert. Dies ist besonders für Anfänger von Vorteil, denn Eingabefehler werden fast völlig ausgeschlossen. Obwohl ich noch keinen Blick in das deutsche Bedienungsbandbuch geworfen hatte, gab es praktisch keine Probleme bei der Handhabung der verschiedenen Funktionen.

Nachteil: auch der geübte Anwender kommt eigentlich erst über mehrere Untermenüs zum Ziel. Zusätzliche Funktionstasten hätten hier Abhilfe schaffen können! Die eigentlichen Funktionen von Multidatei entsprechen denen eines Standard-Dateiprogrammes und reichen für den Hobby-Anwender vollständig aus. Es kann eine eigene Maske definiert werden, welche die entsprechenden Eingabefelder markieren, beispielsweise Name, Vorname oder Ort. Die einzelnen Datensätze können später nach einem dieser Felder sortiert werden. Leider wird bei der Sortieroutine nur nach der Nummer des entsprechenden Feldes gefragt, was bedeutet, daß man sich die Zuordnung der Feldbezeich-

nungen merken bzw. notieren muß. Sicherlich hätte man hier eine bessere Lösung treffen können. Die Sortieroutine arbeitet relativ schnell, benötigt jedoch die Floppy als Zwischenspeicher. Neben den Standard-Funktionen besitzt Multidatei auch die Möglichkeit, eine Säulengrafik zu erstellen. In dieser Betriebsart können bestimmte Datenfelder addiert und im Verhältnis zu den anderen Datensätzen grafisch dargestellt werden. Gerade dieser Menüpunkt hebt das Programm positiv aus der Vielzahl der Dateiprogramme hervor. Allerdings eignet sich Multidatei, wegen der begrenzten Datensatzanzahl, kaum für professionelle Anwendungen. Frank Brall

#### Positiv:

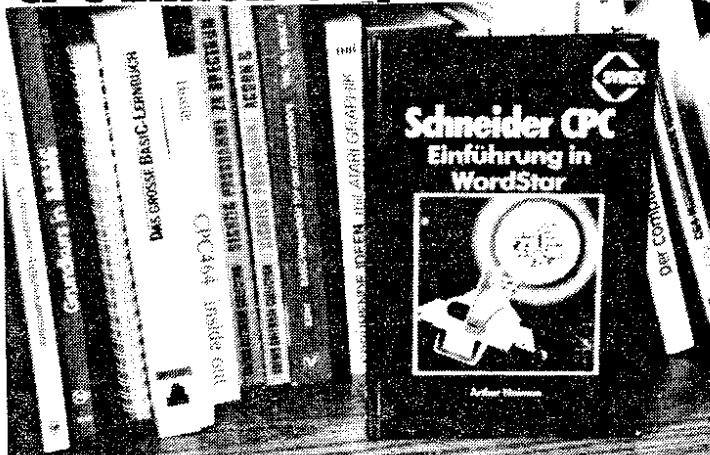
Addition und grafische Darstellung beliebiger Felder  
Sortieroutine für alle Datenfelder  
Einfache Bedienung

#### Negativ:

Für den geübten Anwender viele Umwege über Menü's

### Buchbesprechung:

## CPC-Einführung in WordStar



Aufgepaßt: Anwender aller Schneider CPC-Computer erhalten hier eine umfassende Einführung in das äußerst populäre und leistungsfähige Textverarbeitungsprogramm. Der SYBEX-Bestseller "Arbeiten mit WordStar" wurde vollständig an die CPC-Systemumgebung angepaßt und durch viele zusätzliche Hinweise ergänzt.

Inhaltlich geht das umfassende Nachschlagewerk gezielt vor. Angefangen bei dem Einstieg in die Textverarbeitung mit WordStar, bis hin zu den wichtigen System-

patches. Schrittweise führt man den Anfänger, wie auch den Profi, über die Menüs, Texte editieren, Handhabung von Dateien, Textgestaltung auf dem Bildschirm, Normaler und MIX-Druck (MailMerge) und Hinweisen zur Drucker-Installation zum Schlußkapitel.

Das Buch ist nicht zuletzt wegen des übersichtlichen Aufbaus ein willkommener Wegweiser für Anfänger und Fortgeschrittene.

PREIS: 42,- DM

SYBEX-Verlag GmbH, Postfach 300961, 4000 Düsseldorf

## Schneller Compiler für den Schneider!

Programm: Turbo Basic  
System: :CPC 464/664/6128  
Preis: ca. 100 Mark (Disc)  
Hersteller: Hisoft  
Vertrieb: Datensysteme, Nürnberg

Wer kennt nicht das Problem? Man schreibt ein langes Basic Programm und stellt am Schluß fest, daß es viel zu langsam abläuft. Auch wenn das Spiel noch so gut ist, so macht doch oft nur die Geschwindigkeit den Reiz des Programmes aus. Auch Adventures machen hier keine Ausnahme, denn die interne Wort- bzw. Satzzerkennung muß enorm schnell arbeiten, da sonst riesige Wartezeiten entstehen können. Um dieses Problem zu beseitigen, gibt es schon viele Jahre die sogenannten Compiler. Ein solcher Compiler übersetzt das Programm entweder in Maschinensprache oder in einen P-Code. Im Falle eines P-Code Compilers, wird eine Art Zwischencode erzeugt, welcher später von einem Interpreter ausgeführt wird. Bei Turbo Basic handelt es sich jedoch um einen echten MC-Code Compiler. Das bedeutet, das Basic-Programm wird direkt in die Z80 Maschinensprache übersetzt. Dies bedeutet, daß auch gegenüber eines P-Code Compilers eine wesentlich höhere Geschwindigkeitssteigerung erzielt wird. Je nach Basic-Programm erzeugt der Compiler ein zwischen 7 und 80 mal schnelleren Ablauf des Programmes. Um diese Geschwindigkeiten zu erreichen, mußten jedoch einige

Einschränkungen im Basic gemacht werden. Das bedeutet: Will man ein Programm compilieren, so muß man sich an bestimmte Vorschriften des Compilers richten. Vor allem ist der Befehlssatz sehr eingeschränkt, da der Compiler nur Integer-Zahlen verarbeitet. Funktionen, die mit Fließkommazahlen arbeiten, wie beispielsweise SIN, COS oder Wurzelfunktion, sind dadurch nicht mehr verwendbar. Da bei Spielprogrammen jedoch selten Fließkommazahlen Verwendung finden, dürften hier keine Probleme entstehen. Bei Kalkulation und ähnlichen Programmen dürften hier jedoch Schwierigkeiten aufkommen. Ansonsten hält sich der Compiler recht genau an die Syntax des Schneider-Basic. Eine Bedingung ist jedoch, daß Variablen bei einem Neustart nicht auf 0 gesetzt werden. Das heißt, daß alle Variablen am Programmfang definiert werden müssen. Selbst das mitgelieferte Demoprogramm OHELLO hat hier Probleme: nach einem ersten Spiel läßt sich kein leeres Spielfeld mehr erzeugen. Offensichtlich haben hier die Programmierer Ihre eigene Syntax nicht beachtet. Ansonsten zeigen die beiden mitgelieferten Demoprogramme, Othello und Space-Invaders, was in dem Compiler steckt. Selbst das in Basic völlig unbrauchbare Spiel ist in kompilierter Form noch zu vertreten. Für die Programmierer von Spielprogrammen kann ich diesen Compiler deshalb nur empfehlen. Frank Brall

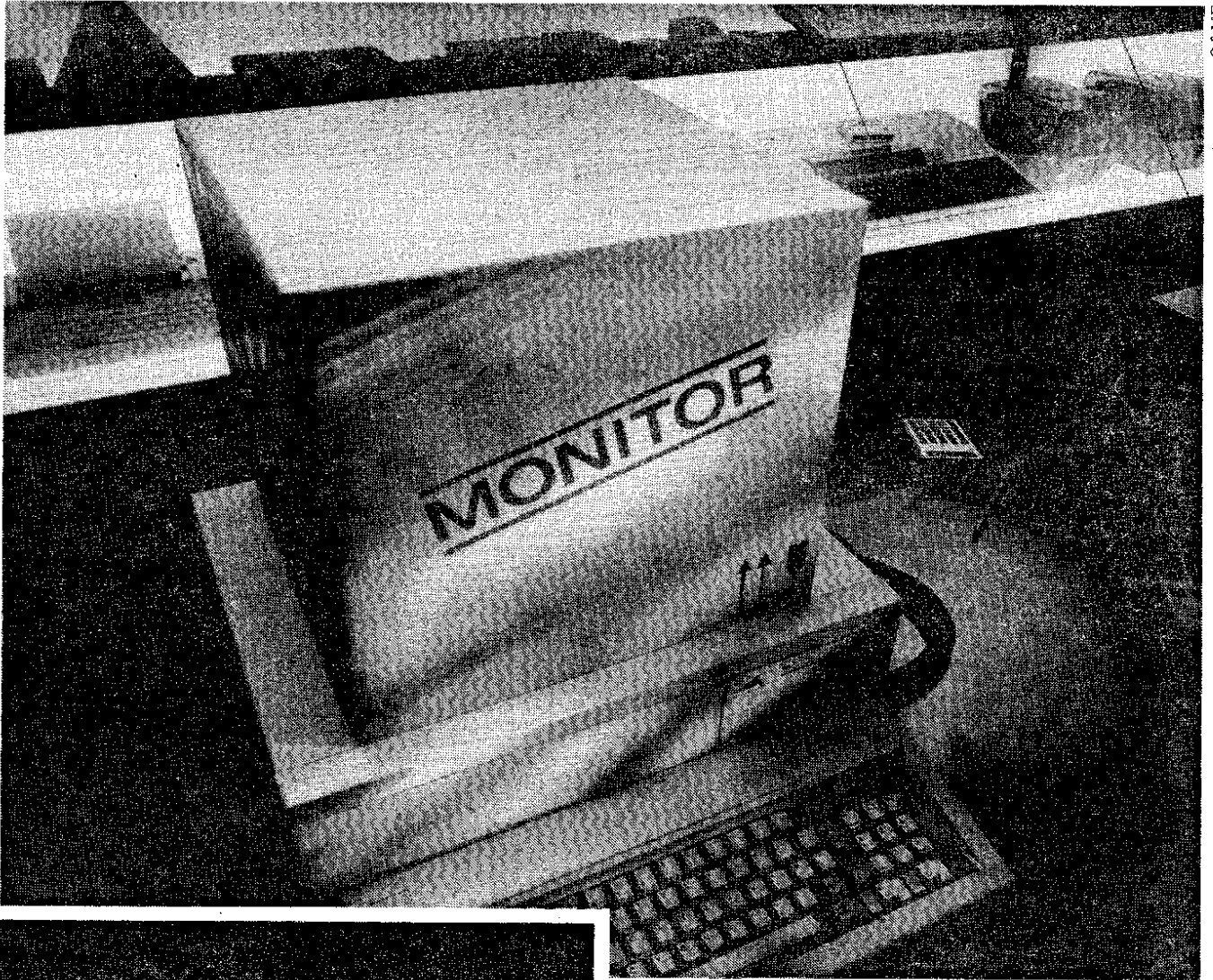
### Korrekturen:

Heft 9 bescherte uns leider wieder einige Fehler, die jedoch die Lauffähigkeit der betroffenen Programme nicht beeinträchtigt. In "Cave of Death" Teil 4 ist eine PRINT-Zeile verdeckt, die aber nur besagt, daß nun die Datazeilen eingelesen werden. "Bitte warten.." ist also vollkommen ausreichend.

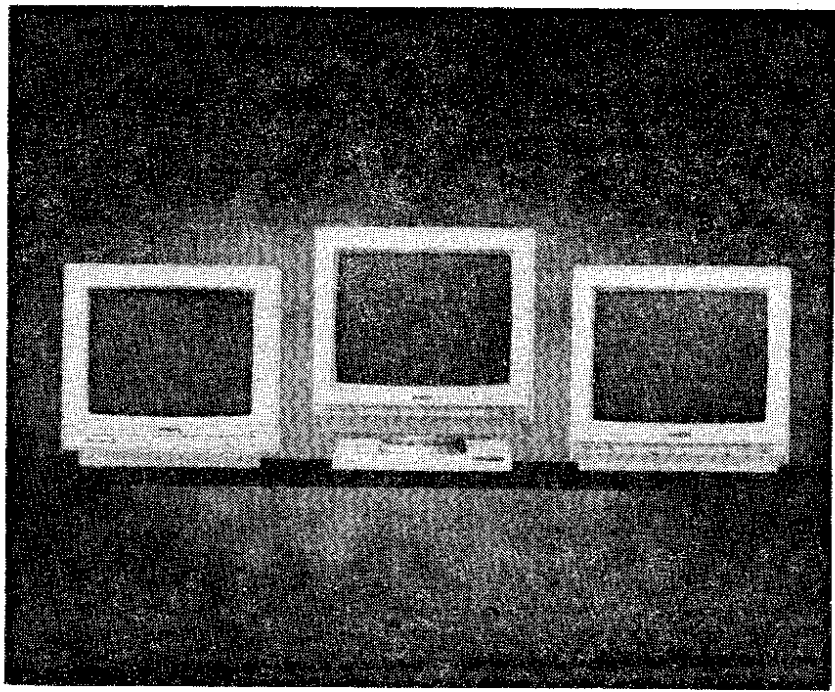
Bei "Reversi" verzweigt das Programm, falls man eine Pause wünscht in das Nichts. Der Autor hat wohl für diesen Teil seines Programmes noch keine Zeit gehabt.

"Moon Patrol" weist bei einigen PRINT-Zeilen einen kleinen Fehler auf. Und zwar fehlt bei diesen nach dem letzten Klartextwort die Anzahl eben dieser Zeichen. Ursache dafür ist die Tatsache, daß der Autor seine PRINT-Zeilen nicht ordnungsgemäß mit Gänsefüßchen abgeschlossen hat. Aber auch dieser Fehler läßt sich auch von weniger erfahrenen Lesern ausbügeln.





O&MF



## Wunderbar, wenn ein echter Philips Monitor drin wäre...

**W**enn Ihnen jemand ein supergünstiges Paket-Angebot macht, fragt es sich, ob die Qualität nicht auf der Strecke geblieben ist. Beim Computer ist dies weniger zu befürchten. Ins Auge geht es jedoch, wenn am Monitor gespart wurde. Am besten, Sie entscheiden sich deshalb gleich für einen echten Philips Monitor: Der bringt Ihnen durch die hohe Auflösung eine erhebliche Verbesserung der Wiedergabe und der Lesbarkeit. Der schont die Augen, statt sie zu ermüden – auch bei längerer Nutzung. In Farbe oder Monochrom (Bernstein, Grün oder Papierweiß) bieten Ihnen die Philips Monitore Kompatibilität mit fast allen Systemen und optimale Zukunftssicherheit. Klar und deutlich zeigen sie, daß sie mindestens so gut sind wie Ihr Personal-Computer. Und das sollte Ihnen etwas wert sein.

Mehr darüber von Philips GmbH, Geschäftsbereich Neue Medien, Postfach 1014 20, 2000 Hamburg 1.

 Philips paßt  
in unsere Zukunft



**PHILIPS**

# Leser & Meckerecke

"Interface", Heft 8/86

Ich bin neulich in den Zeitungsladen bei uns gegangen, um mir die neue "Compute mit" zu kaufen. Ich war total begeistert, als ich folgendes auf der Titelseite sah: "Luxus-Userport für C16/116..."

Natürlich habe ich die Zeitschrift gekauft. Jedoch lief, als ich mir

die Anleitung zum Bau des Interfaces durchlas, ein kalter Schauer über den Rücken. Ich habe nämlich gar nichts verstanden. Es wäre besser gewesen, wenn Sie einen kompletten Schaltplan abgebildet hätten.

Wäre es vielleicht möglich, einen Ätzplan von dem Interface in "Compute mit" abzubilden? ...

Thorsten Hoppensdedt, Celle

Redaktion: Tatsächlich war unsere Anleitung für Anfänger nur sehr schwer zu verstehen. Aber auch eine ausführlichere Version würde Anfängern nicht helfen, eher würde sie dazu anleiten, derartige Bauten vorzunehmen, ohne daß die entsprechenden Kenntnisse vorhanden sind - dann meistens mit der entsprechenden negativen Erfolgsmeldung. Wir haben uns daher darüber Gedanken gemacht, wie wir unseren Lesern solche Hardwarebasteleien narrensicher anbieten können. Herausgekommen ist dabei der "Tronic-Hardwareservice". Wir verschicken in Zukunft zu den Anleitungen, die wir veröffentlichen, komplette Bausätze mit ausführlichen Anleitungen und, wenn nötig, den fertigen Leerplatinen. Den Anfang macht diesen Monat die Speichererweiterung aus Heft 7/86. Für 45,- DM bekommen Sie in einer Hardbox die entsprechenden Speicherbauteile

ne, den Umschalter für 16/64 K (Qualitätsschalter) und Bauanleitung für beide Systeme (C16 und C116). (Bestell-Nummer HWS-1-786)

Werkstatt, "Compute mit" Sonderausgabe Heft 2

Als erstes sage ich, ein gutes Heft, wie auch das erste. Interessant fand ich die Beschreibungen des Kassettenports. Könnten Sie auch mal die Serial-Buchse erklären? Dazu die Frage: Kann ich meine Brother EP 44-Schreibmaschine (mit eingebautem Interface und RS 232-Anschlußmöglichkeit an meinem C-16 anschließen?...

Dietmar Surborg, Hannover

Redaktion: Man kann. Sie benötigen dazu einen Userport, wie er in Heft 8 beschrieben ist. RS 232 und Centronicsroutinen sind in Arbeit, ebenso die komplette Erweiterung incl. Software für unseren Hardwareservice.

## The never ending story von Uwe Knierim

# GEOS intern

(2)

### Zusätzliche Fileinformationen unter »Geos«.

Nachdem wir in der letzten Ausgabe die Struktur des Directoryeintrages unter dem GEOS-Betriebssystem besprochen haben, wollen wir in dieser Ausgabe den Fileheaderblock etwas genauer unter die Lupe nehmen.

Zur Wiederholung ist nebenstehend der GEOS-Directoryeintrag noch mal tabellarisch abgedruckt. Unentbehrliches Werkzeug für diesen GEOS-Kurs ist ein Diskmonitor.

Nur fortgeschrittene Floppyprogrammierer werden in der Lage sein, sich die notwendigen Informationen direkt von der Diskette zu besorgen.

#### Der Fileheaderblock

Wie in der letzten Ausgabe besprochen, weisen die Bytes 19 und 20 des Directoryeintrags auf den GEOS-Fileheaderblock hin, in dem alle für das DeskTop wichtigen Informationen stehen. Im Folgenden soll beschrieben werden, wie man für ein beliebiges Programm den File Header selbst erstellen kann. Sollte das zu ladende Basicprogramm eine eigene Laderoutine in Form eines Maschinenprogramms haben, von dem aus das eigentliche Programm erst nachgeladen wird, muß natürlich dieses Bootfile mit der GEOS-Erweiterung versehen werden.

#### Einen Block reservieren

Als erstes muß ein freier Block gefunden werden, der den File Header aufnimmt. Profis werden den Allocate-Befehl des FloppyDOS

verwenden, hier soll ein einfacher Weg beschrieben werden. Mit einem einfachen Einzeiler-Basic-Programm reservieren wir uns in der BAM einen freien Diskettenblock.

```
10 GOTO 10
```

```
SAVE"NAME",8
```

Mit Hilfe eines Diskettenmonitors suchen wir nun den Directoryeintrag "NAME" (erster Block bei Track 18 Sektor 1). Das erste Byte überschreiben wir mit 00, was bedeutet, daß das File gelöscht wurde. Die nächsten beiden Bytes enthalten die Werte von dem Track und dem Sektor des Blocks, in dem unsere schlichte Programm abgelegt wurde. Diese Werte müssen wir nun in Byte 19 und 20 des Directoryeintrags des Files eintragen, welches wir von GEOS aus starten wollen. GEOS vermutet nun den File

Header in dem Block, in dem bislang noch unser mittlerweile überflüssiges Basicprogramm "NAME" steht. Mit dem Diskmonitor können wir uns nun daran machen, diesen Block mit den entsprechenden Fileheader - Informationen zu füllen.

#### Die Struktur des File Headers

1. Die ersten zwei Bytes sind gewöhnliche Commodore DOS Informationen, die den nächsten Block und das letzte benutzte Byte des aktuellen Blocks anzeigen. Da der File Header nur aus einem Block besteht und bis zum letzten Platz aufgefüllt ist, müssen hier die Werte \$00 und \$FF stehen.
2. Die Bytes 2 - 3 enthalten die Breite und die Höhe des Icons. Dies ist wichtig, damit für zukünftige GEOS-Entwicklungen Kompatibilität gewährleistet

# NEU: Hardwareservice des Tronic-Verlags

**Unser 1. Angebot:**

**64 K-Speichererweiterung für C16 + C116**

**Bausatz – bestehend aus: allen Bauteilen und ausführlicher Anleitung, umschaltbar 64K/16K – geliefert in stabiler Hartbox.**

**Bestell-Nr. HWS-1-786 45,- DM + NN**



## So bestellen Sie:

Der Mindestbestellwert von Software liegt bei **15,00 DM!**

Sie können das Geld in bar oder per Scheck Ihrer schriftlichen Bestellung beilegen.

Bei Nachnahme Versand werden 4,00 DM für Porto Kosten erhoben.

Auslandsbestellungen werden nur per Vorauskasse entgegengenommen!

Rufen Sie uns doch einfach einmal an! Unsere Telefone sind werktags ab 9.30 Uhr für Sie zu erreichen!

## SOFTWARE-VERSAND

ANDREAS BACHLER  
POSTFACH 429  
D-4280 BUCHOLT

## Wir haben die neueste Software!

	Kassette		
<b>Commodore 16/116/Plus 4</b>		Out on a limb	19,95 DM
ACE	32,00 DM	Raider	9,95 DM
Bandits at Zero	14,95 DM	Rockman	9,95 DM
Baby Berks	9,95 DM	Scramble	14,95 DM
Bench Head	24,95 DM	Space Pilot	19,95 DM
Bomb Jack	29,00 DM	Street Olympics	9,95 DM
Classics (4 Spiele)	34,95 DM	Thai Boxing	19,95 DM
Classics II (4 Spiele)	34,95 DM	Water Grand Prix	9,95 DM
Commando	26,95 DM	Wimbledon	29,00 DM
Formula 1	9,95 DM	Winter Olympiade	29,00 DM
Frank Bruno's Boxing	25,00 DM	Daily Thompson	25,00 DM
Jet Set Willy	25,00 DM	Gull Wring Falcon	14,95 DM
Kikstart	9,95 DM	Kung Fu Kid	25,00 DM
Legionär	19,95 DM	Turbo Tape	18,95 DM
		Paint Box	25,00 DM

Fordern Sie kostenloses Informationsmaterial über unser Lieferprogramm für folgende Computer an:

COMMODORE 64/128, COMMODORE 16/116/PLUS 4, SCHNEIDER 464/664  
ATARI 130XE/260ST/520ST/800XL, SINCLAIR SPECTRUM und MSX

Bitte geben Sie immer Ihren Computertyp an!

**TELEFON-NR. (02871) 183088**





## A Block From the Directory Structure

Blocks (sectors) 1 - 19 on track 18

byte no.	Contents
0	Track of next directory block
1	Sector of next directory block
2	C-64 file type or'ed with \$80 to indicate closed file. 0=DELETED, 1=SEQUENTIAL, 2=PROGRAM, 3=USER, 4=RELATIVE
3	Track of first data block.
4	Sector number of first data block.
5	16 character file name padded with shift spaces, \$A0
21 - 22	Track, Sector of GEOS file Header (New Structure)
23	GEOS structure type: 0=SEQUENTIAL, 1=VLIR
24	GEOS File Types: 0=NOT_GEOS, 1=BASIC, 2=ASSEMBLY, 3=DATA, 4=SYSTEM, 5=DESK_ACC, 6=APPLICATION, 7=APPL_DATA, 8=FONT, 9=PRINTER, 10=INPUT_DEVICE, 11=DISK_DEVICE
25	Date: year last modified, offset from 1900
26	Date: month last modified (1 - 12)
27	Date: day last modified (1 - 31)
28	Date: hour last modified (0 - 23)
29	Date: minute last modified (0 - 59)
30 - 31	low byte, high byte: Number of blocks (sectors) in the file.
34 - 63	Directory Entry 2
66 - 95	Directory Entry 3
98 - 127	Directory Entry 4
130 - 159	Directory Entry 5
162 - 191	Directory Entry 6
194 - 223	Directory Entry 7
226 - 255	Directory Entry 8

D I R E C T O R Y E N T R Y

## GEOS File Header Structure

(128 bytes. New GEOS file. Pointed to by Directory Entry)		
Byte No.	Contents	Description
0-1	00,FF	00 indicates this is the last block in the file. FF is the index to the last valid data byte in the block.
2	3	Width of icon in bytes, always 3
3	21	Height of file icon in lines, always 21.
4	\$80 + 63	Bit Map data type. Top bit = 1 means the lower 7 bits contain the number of unique bytes which follow, i.e. 63. Always this value.
5 - 67	\$\$\$FFF,FFF ... \$\$\$FFF,FFF	Start of picture data total of 63 bytes used to define icon graphic End of picture data
68	\$80 + PRG	C-64 file type, used when saving the file under GEOS PRG = 1, Bit 7 = 0 Deleted, Bit 6 = 1 Write Protected. Bit 5, Marked for deletion.
69	\$01	GEOS file type: BASIC = 1, ASSEMBLY = 2, DATA = 3
70	0	GEOS structure type, 1 = VLIR, 0 = SEQUENTIAL
71 - 72	FileStart	Start address in memory for loading the program
73 - 74	FileEnd	End address in memory for loading the program
75 - 76	InitProg	Address of initialization routine to call after loading the program
77 - 96	Filename ,0,0,0,0,0	20 byte ASCII string containing file name for the application. Remaining bytes in Filename set to 0.
97 - 127	Reserved	Set to nulls (ascii \$0)
128-159	GraphString	32 bytes for a graphics string commands. Used to put up a preliminary screen image while the remainder of the file is being loaded. See chapter on graphics.
160-255	Get info	Used for the file menu option getinfo. String must be null terminated and null must be first char if string is empty.

ist, falls diese variable Icongrößen verwenden. Normalerweise haben die Icons die Standardgröße eines Sprites: 3 Bytes in der Breite und 21 Bytereihen in der Höhe.

- Ähnlich verhält es sich mit dem Byte 4, dem Bitmapflag, das normalerweise \$80 enthält.
- Die Bytes 5 - 67 repräsentieren die 63 Sprite-Bytes. Die mit ei-

nem Sprite-Editor gewonnenen Datas können hier mit dem Diskmonitor eingetragen werden (nachdem sie natürlich vorher in Hexcode umgewandelt worden sind).

- Byte 68 markiert den Commodore-Filetyp. Für PRG muß hier 2 eingetragen werden. Ferner muß Bit 7 gesetzt sein, damit hier ein ordnungsgemäß geschlossener File erkannt wird.

- Byte 69 kennzeichnet den GEOS Filetyp. Ist das nachzuladende Programm ein Basicprogramm, muß hier 1 eingetragen werden, bei einem Maschinenprogramm 2.

- Die Bytes 75 - 76 enthalten die Startadresse des zu ladenden Programms und sind daher in der Regel mit den Bytes 71 und 72 identisch.

- Byte 70 ist der GEOS-Struktur-Typ. GEOS kennt zwei verschiedene Strukturtypen von Files. Erstens die standardmäßige Blockorganisation (=0) oder die geospezifische Struktur VLIR, was **Variable Length Indexed Record** bedeutet. Bei eigenen Programmen, die nicht auf GEOS-Routinen zugreifen, ist dieses Byte, wie auch das Byte 21 des Directoryeintrags Null (\$00) zu setzen.

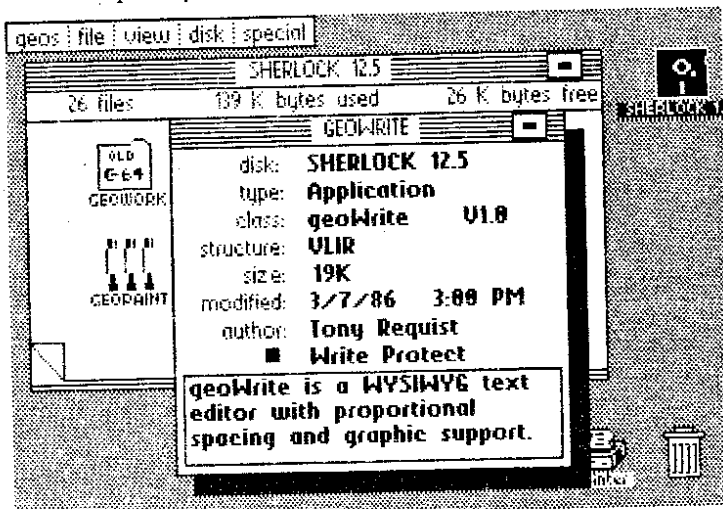
- Die Bytes 77 - 159 werden nur benötigt, wenn das nachzuladende Programm als echter GEOS-File abgelegt ist. Bei Fremdprogrammen werden diese Bytes ignoriert.

- Die Bytes 160 - 255 enthalten den GetInfo-Textstring, der vom DeskTop aus angezeigt wird, wenn man **get info** wählt. Hier sind üblicherweise Benutzerhinweise oder Copyrightvermerke des Autors eingetragen. Dem letzten benutzten Stringdata muß unbedingt ein \$00 folgen. Wird dieser Eintrag nicht benutzt, muß das Byte 160 \$00 enthalten.

### Rückkehr zum DeskTop

In der nächsten Folge veröffentlichen wir ein kurzes Maschinenprogramm, daß am Ende eines eigenen Programms die Rückkehr zu GEOS erlaubt.

- Die Bytes 71 - 72 enthalten die Ladeadresse des Programms. Bei Basicprogrammen ist dies \$0801 - bei Maschinenprogrammen muß hier die entsprechende Ladeadresse eingetragen werden.
- Die Bytes 73 - 74 werden bei Fremdfilen von GEOS ignoriert.



## TREASURE ROBBER VC-20

Schatzräuber, wir schnappen dich

Sie sind in die Rolle eines Schatzräubers gesschlüpft. Allerdings machen Ihnen die Geister zu schaffen, die aus dem Computer kommen, um die Schätze zu schützen.

Am Anfang des Spieles stehen Sie in der rechten oberen Bildschirmcke. In der linken unteren Ecke lauert schon der Geist. Er

kann durch Wände gehen und weiß ausserdem immer, wo sich befinden und nimmt immer den kürzesten Weg um Sie zu fangen.

Ihre Aufgabe ist es nun, alle Schätze und Münzen einzusammeln. Haben Sie Ihre Aufgabe bewältigt, erklingt eine Melodie und das Spiel beginnt unter erschwerten Bedingungen neu.

\* \* \*



### Teil 1

#### VC20 LISTING + CHECKSUMMEN (VC V1.0)

```

5 POKE56,28:CLR <24>
6 POKE36879,8 <226>
7 PRINT"(CLEAR DOWN8 RIGHT4 WHITE)BITTE<
RIGHT>WARTEN<SPACE>!" <92>
8 FORT=7168TD7482:READF:POKET,F:S=S+F:NE
XT <42>
9 IFS<>24167THENPRINT"(CLEAR)FEHLER<SPAC
E>IN<SPACE>DEN<SPACE>DATAS<SPACE>!":END <240>
10 POKE36869,255 <160>
11 PRINT"(CLEAR DOWN2 RIGHT3 YELLOW)FGHI
JKGH<RIGHT>GLMMHG" <80>
12 PRINT"(RIGHT3 RED RVSON)
-----" <158>
13 PRINT"(DOWN3 RIGHTS WHITE)A<RIGHT RVS
ON>7/1985<RIGHT>BY" <234>
14 PRINT"(DOWN RVSON RIGHT4)OLIVER<RIGHT
>WINDEL" <79>
15 PRINT"(DOWN RVSON RIGHT3)EHRLICHERSTR
.30" <253>
16 PRINT"(DOWN RVSON RIGHT3)3200<RIGHT>H
ILDESHEIM" <66>
17 PRINT"(DOWN5)BC"SPC(17)"BC" <194>
18 PRINT"(BLUE)DE<RIGHT2 CYAN RVSON>-ZUM
<RIGHT>STARTEN-<RIGHT2 RVSOFF BLUE>DE" <70>
19 PRINT"(WHITE)BC"SPC(17)"BC" <142>
20 BETA$:IFA$="(F1)"THEN22 <72>
21 GOTO20 <146>
22 PRINT"(CLEAR DOWN RVSON YELLOW)SPIELA
NLEITUNG:" <161>
23 PRINT"(RED RVSON)
-----" <228>
24 PRINT"(DOWN2 WHITE RVSON)SIE<RIGHT>SI
ND<RIGHT>IN<RIGHT>EINE<RIGHT>BURG" <251>
25 PRINT"(RVSON)EINGEDRUNGEN,<RIGHT>UM<R
IGHT>DIE" <83>
26 PRINT"(RVSON)REICHTUEMER<RIGHT>DIESER
<RIGHT>ZU" <58>
27 PRINT"(RVSON)STEHLEN." <31>
28 PRINT"(RVSON)ABER<RIGHT>LEIDER<RIGHT>
TREIBT<RIGHT>EIN"; <121>

```

```

29 PRINT"(RVSON)GEIST<RIGHT>'<RVSOFF CYA
N>N<WHITE RVSON>'<RIGHT>IN<RIGHT>DER<RIG
HT>BURG" <76>
30 PRINT"(RVSON)SEIN<RIGHT>UNWESEN." <73>
31 PRINT"(RVSON)SAMMELN<RIGHT>SIE<RIGHT>
SOVIEL" <79>
32 PRINT"(RVSON)MUENZEN<RIGHT>&<RIGHT>SC
HAETZE<RIGHT>EIN"; <124>
33 PRINT"(RVSON)WIE<RIGHT>SIE<RIGHT>NUR<
RIGHT>KOENNEN." <2>
34 PRINT"(DOWN4)BC"SPC(17)"BC" <181>
35 PRINT"(BLUE)DE<RIGHT2 CYAN RVSON>-ZUM
<RIGHT>STARTEN-<RIGHT2 RVSOFF BLUE>DE" <87>
36 PRINT"(WHITE)BC"SPC(17)"BC" <159>
37 BETA$:IFA$="(F1)"THEN39 <200>
38 GOTO37 <186>
39 PRINT"(CLEAR RIGHT2 DOWN YELLOW RVSON
)PUNKTWERUNG:" <27>
40 PRINT"(RED RVSON RIGHT2)
-----" <197>
41 PRINT"(RIGHT2 DOWN3 YELLOW)O<WHITE RI
GHT RVSON>=<RIGHT6>1<RIGHT>PUNKT" <204>
42 PRINT"(RIGHT2 DOWN YELLOW)P<WHITE RIG
HT RVSON>=<RIGHT5>10<RIGHT>PUNKTE" <252>
43 PRINT"(RIGHT2 DOWN YELLOW)O<WHITE RIG
HT RVSON>=<RIGHT5>20<RIGHT>PUNKTE" <21>
44 PRINT"(RIGHT2 DOWN YELLOW)R<WHITE RIG
HT RVSON>=<RIGHT5>30<RIGHT>PUNKTE" <46>
45 PRINT"(RIGHT2 DOWN YELLOW)S<WHITE RIG
HT RVSON>=<RIGHT>UNBEK.<RIGHT>PUNKTE" <219>
46 PRINT"(DOWN4)BC"SPC(17)"BC" <193>
47 PRINT"(BLUE)DE<RIGHT2 CYAN RVSON>-ZUM
<RIGHT>STARTEN-<RIGHT2 RVSOFF BLUE>DE" <99>
48 PRINT"(WHITE)BC"SPC(17)"BC" <171>
49 BETA$:IFA$="(F1)"THEN51 <126>
50 GOTO49 <206>
51 PRINT"(CLEAR RVSON DOWN5 RIGHT3)VC-20
<SPACE>LAEDT<SPACE>DEN" <185>
52 PRINT"(DOWN RVSON RIGHTS)2.<SPACE>TEI
L<SPACE>VON" <237>
53 POKE 631,131:POKE 198,1:STOP:REM NACH
LADEN <18>
54 POKE631,19:POKE632,13:POKE633,82:POKE
634,117:POKE635,58:POKE636,13:POKE198,6 <151>
55 DATA028,028,008,062,093,093,020,054 <75>
56 DATA062,065,077,081,081,077,065,062 <97>
57 DATA000,117,085,117,069,071,000,000 <245>
58 DATA000,117,069,119,021,117,000,000 <96>
59 DATA127,113,119,115,119,119,127,000 <219>
60 DATA255,239,207,239,239,199,255,000 <153>
61 DATA126,126,024,024,024,024,024,024 <36>
62 DATA126,126,102,126,126,120,108,102 <19>
63 DATA126,126,096,120,120,096,126,126 <156>
64 DATA126,126,102,126,126,102,102,102 <111>
65 DATA126,126,096,126,126,006,126,126 <167>
66 DATA102,102,102,102,102,102,126,126 <189>
67 DATA126,126,102,102,102,102,126,126 <2>
68 DATA124,126,102,124,124,102,126,124 <71>
69 DATA093,063,042,062,034,062,042,042 <134>
70 DATA000,000,000,024,024,000,000,000 <145>
71 DATA062,062,028,028,008,008,028,000 <156>
72 DATA085,085,127,028,008,008,028,000 <211>
73 DATA008,028,008,073,107,127,127,000 <197>
74 DATA008,020,034,127,034,020,008,000 <115>
75 DATA251,251,251,000,223,223,223,000 <77>
76 DATA000,023,021,021,021,023,000,000 <233>
77 DATA000,119,021,117,069,119,000,000 <243>
78 DATA000,119,021,117,021,119,000,000 <8>
79 DATA000,087,085,117,021,023,000,000 <7>
80 DATA000,119,069,117,021,119,000,000 <150>
81 DATA060,126,090,126,060,036,219,090 <182>
82 DATA000,119,068,116,020,119,000,000 <104>
83 DATA000,119,085,087,086,117,000,000 <128>
84 DATA000,112,064,096,064,112,000,000 <96>
85 DATA000,087,082,114,082,087,000,000 <41>
86 DATA000,117,069,087,085,117,000,000 <121>
87 DATA000,000,000,000,000,000,000,000 <206>

```

```

88 DATA024,024,008,028,122,024,028,052 <240>
89 DATA024,024,016,056,094,024,056,044 <198>
90 DATA169,127,141,034,145,173,032,145 <136>
91 DATA041,128,133,251,169,255,141,034 <202>
92 DATA145,173,031,145,041,028,024,101 <61>
93 DATA251,133,251,173,031,145,041,032 <120>
94 DATA133,252,096 <26>

```

ENDE DES LISTINGS



## Teil 2

VC20 LISTING + CHECKSUMMEN (VC V1.0)

```

10 POKE36879,8:0=4 <28>
11 POKE36869,255:POKE36878,15 <13>
12 GOT077 <168>
13 PRINT"(CLEAR DOWN BLUE)TTTTTTTTTTTTTT
TTTTTTTTT(RIGHT2 CYAN)F(PURPLE)G(GREEN)H
(RED)I(CYAN)J(PURPLE)K(GREEN)G(RED)H(RIG
HT2 CYAN)G(PURPLE)L(GREEN)M(RED)M(CYAN)H
(PURPLE)G(RIGHT2 BLUE)T"; <236>
14 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTT(RIGHT H
HITE)(\)(RIGHT7)^_(RIGHT7 BLUE)T"; <71>
15 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTT(YELLOW)
0000000000000000(SPACE BLUE)T"; <237>
16 PRINT"TYELLOW)O(BLUE)TTT(YELLOW)O(BL
UE)(YELLOW)O(BLUE)TYELLOW)O(BLUE)TT(YE
LLOW)O(BLUE)TYELLOW)O(BLUE)TYELLOW)O(B
LUE)TTT(YELLOW)O(BLUE)TYELLOW)O(BLUE)T
(YELLOW)000(BLUE)TYELLOW)0000000(BLUE)
TYELLOW)000(BLUE)TYELLOW)O(BLUE)T"; <24>
17 PRINT"TYELLOW)O(BLUE)TYELLOW)O(BLUE)
TTT(YELLOW)O(BLUE)TTTTTT(YELLOW)O(BLUE)
TTT(YELLOW)O(BLUE)TYELLOW)O(BLUE)TT(YEL
LOW)0000000(BLUE)TTTT(YELLOW)0000000(B
LUE)T"; <106>
18 PRINT"TYELLOW)O(BLUE)TYELLOW)O(BLUE)
TTTT(YELLOW)000000(BLUE)TTTT(YELLOW)O(B
LUE)TYELLOW)O(BLUE)TT(YELLOW)O(BLUE)TY
ELLOW)000000(BLUE)TTTT(YELLOW)000000(BL
UE)TYELLOW)O(BLUE)T"; <202>
19 PRINT"TYELLOW)000(BLUE)TTTTTTTTTTTTTT
TYELLOW)000(BLUE)TYELLOW)O(BLUE)TYEL
LOW)000000(BLUE)TTTT(YELLOW)000000(BLUE)
TYELLOW)O(BLUE)T"; <178>
20 PRINT"TYELLOW)O(BLUE)TYELLOW)O(BLUE)
TTTT(YELLOW)000000(BLUE)TTTT(YELLOW)O(B
LUE)TYELLOW)O(BLUE)TYELLOW)00000000(B
LUE)TTTT(YELLOW)00000000(BLUE)T"; <142>
21 PRINT"TYELLOW)O(BLUE)TYELLOW)O(BLUE)
TTT(YELLOW)O(BLUE)TTTTTT(YELLOW)O(BLUE)
TTT(YELLOW)O(BLUE)TYELLOW)O(BLUE)TT(YEL
LOW)O(BLUE)TYELLOW)000(BLUE)TYELLOW)00
000000(BLUE)TYELLOW)000(BLUE)TYELLOW)O
(BLUE)T"; <75>
22 PRINT"TYELLOW)O(BLUE)TTT(YELLOW)O(BL
UE)TYELLOW)O(BLUE)TYELLOW)O(BLUE)TT(YE
LLOW)O(BLUE)TYELLOW)O(BLUE)TYELLOW)O(B
LUE)TTT(YELLOW)O(BLUE)TYELLOW)00000000
000000000000(BLUE)T"; <174>
23 PRINT"(BLUE)TTTTTTTTTTTTTTTTTTTTTTT(HO
ME)" <129>
24 F=30720:D=5:G=16:D1=7:D2=15:S=36874 <252>
25 A=7832:AX=20:AY=6 <235>
26 B=8121:BX=1:BY=20 <199>
27 FORT=1TOD <234>
28 I=INT(RND(1)*327) <151>
29 IFPEEK(I+7791)<>15THEN28 <1>

```

```

30 POKEI+7791,6:POKEI+7791+F,7 <180>
31 NEXT <161>
32 IFG=19THEN34 <31>
33 G=G+1:D=D-1:GOTO27 <213>
34 SYS7448 <178>
35 K1=PEEK(251) <78>
36 A1=0:A2=0:A3=0 <0>
37 IFK1=148THENA1=22:A2=1:A4=0:GOTO41 <46>
38 IFK1=152THENA1=-22:A2=-1:A4=0:GOTO41 <87>
39 IFK1=140THENA1=-1:A3=-1:A4=33:GOTO41 <2>
40 IFK1=28THENA1=1:A3=1:A4=34 <115>
41 IFPEEK(A+A1)=20THENA1=0:A2=0:A3=0 <133>
42 IFA1<>0THENPOKEA,32 <254>
43 A=A+A1:AY=AY+A2:AX=AX+A3 <76>
44 IFPEEK(A)=15THENPOKES+3,242:FORT=1TOD
0:NEXT:POKES+3,0:SC=SC+1:U=U+1 <182>
45 IFPEEK(A)>15ANDPEEK(A)<20THENU=U+1:GO
SUB63 <235>
46 IFPEEK(A)=14THEN72 <212>
47 IFU=185THENPOKEA+F,0:POKEA,A4:R=0:U=0
:0=0-1:GOSUB81:GOTO13 <33>
48 POKEA+F,1:POKEA,A4 <235>
49 PRINT"(HOME DOWN4 RIGHTS CYAN RVSON)"
SC <23>
50 IFSC>HITHEHHI=SC <25>
51 PRINT"(HOME DOWN4 CYAN RVSON)"SPC(14)
HI <197>
52 R=R+1:IFR>OTHERN=0:GOTO54 <87>
53 GOT034 <192>
54 POKEB+F,D1:POKEB,D2 <131>
55 IFBX>AXTHENBX=BX-1:B=B-1 <232>
56 IFBX<AXTHENBX=BX+1:B=B+1 <209>
57 IFBY>AYTHENBY=BY-1:B=B-2 <54>
58 IFBY<AYTHENBY=BY+1:B=B+2 <31>
59 D1=PEEK(B+F):D2=PEEK(B) <249>
60 IFA=BTHEN72 <82>
61 POKEB+F,3:POKEB,14 <95>
62 GOT034 <201>
63 IFPEEK(A)=19THEN67 <58>
64 SC=SC+(PEEK(A)-15)*10 <210>
65 POKEA,PEEK(A)+5 <128>
66 GOT068 <223>
67 V=(INT(RND(1)*5)+1)*10:SC=SC+V:POKEA,
(V/10)+20 <95>
68 POKEA+F,1 <117>
69 FORT=220T0245STEP.2:POKES+2,T:NEXT <18>
70 POKES+2,0 <69>
71 RETURN <213>
72 POKEA+F,1:POKEA,26 <118>
73 FORT=250T0210STEP-.2:POKES+1,T:NEXT <101>
74 POKES+1,0 <69>
75 SC=0:A4=0 <78>
76 FORT=1T02000:NEXT <9>
77 PRINT"(CLEAR DOWN3 RIGHT3 CYAN)F(PURP
LE)G(GREEN)H(RED)I(CYAN)J(PURPLE)K(GREEN)
G(RED)H(RIGHT CYAN)G(PURPLE)L(GREEN)M(R
ED)M(CYAN)H(PURPLE)G" <26>
78 PRINT"(DOWN4 WHITE)"SPC(8)"^(RVSON)"
HI:PRINT"(DOWN4)"SPC(5)"(RVSON)PRESS(SPA
CE2)FIRE" <57>
79 WAIT37137,32,32 <200>
80 U=0:0=4:GOTO13 <96>
81 READQ,W <90>
82 IFW=-1THENRESTORE:RETURN <206>
83 POKES+1,0 <20>
84 POKES+2,0 <25>
85 FORT=1TOW:NEXT <60>
86 POKES+1,0 <81>
87 POKES+2,0 <86>
88 FORT=1T020:NEXT <109>
89 GOT081 <229>
90 DATA15,050,215,150,215,100,225,400 <226>
91 DATA225,400,228,400,228,400,235,600 <209>
92 DATA231,200,225,200,000,100,235,100 <183>
93 DATA231,300,225,100,219,400,232,800 <91>
94 DATA228,300,223,100,225,900,000,-1 <166>

```

ENDE DES LISTINGS



## SCRUMBLE VC-20

Das Spiel "Scrumble" ist, wie man schon am Namen vermuten kann, dem Spielhallenhit "Scramble" nachempfunden. Sie fliegen mit einem Raumschiff durch eine Höhle und müssen den Hindernissen ausweichen. Außerdem tauchen am linken Höhlenrand ab und zu Raketen und Treib-

stofftanks auf, die abgeschossen werden müssen. Sie können sowohl Laserschüsse abfeuern, wie auch Bomben werfen. Wenn ein Laserschuss unterwegs ist, können Sie durch nochmaliges Drücken des Feuerknopfes eine Bombe werfen und umkehrt.

Folgende Spielstufen können Sie erreichen.:

1. die einfache Höhle
2. die Höhle mit Kometen (können nicht abgeschossen werden)
3. die Höhle mit UFOs (kann man abschießen)
4. nochmal Kometen

Hat man alle Spielstufen durchlaufen, kommt zusätzlich zu den Kometen noch das MYSTERY. Es erscheint 5 mal. Es muß mit dem Laser oder einer Bombe getroffen oder aber gerammt werden. Hat man das nicht geschafft,

hat man seine Mission nicht erfüllt.

### Eingabehinweise:

Tippen Sie das Vorprogramm ab, prüfen Sie es und speichern es. Starten Sie es nun. Wenn "LOAD/PRESS PLAY ON TAPE" erscheint, stoppen Sie es durch Drücken der "RUN-STOP / RE-STORE"-Taste. Löschen Sie den Basicteil mit "NEW" und tippen nun das Hauptprogramm ab.

Speichern Sie es ab, wenn Sie es überprüft haben.

Die Steuerung erfolgt mit dem Joystick.

## Teil 1



### VC20 LISTING + CHECKSUMMEN (VC V1.0)

```

0 REM. SCRUMBLE. <11>
1 REM. MARTIN RIEDL. <27>
2 REM. BAHNHOFSTR. 23. <122>
3 REM. 6970 LAUDA. <91>
4 REM. TEL (09343/2812). <73>
5 REM. (C) 1985 FUER TRONIC VERLAG. <239>
10 PRINT "CLEAR WHITE": CLR <254>
20 POKE36879,110:PRINT "{DOWN10 RIGHT4}BI
TTE{SPACE}WARTEN" <90>
30 FORI=0TO511:POKE7168+I,PEEK(32768+I):
NEXT: <236>
40 FORA=0TO111 <58>
50 READB:POKE7168+A,B:NEXT <72>
100 POKE36869,242:PRINT "CLEAR-LIEGEN{S
PACE}VIE{SPACE}DURCH{SPACE}DAS{DOWN SPAC
E}IDEHLENSYSTEM." <97>
110 PRINT "{DOWN}IERUEHREN{SPACE}VIE{SPAC
E}NICHT{SPACE3 DOWN}DIE{SPACE}GAENDE.{SP
ACE}VCHIESSEN{SPACE DOWN}VIE{SPACE}DIE{S
PACE}" <234>
125 PRINT "{DOWN}IREIBSTOFFTANKS{SPACE7 D
OWN}AKETEN{SPACE}UND{SPACE}FOS{SPACE}A
B.":GOSUB225 <107>
126 PRINT "CLEAR-IE{SPACE}FOS{SPACE}KO
ENNEN{SPACE}VIE{SPACE DOWN SPACE}ABSCHIE
SSEN,DIE{SPACE}OMETN{DOWN}NICHT." <164>
127 PRINT "{DOWN}M{SPACE}NDE{SPACE}KOMM
T{SPACE}DAS{SPACE3 DOWN}YSTERY.IREFFEN{
SPACE}VIE{SPACE}ES{DOWN}HABEN{SPACE}VIE{
SPACE}IHRE{SPACE}UFGABE{DOWN}ERFUELLT <2>
128 GOSUB225 <219>
130 PRINT "CLEAR}TEUERN{SPACE}VIE{SPACE
}IHR{SPACE}AUM-{SPACE DOWN}SCHIFF{SPACE
}MIT{SPACE}DEM{SPACE}OY-{SPACE3 DOWN}ST
ICK. <151>
140 PRINT "{DOWN}VIE{SPACE}KOENNEN{SPACE}
SCHIESSEN{SPACE DOWN}UND{SPACE}BOMBEN.":
GOSUB225 <177>
150 POKE36869,255:PRINT "CLEAR" <152>
160 PRINT "@{DOWN LEFT}A{SPACE}={SPACE RV
SON}DEIN{SPACE}RAUMSCHIFF{RVSOFF}" <97>
170 PRINT "{DOWN}C{SPACE}={SPACE RVSON}TA
NK{SPACE} (10{SPACE}PTS) {RVSOFF SPACE4 DO
WN SPACE}F{SPACE}={SPACE RVSON}RAKETE{SP
ACE} (10{SPACE}PTS) {RVSOFF}" <146>

```

```

180 PRINT "{DOWN}B{SPACE}={SPACE RVSON}UF
D{SPACE} (10{SPACE}PTS) {RVSOFF}" <233>
185 PRINT "{DOWN}D{SPACE}={SPACE RVSON}KO
MET{RVSOFF}" <217>
190 PRINT "{DOWN}M{SPACE}={SPACE RVSON}MY
STERY{SPACE} (50{SPACE}PTS) {RVSOFF DOWN}"
:GOSUB225 <88>
224 POKE36869,240:POKE198,3:POKE632,147:
POKE633,131:END <222>
225 PRINT "{DOWN RIGHT6 RVSON}<RETURN>{RV
SOFF}" <184>
230 GETA$:IFA$<>" THENRETURN <59>
240 GOTO230 <54>
900 END <7>
990 DATA129,153,153,90,60,126,255,255 <170>
1000 DATA255,195,126,60,36,36,36,0 <216>
1010 DATA60,126,195,255,255,126,24,126 <146>
1020 DATA0,24,188,254,254,188,24,0 <102>
1030 DATA0,56,124,254,254,84,40,16 <71>
1040 DATA0,0,100,254,254,100,0,0 <117>
1050 DATA0,136,254,127,254,136,0,0 <93>
1060 DATA36,36,36,36,36,36,36,36 <3>
1070 DATA192,192,240,252,252,254,224,128 <202>
1080 DATA1,7,127,63,63,15,3,3 <179>
1090 DATA255,255,255,255,255,255,255,255 <53>
1100 DATA0,36,144,136,162,136,2,0 <141>
1110 DATA0,0,0,0,0,0,0,0 <49>
1120 DATA128,252,130,238,238,130,252,128 <221>
ENDE DES LISTINGS

```

## Teil 2

### VC20 LISTING + CHECKSUMMEN (VC V1.0)

```

1 REM . SCRUMBLE HAUPTPROGRAMM . <174>
2 REM . (C) 1985 BY MARTIN RIEDL . <215>
10 POKES5,0:POKES6,28:CLR:GOSUB9000 <252>
160 PRINT "CLEAR":POKE36879,25:POKE3686
9,255 <13>
170 PP=0 <112>
190 PRINT "CLEAR DOWN23":POKE36879,25:R
EM 23 RUNTER <227>
200 A=8169:B=8180:C=7844:L=30720:T2=12:T

```

# programme

```

3=1
219 REM HAUPTPROGRAMM
220 POKEC,1:POKEC-22,0
230 PP=PP+1
240 IFPP=100THENB=B-1
250 IFPP=300THENA=A+1
260 IFPP=500THENB=B-1
270 IFPP=700THENA=A+1
280 D=INT (B*RND (1))
290 IFD=PRINT#:POKEA+1+L,0:GOTO388
383 IFW=6THENPOKER-1,9:POKEB-1+L,0
384 K=INT (RND (1)*6)+1
385 IFK=1THENPOKEA+1,6:POKEA+1+L,3:GOTO3
88
45634 6THENPOKEA+1,3:POKEA+1+L,4:GOTO388
387 IFK=3THENT1=INT (RND (1)*B-1)-(A+1)
)+1:POKEA+T1,T2:POKEA+T1+D,T1:GOTO388END
IFPP=750ORPP=775ORPP=800ORPE=8025ORPP=80
50THENPOKE3&9,298POKEA+1,13:POKEA+1+L,
7
389 PRINT
390 IFA=8164TAB (A=A+1:B=B+1
400 DATAB=8185THENA=A-1:B=-1
410 GOSUB1000
440 C=C+Y
450 IFPEEK (TAB (IFPEEK (C)=4THEN10010
457 IFPEEK (C)=13ORPEEK (C-22)=13THEN10000
460 IFPEEK (C-22)=10ORPEEK (C-22)=8ORPEEK (
C-22)=9ORPEEK (C-22)=6ORPEEK (C-22)=3THEN1
0010
465 IFPEEK (C-22)=2ORPEEK (C-22)=(4THEN0001
0
470 POKEC,1:POKEC+L,2:POKEC-22,0:POKEC-2
2+L,2:Y=0
480 IFZ>0THENGOSUB2000
490 IFG>0THENGOSUB3000
590 GOTO230
615 REM KAPUTT!
620 FORT=0T02000:NEXT
622 PP=PP+SC
625 POKE36879,127:PRINT" (BLUE DOWN6 SPAC
E3) SCORE (SPACE):";PP:IFPP>HPTHENHP=PP
630 PRINT" (DOWN3) HIGHSCORE (SPACE):";HP
640 PRINT" (DOWN3 RIGHT3) NOCH (SPACE) MAL? (
SPACE) (J/N)"
650 BETA#
660 IFA#="N"THENPRINT" (CLEAR)":NEW:END
670 IFA#="J"THENSC=0:Z=0:G=0:GOTO160
680 GOTO650
730 PRINT" (CLEAR BLUE)":POKE36879,27:POK
E36869,240:END
799 REM JOYSTICKABFRAGE
1000 SYS828:IFPEEK (781)=0THENY=0:RETURN
1010 IFPEEK (781)=4THENY=Y-1
1020 IFPEEK (781)=32THENY=Y+1
1030 IFZ>0THEN1050
1040 IFPEEK (781)=8THENZ=1:S=C-1
1050 IFG>0THENRETURN
1060 IFPEEK (781)=8THENG=1:F=C+22:RETURN
<197>
<151>
<244>
<15>
<13>
<1>
<54>
<42>
<19>
<155>
<56>
<133>
<210>
<76>
<241>
<31>
<231>
<252>
<202>
<142>
<235>
<140>
<96>
<247>
<36>
<242>
<221>
<150>
<134>
<39>
<159>
<60>
<211>
<163>
<27>
<77>
<165>
<254>
<166>
<136>
<34>
<136>
<85>
<45>
<218>
<161>
<57>
1070 RETURN
1999 REM BOMBE (POKEN UND AUF TREFFER UE
BERPRUEFEN)
2000 POKES-21,32
2050 IFPEEK (S)=3THENPOKES,11:POKES+L,2:Z
=0:SC=SC+10:GOSUB2200:POKES,32:RETURN
<110>
2060 IFPEEK (S)=6THENPOKES,11:POKES+L,2:Z
=0:SC=SC+10:GOSUB2200:POKES,32:RETURN
<141>
2065 IFPEEK (S)=2THENPOKES,11:POKES+L,2:Z
=0:SC=SC+10:GOSUB2200:POKES,32:RETURN
<118>
2067 IFPEEK (S)=13THENPOKES,11:POKES+L,2:
Z=0:SC=SC+50:GOSUB2200:GOTO10000
<26>
2070 IFPEEK (S)=8THENZ=0:RETURN
<198>
2080 IFPEEK (S)=10THENZ=0:RETURN
<202>
2085 IFPEEK (S)=4THENZ=0:RETURN
<185>
2087 POKES,5:POKES+L,5
<176>
2090 S=S-1:Z=1
<158>
2100 RETURN
<202>
2200 POKE36878,15:POKE36877,230:FORT=1T0
20:NEXT:POKE36878,0:POKE36877,0
<114>
2210 RETURN
<56>
2999 REM LASER (POKEN UND AUF TREFFER UE
BERPRUEFEN)
3000 POKEF-44,32
3050 IFPEEK (F)=3ORPEEK (F)=6ORPEEK (F)=2OR
PEEK (F)=13THENGOSUB2200:GOTO4000
<124>
3055 IFPEEK (F)=13ORPEEK (F+22)=13THENSC=S
C+50:GOSUB2200:GOTO10000
<20>
3060 IFPEEK (F+22)=6ORPEEK (F+22)=3ORPEEK (
F+22)=2THENGOSUB2200:GOTO4010
<246>
3070 IFPEEK (F)=8ORPEEK (F)=9ORPEEK (F+22)=
8ORPEEK (F+22)=9THENG=0:RETURN
<206>
3080 IFPEEK (F)=10ORPEEK (F+22)=10ORPEEK (F
)=4ORPEEK (F+22)=4THENG=0:RETURN
<162>
3085 POKEF,7:POKEF+L,5
<45>
3090 F=F+22:G=1
<98>
3095 IFF>8010THENPOKEF-22,32:G=0:RETURN
<90>
3100 RETURN
<182>
3999 REM TREFFER
4000 POKEF,11:POKEF+L,2:G=0:SC=SC+10:FOR
T=1T010:NEXT:POKEF,32:RETURN
<132>
4010 POKEF+22,11:POKEF+22+L,2:G=0:SC=SC+
10:FORT=1T010:NEXT:POKEF+22,32:RETURN
<128>
8999 REM JOYSTICK - DATEN EINLESEN
<209>
9000 DATA24,169,0,237,17,145,41,127,141,
13,3,169,0,141,34,145,24,237,32,145
<221>
9010 DATA24,109,13,3,74,74,170,169,255,1
41,34,145,96
<155>
9030 FORT=0T032:READI:POKE828+T,I:NEXT
<236>
9050 RETURN
<11>
10000 POKE36869,240:PRINT" (CLEAR RIGHT2)
MISSION (SPACE) COMPLETED":GOTO622
<112>
10010 POKE36869,240:PRINT" (CLEAR RIGHT3)
LEIDER (SPACE) VERLOREN":GOTO622
<156>
10020 POKE36869,240:PRINT" (CLEAR) SIE (SPA
CE) HABEN (SPACE) DAS (SPACE) MYSTERY (DOWN SP
ACE) LEIDER (SPACE) NICHT (SPACE) ERWISCHT":G
OTO622
<61>
ENDE DES LISTINGS

```

## LAUSCHRIFT mal anders

WIG-20

Bei diesem Programm laufen zwei Schriftbänder über den Bildschirm. Das erste Schriftband von links nach rechts, das zweite allerdings von links nach rechts und zwar in Spiegelschrift. Das einmalige Umschreiben des Zeichensatzes dauert etwa 47 Sekunden.

Über zwei INPUT-Befehle kann dann ein String bis zu 170 Zeichen eingegeben werden.

## VC20 LISTING + CHECKSUMMEN (VC V1.0)

```

1 IFPEEK(5120)=28ANDPEEK(5121)=34THEN14 <39>
2 POKE55,19:POKE56,19:CLR <73>
3 CH=32768:FORL=5120TO5672:POKEL,PEEK(CH)
  :CH=CH+1:NEXT <1>
4 CH=32768:FORL=6144TO6648:A=PEEK(CH):CH
  =CH+1:B=0 <212>
5 IFINT(A/2)*2<ATHENB=B+128:A=A-1 <51>
6 IFINT(A/4)*4<ATHENB=B+64:A=A-2 <113>
7 IFINT(A/8)*8<ATHENB=B+32:A=A-4 <135>
8 IFINT(A/16)*16<ATHENB=B+16:A=A-8 <221>
9 IFINT(A/32)*32<ATHENB=B+8:A=A-16 <127>
10 IFINT(A/64)*64<ATHENB=B+4:A=A-32 <82>
11 IFINT(A/128)*128<ATHENB=B+2:A=A-64 <104>
12 IFINT(A/256)*256<ATHENB=B+1 <216>
13 POKEL,B:NEXT <246>
14 POKE36879,24:PRINT"(CLEAR BLACK DOWN)
TEXT?(SPACE2)PRO(SPACE)STRING(SPACE)MAX.
  
```

```

(SPACES DOWN)85(SPACE)ZEICHEN(SPACE)!(DO
MN)":C$="(SPACE)" <212>
15 INPUTA$:INPUT"(DOWN)";B$:IFB$=""THENC
  $="" <237>
16 A$=A$+C$+B$+"(SPACE)*(SPACE)" <156>
17 N=LEN(A$):IFN<22THENA$=A$+A$:GOTO17 <157>
18 O=LEN(A$):FORT=OTO1STEP-1:D$=D$+MID$(
  A$,T,1):NEXT:D$=RIGHT$(D$,22)+D$ <44>
19 A$=A$+LEFT$(A$,22) <211>
20 POKE36879,B:POKE36869,253 <81>
21 PRINT"(CLEAR WHITE DOWN)
(DOWN)" <234>
22 PRINT" (BLUE)" <46>
23 FORL=1TO10:PRINT"(HOME DOWN2)"MID$(A$,
  L,22)"(RVSON DOWN3)"MID$(D$,N+1-L,22) <107>
24 FORM=1TO50:NEXTM:NEXT:GOTO23 <87>
ENDE DES LISTINGS
  
```

# RAILWAY — Einmal Agent sein!



In dem Spiel Railway dreht sich alles um die lieben Agenten. Die Geheimdienste KMA (Karchen's Möchregern-Agenten) und STG (Spezial-technischer Geheimdienst) sind arg verfeindet. Soeben hat der STG einen seiner Spione aus dem gegnerischen Lager befreit und in die Leitstelle, die als Zug getarnt ist, gebracht. Doch schon gibt es neue Probleme. Die Agenten des KMA sind auf Zack und verstellen die Weichen, so daß der Zug zu entgleisen droht. Nun trete ich als Agent des STG in Aktion. Meine Aufgabe ist es, die Weichen von 1-9 mit den Tasten von 1-9 so umzustellen, daß der Zug nicht entgleist. Hat man 200 Punkte erreicht, erscheint unten im Screen eine Verbindung auf das alleinstehende Gleis, wo man die Lok nun hinbugsieren muß. Hat man das geschafft, geht das Spiel ein Level weiter, und wieder heißt es den Zug vor dem entgleisen bewahren.

## Teil 1

### C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

1 REM ----- <33>
2 REM -- RAILWAY 1 -- <196>
3 REM ----- <35>
4 REM -- WRITTEN FOR THE C-16 BY : -- <224>
5 REM -- CARSTEN REINKE -- <167>
6 REM -- 1000 BERLIN 21 -- <89>
7 REM -- <12>
8 REM --ALLE RECHTE(86) BEI EQUUS-SOFT-- <116>
9 REM ----- <41>
10 COLOR0,2,1:COLOR4,2,1:COLOR1,5,5:POKE
  
```

```

52,55:POKE56,55:POKE2022,0:SCNCLR:VOL3 <234>
20 FORA=1TO14:READA$:PRINTTAB(10);A$:NEX
  TA <233>
30 CHAR1,06,22,"(YELLOW RVSON SPACE2)ALL
  E(SPACE)RECHTE(SPACE)BEI(SPACE)CR(1986)(
  SPACE2 RVSOFF)" <122>
40 DATA"(DOWN3)" <135>
50 DATA"(SPACE) (SHIFTSPACE) (
  SHIFTSPACE2 SPACE)" <226>
60 DATA"(SPACE) | | (SHIFTSPACE) | | | | |
  (SHIFTSPACE3)" <201>
70 DATA"(SPACE) | | | | | (SHIFTSPACE
  3)" <210>
80 DATA"(SPACE) | | | | | (SHIFTSPACE
  3)" <250>
90 DATA"(SPACE) | | (SHIFTSPACE) | | | | (SHIFTS
  
```



```

PAGE3| | | | | " <218>
100 DATA" (SPACE) (SPACE) (SHIFTSPACE) <80>
3 0 0 " <237>
110 DATA" (CL.BLU DOHN3)" <79>
120 DATA" (SHIFTSPACE) (SHIFTSPACE) (SHIFTSPACE) <185>
(SHIFTSPACE2) " <163>
130 DATA" | | (SHIFTSPACE) | | (SHIFTSPACE) | | | <161>
| | | | | " <21>
140 DATA" | | (SHIFTSPACE) | | (SHIFTSPACE) | | | <66>
| | | | | " <9>
150 DATA" | | (SHIFTSPACE) | | (SHIFTSPACE) | | | <118>
| | | | | " <170>
160 DATA" | | | | | (SHIFTSPACE) | | (SPACE2) <161>
3 | | | " <234>
170 DATA" | | | | | (SHIFTSPACE) (SHIFTS <239>
PAGE2) " <158>
180 TRAP530:FE=0:FORA=832T0849:READB$:FE <139>
=FE+DEC(B$):POKEA,DEC(B$) <210>
185 SO=SO+6:SOUND1,SO,4:SOUND2,SO+1,4:NE <96>
XTA <232>
190 IFFE<>2131THENGOTO530:ELSE SYS832 <93>
200 REM -----ZEICHEN-ROM KOPIEREN----- <223>
210 DATA2,00,00,00,00,00,00,00,38,00,00,D1 <178>
,9D,00,39,CA,D0,F1,60 <4>
220 TRAP540:FE=0:FORA=14552T014671:READB <32>
:FE=FE+B:POKEA,B <186>
225 SO=SO+6:SOUND1,SO,4:SOUND2,SO+1,4:NE <139>
XTA <132>
230 IF FE<>10796THENPRINT"(CLEAR)FEHLER: <106>
":LIST240-360:END <149>
240 REM ----DATAS FUER NEUE ZEICHEN----- <138>
250 DATA0,15,24,48,104,68,67,66 <104>
260 DATA66,67,68,104,48,24,15,0 <32>
270 DATA0,240,24,12,22,34,194,66 <46>
280 DATA66,194,34,22,12,24,240,0 <52>
290 DATA68,255,68,68,68,68,255,68 <79>
300 DATA0,0,0,0,0,0,0,0 <73>
310 DATA66,255,66,66,66,255,66,66 <221>
320 DATA102,102,102,0,0,0,0,0 <228>
330 DATA248,252,226,255,255,36,255,36 <72>
340 DATA31,63,71,255,255,36,255,36 <120>
350 DATA24,60,126,102,102,126,126,126 <123>
360 DATA126,126,126,102,102,126,60,24 <132>
370 DATA52,110,173,255,119,255,122,52 <93>
380 DATA60,60,52,60,60,60,63,90 <153>
390 DATA66,255,66,66,66,66,255,66 <173>
400 REM <79>
410 REM NAECHSTES PRG-TEIL LADEN <100>
420 REM <100>
430 TRAP 550 <100>
440 PRINT:PRINT"(DOWN WHITE SPACE3)WOLLE <73>
N(SPACE)SIE(SPACE)ANWEISUNGEN(SPACE)(J/N <221>
)(SHIFTSPACE)?" <228>
450 GETKEYA$:IFA$<>"J"ANDA$<>"N"THEN450 <72>
460 SCNCLR:PRINT"(DOWN2)NEW" <120>
470 IFA$="J"THENA$="RAILWAY2":ELSEA$="RA <123>
ILWAY3" <132>
480 PRINT"(DOWN2)LOAD";CHR$(34);A$;CHR$( <93>
34);"(HOME)"; <153>
490 POKE239,5:POKE1319,13:POKE1320,13:PO <173>
KE1321,82:POKE1322,213:POKE1323,13:END <79>
500 REM <100>
510 REM FEHLERABFRAGEN UND ANZEIGEN <100>
520 REM <100>
530 AN=200:EN=210::GOSUB560:GOTO600 <100>
540 AN=240:EN=390:GOSUB560:GOTO600 <100>
550 AN=1:EN=610:GOSUB560:GOTO610 <100>
560 PRINT"(CLEAR SPACE)ES(SPACE)BIGT(SPA <148>
CE)EINEN(SPACE)FEHLER(SPACE)DER(SPACE)IN <166>
(SPACE)ZEILE"EL <249>
570 PRINT"(SPACE)ODER(SPACE)ZEILE(SPACE) <222>
"AN"-EN(SPACE)LIEGT." <37>
580 POKE2022,2 <228>
590 RETURN <37>
600 PRINT"(CLEAR DOWN2)LIST"EL"-EN"(HOW <228>
E)";:POKE239,1:POKE1319,13:POKE2022,0 <37>
610 END <228>

```

ENDE DES LISTINGS

## Teil 2

C-16 LISTING + CHECKSUMMEN (DC V1.0)

```

1 REM ----- <33>
2 REM --- R A I L W A Y 2 --- <207>
3 REM ----- <35>
4 REM --- WRITTEN FOR THE C-16 BY : --- <224>
5 REM --- CARSTEN REINKE --- <167>
6 REM --- 1000 BERLIN 21 --- <89>
7 REM --- <12>
8 REM ---ALLE RECHTE (86) BEI EQUUS-SOFT--- <116>
9 REM ----- <41>
10 VOL4:SCNCLR <155>
20 COLOR0,14,6:COLOR1,1,0:COLOR4,15,3 <53>
30 CHAR1,10,11,"EQUES-SOFT(SPACE)PRESENT <249>
S" <58>
40 CHAR1,13,13,CHR$(130)+"R(SPACE)A(SPAC <222>
E)I(SPACE)L(SPACE)W(SPACE)A(SPACE)Y" <11>
50 CHAR1,11,15,"(FLASHOFF)BY(SPACE)CARST <46>
EN(SPACE)REINKE" <89>
60 FORA=1T05000STEP25:SOUND1,A/5,2:SOUND <52>
2,A/5+2,2:PRINTCHR$(14);:NEXTA <235>
70 SCNCLR <223>
80 PRINT"(DOWN SPACE2)IERZLICH(SPACE)OI <143>
LLKOMMEN(SPACE)ZU(SPACE)_AILWAY(SPACE)! <97>
!" <102>
90 PRINT"(DOWN SPACE2)-EINE(SPACE)AUFGAB <83>
E(SPACE)IST(SPACE)ES,(SPACE)DIE(SPACE)OE <178>
ICHEN(SPACE)SO" <136>
100 PRINT"(DOWN SPACE2)ZU(SPACE)STELLEN, <110>
(SPACE)DASS(SPACE)DIE(SPACE)LOKOMOTIVEN( <119>
SPACE)DIE" <119>
110 PRINT"(DOWN SPACE2)|LEISE(SPACE)NICH <119>
T(SPACE)VERLASSEN." <79>
120 PRINT"(DOWN SPACE2)-U(SPACE)STARTEST <230>
(SPACE)MIT(SPACE)EINER(SPACE)LOK,(SPACE) <230>
UND(SPACE)JE" <230>
130 PRINT"(DOWN SPACE2)LAENGER(SPACE)DU( <97>
SPACE)DIE(SPACE)LOKS(SPACE)AUF(SPACE)DEN <102>
(SPACE)CHIENEN" <83>
140 PRINT"(DOWN SPACE2)HAELST,(SPACE)DES <178>
TO(SPACE)MEHR(SPACE)UNKTE(SPACE)BEKOMMS <136>
T" <110>
150 PRINT"(DOWN SPACE2)DU." <119>
160 CHAR1,7,23,"|ITTE(SPACE)DRUECKE(SPAC <119>
E)EINE(SPACE)ASTE" <79>
170 POKE239,0:WAIT239,1:POKE2022,4:SCNCL <230>
R:POKE2022,0 <94>
180 PRINT"(DOWN SPACE2)-MMER(SPACE)WENN( <230>
SPACE)DU(SPACE)200(SPACE)UNKTE(SPACE)ER <230>
REICHT" <230>
190 PRINT"(DOWN SPACE2)HAST,(SPACE)MUST( <230>
SPACE)DU(SPACE)1(SPACE)LOK(SPACE)IN(SPAC <230>
E)DEN(SPACE)IAHNHOF" <230>
200 PRINT"(DOWN SPACE2)(UNTER(SPACE)DEM( <230>
SPACE)C.REINKE)(SPACE)LEITEN.-ANN" <230>
210 PRINT"(DOWN SPACE2)SPRINGT(SPACE)DAS <230>
SPACE)PIEL(SPACE)IN(SPACE)DAS(SPACE)NA <230>
ECHSTE" <230>
220 PRINT"(DOWN SPACE2)LEVEL." <230>
230 PRINT"(DOWN SPACE2)|EI(SPACE)EINEM(S <230>
PACE)_NFALL(SPACE)WERDEN(SPACE)-IR(SPACE <230>
)20" <230>
240 PRINT"(DOWN SPACE2)UNKTE(SPACE)ABGE <0>
ZOGEN.(SPACE)/ACH(SPACE)5(SPACE)_NFAELLE <239>
N" <83>
250 PRINT"(DOWN SPACE2)IST(SPACE)DAS(SPA <239>
CE)PIEL(SPACE)ZU(SPACE)-NDE." <239>
260 PRINT"(DOWN SPACE2)TEUERUNG(SPACE): <239>
SPACE|ENUTZE(SPACE)DIE(SPACE)|ASTEN(SP <239>
ACE)1-9(SPACE)!" <239>
270 CHAR1,7,23,"|ITTE(SPACE)DRUECKE(SPAC <239>

```

```

E)EINE{SPACE}|ASTE"
280 POKE239,0:WAIT239,1:PRINTCHR$(142)
290 PRINT"CLEAR DOWN2"NEW"
300 PRINT"DOWN2"LOAD";CHR$(34);"RAILWAY
3";CHR$(34);"HOME";
310 FORA=0TO4:READB:POKEA+1319,B:NEXTA:P
OKE239,5
320 DATA13,13,82,213,13
ENDE DES LISTINGS
    
```

```

)=-1:GOTO460
360 GOTO480
370 IFYR(A)=1THENIFP<>33ANDP<>28ANDP<>30
THENGOSUB730
380 IFYR(A)=1THENIFP=28THENXR(A)=1:YR(A)
=0:GOTO440
390 IFYR(A)=1THENIFP=30THENXR(A)=-1:YR(A)
)=0:GOTO450
400 IFYR(A)=-1THENIFP<>33ANDP<>27ANDP<>2
9THENGOSUB730
410 IFYR(A)=-1THENIFP=27THENXR(A)=1:YR(A)
)=0:GOTO440
420 IFYR(A)=-1THENIFP=29THENXR(A)=-1:YR(A)
)=0:GOTO450
430 GOTO480
440 PL$(A)="#":GOTO480
450 PL$(A)="$":GOTO480
460 PL$(A)="%":GOTO480
470 PL$(A)="$"
480 PW=XE+40*YE:POKE3072+PW,AL(A):POKE20
48+PW,77:CHAR1,XP(A),YP(A),PL$(A):AL(A)=
P
490 GOTO1290
500 SOUND3,0,0:SOUND3,900,20:GOTO520
510 NEXTA:GOTO140
520 GETE$:E=ASC(E$)-48:IFE<10RE>9THEN510
:ELSEEF=1
530 P=PEEK(3072+WX(E)+40*WY(E)):PW=3072+
WX(E)+40*WY(E)
540 ONEGOTO560,560,600,560,650,690,690,6
50,650
550 GOTO510
560 IFP=29THENPOKEPW,31
570 IFP=31THENPOKEPW,27
580 IFP=27THENPOKEPW,29
590 GOTO510
600 IFP<31THEN630
610 IFP=31THENPS=33:GOTO640
620 IFP=33THENPS=27:GOTO640
630 PS=P+1
640 POKEPW,PS:GOTO510
650 IFP=30THENPOKEPW,31
660 IFP=31THENPOKEPW,28
670 IFP=28THENPOKEPW,30
680 GOTO510
690 IFP=30THENPOKEPW,33
700 IFP=33THENPOKEPW,29
710 IFP=29THENPOKEPW,30
720 GOTO510
730 POKE3072+XE+40*YE,AL(A):POKE2048+XE+
40*YE,77
740 XP(A)=10:YP(A)=2:XR(A)=1:YR(A)=0:PL$
(A)="#":
750 AL(A)=31:XE=10:YE=2:P=31:IFGL=0THENS
C=SC-20:IFSC<0THENS C=0
760 CH=CH+1:IFCH<6THENCHAR1,33,24,"":PRI
NTUSING"YELLOW###";CH:COLOR1,12,7
770 SOUND1,516,10:SOUND1,453,10:SOUND1,3
83,10:SOUND1,516,20:SOUND1,345,10
780 IFCH<5THENRETURN:ELSESCNCLR
790 CHAR1,10,10,"YELLOW RVSON SPACE}G{S
PACE}A{SPACE}M{SPACE}E{SPACE2}D{SPACE}V{
SPACE}E{SPACE}R{SPACE}RVSOFF"
800 CHAR1,6,12,"DU{SPACE}HAST"+STR$(SC+Q
S+LE*500)+"{SPACE}PUNKTE{SPACE}ERREICHT.
"
810 CHAR1,4,14,"DU{SPACE}BIST{SPACE}IM{S
PACE}LEVEL"+STR$(LE)+"{SPACE}GESCHEITERT
"
820 CHAR1,3,17,"MOECHTEST{SPACE}DU{SPACE
}WEITERSPIELEN{SPACE}?{SPACE}J/N"
830 GETKEYA$:IFA$<>"J"ANDAS$<>"N"THENB30
840 IFA$="J"THENRUN60
850 END
860 COLOR1,14,4
870 CHAR1,2,2,"BLUE}[
    
```

## Teil 3

### C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

1 REM -----<250>
2 REM -- R A I L W A Y 3 --<218>
3 REM -----<252>
4 REM -- WRITTEN FOR THE C-16 BY : --<224>
5 REM -- CARSTEN REINKE ---<167>
6 REM -- 1000 BERLIN 21 ---<89>
7 REM -- ---<12>
8 REM --ALLE RECHTE(86) BY EQUUS-SOFT--<246>
9 REM -----<2>
10 COLOR0,1:COLOR4,1<170>
20 POKE65298,192:POKE65299,56:VOL2<106>
30 LE=1:SCNCLR:GOSUB860<130>
40 CHAR1,3,24,"MOECHTEST{SPACE}DU{SPACE}
JETZT{SPACE}SPIELEN{SPACE}?{SPACE}J"
50 GETKEYA$:IFA$<>"J"THENEND<226>
60 CLR:LE=1<228>
70 DIMWX(9),WY(9),XR(5),YR(5),XP(5),YP(5)
,AL(5):GOSUB1270:PUDEF"0"
80 COLOR0,1:COLOR4,1:COLOR1,3,4:SCNCLR<110>
90 R$=")))))))))"
)))":CHAR1,0,0,R$
100 FORA=1TO22:CHAR1,0,A,"{SPACE38}":N
EXTA:CHAR1,0,23,R$
110 CHAR1,2,24,"YELLOW}LEVEL:00":CHAR1,
14,24,"SCORE:0000":CHAR1,26,24,"CRASHS:0
0"
120 GOSUB860:CHAR1,8,24,"YELLOW":PRINT
USING"##";LE:
130 TI$="000000"
140 COLOR1,12,7:FORA=1TOLE:XE=XP(A):YE=Y
P(A):XP(A)=XP(A)+XR(A):YP(A)=YP(A)+YR(A)
150 SOUND3,0,0:SOUND3,700,20
160 P=PEEK(3072+XP(A)+40*YP(A)):IFP=40TH
EN1360
170 ONAL(A)-26GOTO180,210,240,270,300,86
0,370
180 IFXR(A)=1THENIFP<>31THENGOSUB730
190 IFYR(A)=1THENIFP<>33THENGOSUB730
200 GOTO480
210 IFXR(A)=1THENIFP<>31THENGOSUB730
220 IFYR(A)=-1THENIFP<>33THENGOSUB730
230 GOTO480
240 IFXR(A)=-1THENIFP<>31THENGOSUB730
250 IFYR(A)=1THENIFP<>33THENGOSUB730
260 GOTO480
270 IFXR(A)=-1THENIFP<>31THENGOSUB730
280 IFYR(A)=-1THENIFP<>33THENGOSUB730
290 GOTO480
300 IFXR(A)=1THENIFP<>31ANDP<>29ANDP<>30
THENGOSUB730
310 IFXR(A)=1THENIFP=29THENXR(A)=0:YR(A)
=1:GOTO470
320 IFXR(A)=1THENIFP=30THENXR(A)=0:YR(A)
=-1:GOTO460
330 IFXR(A)=-1THENIFP<>31ANDP<>27ANDP<>2
8THENGOSUB730
340 IFXR(A)=-1THENIFP=27THENXR(A)=0:YR(A)
)=1:GOTO470
350 IFXR(A)=-1THENIFP=28THENXR(A)=0:YR(A)
    
```

# programme

```

880 CHAR1,2,3,"!{SPACE L.GRN}'{SPACE}'{S
PACE2}'{SPACE3}'{SPACE4 BLUE}'{SPACE2 L.
GRN}'{SPACE2}'{SPACE}'{SPACE2 BLUE}!" <145>
890 CHAR1,2,4,"!{SPACE BRN}'{SPACE}'{SPA
CE L.GRN}'{BRN}'{SPACE3}'{SPACE4 BLUE}'{
SPACE2 BRN}'{SPACE2}'{SPACE}'{SPACE L
.GRN}'{SHIFTSPACE}'{SHIFTSPACE}'{SPACE B
LUE}!" <29>
900 CHAR1,2,5,"!{SPACES BRN}'{SPACE2 BLU
E}'{SPACE12 BRN}'{SPACE}'{SPACE}'{SPACE
BLUE}!" <43>
910 CHAR1,2,6,"\-----\-----1
^" <177>
920 CHAR1,2,7,"{L.GRN}'{SPACE2 BLUE}'{L
.GRN}'{SPACE}'{SPACE}'{SPACE2}'{SPACE
BLUE}'{SPACE2 L.GRN}'{SPACE}'{SPACE BLUE
}'{SPACE3 L.GRN}'{SPACE2}'{SPACE2}'" <171>
930 CHAR1,2,8,"{BRN}'{L.GRN}'{SPACE BLUE
E}'{BRN}'{L.GRN}'{BRN}'{SPACE}'{SPACE
L.GRN}'{BRN}'{L.GRN}'{BLUE}'{SPACE2 BRN}'
{SPACE}'{SPACE BLUE}'{SPACE3 BRN}'{SPAC
E2}'{L.GRN}'{SPACE BRN}'{" <207>
940 CHAR1,2,9,"{SPACE L.GRN}'{BRN}'{SPAC
E BLUE}'{SPACE BRN}'{SPACE3 L.GRN}'{SPAC
E2 BRN}'{SPACE}'{BLUE}'{SPACE6}\-----}{
SPACE BRN}'{SPACE3}" <127>
950 CHAR1,2,10,"{SPACE}'{SPACE2 BLUE}'{S
PACES BRN}'{SPACES BLUE}'{SPACE L.GRN}'
{SPACE3}'{SPACE}'{SPACE2 BLUE}'{SPACE2
L.GRN}'{SPACE2}" <39>
960 CHAR1,2,11,"{BLUE}[ ^-----^
SHIFTSPACE BRN}'{L.GRN}'{SPACE2}'{BRN}'{
(L.GRN}'{BRN}'{SPACE L.GRN}'{BLUE}'{SPA
CE L.GRN}'{BRN}'{L.GRN}'{SPACE}" <199>
970 CHAR1,2,12,"{BLUE}'{SPACE15}'{SPACE2
BRN}'{SPACE2}'{SPACE3}'{SPACE2}'{BLUE}'
{SPACE BRN}'{SPACE}'{SPACE}" <69>
980 CHAR1,2,13,"{BLUE}'{SPACE15}\-----
^'{SPACES}" <37>
990 CHAR1,2,14,"!{SPACE20 L.GRN}'{SPACE
2}'{SHIFTSPACE2}'{SPACE BLUE}'{SPACE2 L.
GRN}'{SPACE}" <191>
1000 CHAR1,2,15,"{BLUE}'{SPACE20 BRN}'{
SPACE2}'{L.GRN}'{SPACE BRN}'{SPACE BLUE}'
{SPACE2 BRN}'{L.GRN}'" <97>
1010 CHAR1,2,16,"{BLUE}'{SPACE7 RED}'{SP
ACE17 BRN}'{SPACE3 BLUE}'{SPACE4 BRN}'{" <248>
1020 CHAR1,2,17,"{BLUE}'{SPACE3 L.GRN}'
{SPACE2 RED}'{BLUE}'{SPACE}[
^-----1" <48>
1030 CHAR1,2,18,"!{SPACE L.GRN}'{SPACE B
RN}'{L.GRN}'{SPACE RED}'{L.GRN}'{SPACE
4}'{BLUE}'{SPACE L.GRN}'{SPACE2}'{SPA
CE4}'{SHIFTSPACE2}'{SPACE2 BLUE}!" <161>
1040 CHAR1,2,19,"!{SPACE BRN}'{SPACE2 L.
GRN}'{BRN}'{L.GRN}'{SPACE BRN}'{SHIFTSP
ACE L.GRN}'{SPACE BRN}'{BLUE}'{SPACE BR

```

```

N}'{SPACE2}'{SHIFTSPACE4}'{SPACE2}'{S
PACE2 BLUE}!" <207>
1050 CHAR1,2,20,"!{SPACE4 BRN}'{SPACE}'{
SPACE4}'{SPACE2 BLUE}'{SPACE18}!" <243>
1060 CHAR1,2,21,"\-----
^" <218>
1070 COLOR1,8,4:CHAR1,6,13,"{RVSON SPACE
}RAILWAY{SPACE RVSOFF}" <112>
1080 CHAR1,5,15,"BY{SPACE}C.{SPACE}REINK
E" <161>
1090 RESTORE1110 <123>
1100 FORB=1TO9:READX,Y:CHAR1,X,Y,"{RVSON
}" +STR$(B) + "{RVSOFF}":NEXTB <139>
1110 DATA19,3,7,7,19,7,26,7,7,10,16,10,3
0,12,30,16,19,20 <113>
1120 FORA=1TOLE <32>
1130 X=INT(RND(1)*36)+2:Y=INT(RND(1)*19)
+2 <80>
1140 IFY<>17THEN1160 <207>
1150 IFX>9ANDX<17THEN1130 <175>
1160 P=PEEK(3072+X+40*Y):IFP<270RP>33THE
N1130 <97>
1170 COLOR1,12,6 <163>
1180 IFP<>32THENAL(A)=PEEK(3072+X+40*Y) <194>
1190 UNP=Z:GOTO1210,1220,1230,1240,1230,
1130,1240 <86>
1200 GOTO1130 <238>
1210 XR(A)=0:YR(A)=1:PL$(A)="&":GOTO1250
<77>
1220 XR(A)=1:YR(A)=0:PL$(A)="#":GOTO1250
<4>
1230 XR(A)=-1:YR(A)=0:PL$(A)="*":GOTO125
0 <143>
1240 XR(A)=0:YR(A)=-1:PL$(A)="%":GOTO125
0 <186>
1250 XP(A)=X:YP(A)=Y:CHAR1,X,Y,PL$(A):NE
XTA <27>
1260 RETURN <126>
1270 RESTORE1280:FORB=1TO9:READX,Y:WX(B)
=X:WY(B)=Y:NEXTB:RETURN <151>
1280 DATA18,2,6,6,18,6,25,6,6,11,18,11,3
2,13,32,17,18,21 <254>
1290 SC=SC+2:IFEF=1THENE=0:TI$="000000"
<29>
1300 IFVAL(TI$)>100RGL=1THENSC=SC-1.5
<70>
1310 IFSC>200THEN1340 <182>
1320 CHAR1,20,24,"{YELLOW}":PRINTUSING"#
###";SC; <146>
1330 COLOR1,12,7:GOTO500 <78>
1340 IFGL=0THENSC=0:CHAR1,17,17,"{BLUE}_
_":QS=QS+200:GL=1 <254>
1350 GOTO500 <142>
1360 GL=0:A=LE:LE=LE+1:QS=QS-SC:SC=0:NEX
TA:GOTO120 <228>

```

ENDE DES LISTINGS



# DOMINO

## Stein an Stein

Gespielt wird nach den internationalen Domino-Regeln (Werden im Programm auch noch einmal erklärt.). Während des Spiels braucht man nur die Zahlen ein

zugeben, ohne 'Return' zu drücken. Kann man keinen Stein mehr setzen, so gibt man einfach \*\*\* ein. Könnte man allerdings doch noch setzen, macht sich der

Computer bemerkbar. Auch wenn man falsche Zahlen eingegeben hat fordert der Computer mich erneut auf zu setzen. Vorsicht: Wenn sie nach der Frage

"Noch ein Spiel" ein 'N' eingeben, wird ein Reset erzeugt. Also nach dem Eintippen das Programm erst abspeichern!

\*\*\*

## CHECKSUMMEN (OC V1.0)

```

10 REM ***** <191>
20 REM * * * * * <117>
30 REM * D O M I N O * <84>
40 REM * * * * * <137>
50 REM * BY * <17>
60 REM * * * * * <157>
70 REM * D. KRAUSE * <234>
80 REM * * * * * <177>
90 REM ***** <15>
100 PRINT "CLEAR" <130>
110 COLOR 1:COLOR 4,4,3 <18>
120 CHAR1,19,1,"(BLUE RVSON) [C] (SPACE) 19
86 (SPACE) BY (SPACE) D. KRAUSE (RVSOFF GREEN)
" <126>
130 CHAR1,21,3,"(RVSON WHITE) <205>
140 CHAR1,21,4,"| |(SPACE) | | | | | | | | | | |(SP
ACE) |" <148>
150 CHAR1,21,5,"| |(SPACE) |(SPACE) |(SPACE)
| | |(SPACE3) |(SPACE3) |" <8>
160 CHAR1,21,6,"| |(SPACE) | | | | | | | | | | |(SP
ACE) |" <182>
170 CHAR1,21,7,"| | | | | | | | | | | | | | | | |
" <129>
180 CHAR1,34,8,"| | | | | | | | | | | | | | | | |
" <58>
190 CHAR1,34,9,"| |(SPACE) |" <195>
200 CHAR1,34,10,"| |(SPACE3) |" <209>
210 CHAR1,34,11,"| |(SPACE) |" <221>
220 CHAR1,34,12,"| | | | | | | | | | | | | | | | |
" <236>
230 CHAR1,34,13,"| |(SPACE2) |" <83>
240 CHAR1,34,14,"| |(SPACE3) |" <25>
250 CHAR1,34,15,"| |(SPACE2) |" <217>
260 CHAR1,34,16,"| | | | | | | | | | | | | | | | |
(RVSOFF)" <35>
270 CHAR1,34,18,"(BLUE RVSON) C-164 (RVSOFF
E GREEN)" <135>
280 PRINT "HOME":FORA=1TO18:PRINTTAB(A)
"(RVSON) DO (RVSOFF) MI (RVSON) NO":NEXTA:FOR
A=1TO99:NEXTA <22>
290 PRINT "HOME":FORA=1TO18:PRINTTAB(A)
"DO (RVSON) MI (RVSOFF) NO":NEXTA:FORA=1TO99
:NEXTA <207>
300 PRINT "(RIGHT DORN3 WHITE) KENNEN (SPAC
E) SIE (SPACE) DIE (SPACE) SPIELREGELN? (SPACE
6) <(RVSON FLASHON) J/N (FLASHOFF RVSOFF) > (
GREEN)" <42>
310 GETA$: IFA$="" THEN 280 <250>
320 IFA$="N" THEN 1530 <98>
330 PRINT "UP": <138>
340 PRINT "(WHITE SPACE2) BITTE (SPACE RV
S ON FLASHON) WARTEN (FLASHOFF RVSOFF SPACE2
)" <11>
350 DIMA$(28),B$(28),C$(28) <135>
360 C=0:FORA=0TO6:FORB=0TOC:D=D+1:A$(D)=
CHR$(48+A)+CHR$(48+B):NEXTB:C=C+1:NEXTA <248>
370 FORA=1TOB <38>
380 C=0:D=0:FORA=1TO28:B=INT(2*RND(1)+1)
:IFB=1THENC=C+1:B$(C)=A$(A):GOTO400 <27>
390 D=D+1:C$(D)=A$(A) <122>
400 A$(A)="" :NEXTA:A=0:FORB=1TOC <28>
410 A=A+1:A$(A)=B$(B):NEXTB:FORB=1TOD:A=
A+1:A$(A)=C$(B):NEXTB:NEXTA1 <132>
420 B=0:FORA=1TO28:B$(A)="" :C$(A)="" <140>
430 IFA/2=INT(A/2)THENC$(A/2)=A$(A):GOTO
450 <132>
440 B=B+1:B$(B)=A$(A) <137>
450 A$(A)="" :NEXTA <168>
460 FORA=1TO14:C1$=MID$(C$(A),1,1):C2$=M
ID$(C$(A),2,1) <213>
470 C$(A)="" |(DOWN LEFTS) | "+C1$+" | "+C
2$+" |(DOWN LEFTS) | |(UP2)" <76>
480 NEXTA <229>
490 C$="(SPACES DOWN LEFTS SPACES DOWN L
EFTS SPACES UP2)" <173>
500 SD=14:BD=14 <96>
510 PRINT "CLEAR":;COLOR 4,3,3 <99>
520 FORA=1TO40:PRINT "(GREEN) (WHITE)":;N

```

```

EXTA:PRINT <250>
530 A=29:PRINTTAB(A) " _____ ":PRINTTA
B(A) "(RVSON SPACE) ICH (SPACE) HABE: " <247>
540 PRINTTAB(A) "(RVSON SPACE) 14 (SPACE) ST
EINE":PRINTTAB(A) " _____ " <138>
550 FORA=1TO40:PRINT "(GREEN) (WHITE)":;N
EXTA:PRINT <24>
560 PRINT " _____ " <213>
570 FORA=1TO19:PRINT "| (SPACE)":;NEXTA:PR
INT"|" <117>
580 PRINT " _____ " <135>
590 PRINTTAB(36) "| (SHIFTSPACE) |" <93>
600 PRINT "(SHIFTSPACE2) _____ " <79>
610 PRINT "(SPACE2)":;FORA=1TO18:PRINT"|(
SPACE)":;NEXTA:PRINT"|" <147>
620 PRINT "(SPACE2) _____ " <196>
630 PRINT "(SPACE2) |(SPACE) |" <97>
640 PRINT "(SPACE2) _____ " <119>
650 PRINT "(SPACE2) |":;FORA=1TO17:PRINT"|(
SPACE) |":;NEXTA:PRINT <134>
660 PRINT "(SPACE2) _____ " <95>
670 FORA=1TO40:PRINT "(GREEN) (WHITE)":;N
EXTA:PRINT <144>
680 PRINT "(SPACE2)":;FORA=1TO7:PRINTC$(A
):;NEXTA:PRINT "(DOWN <158>
690 PRINT "(SPACE2)":;FORA=1TO7:PRINTC$(A
+7):;NEXTA:PRINT "(DOWN <99>
700 DATA 9,51,53,55,57,59,61,63,65,67,69
,71,73,75,77,79,81,83,85,165,245,243 <13>
710 DATA 241,239,237,235,233,231,229,227,
225,223,221,219,217,215,213,211,291 <92>
720 DATA 371,373,375,377,379,381,383,385,
387,389,391,393,395,397,399,401,403 <55>
730 C=14:B=14 <206>
740 B$="6":FORA=1TO14:C1$=MID$(C$(A),13,
1):C2$=MID$(C$(A),15,1) <134>
750 IFC1$="6"ANDC2$="6"THEN 780 <156>
760 IFB$(A)="66"THEN 1110 <148>
770 NEXTA <9>
780 IFAA=28ORSO=0THENS0=0:BD=5:GOTO1380 <91>
790 GOSUB 930:PRINT "(HOME DOWN2) WELCHER (S
PACE) STEIN? (SPACE4)":;VOLB:SOUND 2,800,5 <163>
800 FORA=1TO2 <96>
810 GETA$: IFA$="" THEN 810 <221>
820 A$(A)=A$:PRINTA$:"(SPACE)":;NEXTA:PR
INT <23>
830 IFA$(1)=""AND A$(1)="" THEN 1460 <86>
840 IFASC(A$(1))<48ORASC(A$(1))>55THEN GO
SUB 930:GOTO 780 <116>
850 IFASC(A$(2))<48ORASC(A$(2))>55THEN GO
SUB 930:GOTO 780 <152>
860 FORA=1TO14:C1$=MID$(C$(A),13,1):C2$=
MID$(C$(A),15,1) <74>
870 IFAA>0THEN 890 <215>
880 IFA$(1)<"6"OR A$(2)<"6" THEN 920 <52>
890 IFA$(1)=C1$AND A$(2)=C2$AND C1$=B$ THEN
960 <92>
900 IFA$(1)=C2$AND A$(2)=C1$AND C2$=B$ THEN
1010 <246>
910 NEXTA <149>
920 GOSUB 930:GOTO 780 <60>
930 PRINT "(HOME DOWN2 SPACE2)" <122>
940 FORA=1TO999:NEXTA <86>
950 RETURN <71>
960 REM <82>
970 AA=AA+1:READ T1,T2 <68>
980 POKE 3304 +T1,ASC(C1$):POKE 3304 +T2,A
SC(C2$) <176>
990 C$(A)=C$:B$=C2$ <151>
1000 GOTO 1060 <46>
1010 REM <132>

```



# tips & tricks

```

1020 AA=AA+1:READT1,T2
1030 POKE3304 +T1,ASC(C2$):POKE3304 +T2,
ASC(C1$)
1040 C$(A)=C$:B$=C1$
1050 GOTO1060
1060 BD=SD-1:UU=0
1070 PRINT"(HOME DOWN17)"
1080 PRINT"(SPACE2)":FORA=1TO7:PRINTC$(
A):NEXTA:PRINT"(DOWN2)"
1090 PRINT"(SPACE2)":FORA=1TO7:PRINTC$(
A+7):NEXTA:PRINT"(DOWN2)"
1100 GOTO1150
1110 AA=AA+1:READT1,T2
1120 POKE3304 +T1,54+128:POKE3304 +T2,54
+128
1130 B$(A)="**"
1140 B$="6":BD=BD-1:GOTO780
1150 IFAA=280RBD=0THENSDB=5:BD=0:GOTO1380
1160 GOSUB930:PRINT"(HOME DOWN2)ICH(SPAC
E)SETZE:(SPACE3)":SOUND1,500,5
1170 FORA=1TO14:B1$=MID$(B$(A),1,1):B2$=
MID$(B$(A),2,1)
1180 IFB$=B1$THEN1250
1190 IFB$=B2$THEN1290
1200 NEXT:GOSUB930
1210 PRINT"(HOME DOWN2)ICH(SPAC)KANN(SP
ACE)NICHT(SPAC)SETZEN!":FORT=800TO500ST
EP-50:SOUND2,T,5:NEXTT
1220 FORZ=1TO999:NEXT
1230 IFUU=1THEN1380
1240 UU=1:GOTO780
1250 AA=AA+1:READT1,T2
1260 POKE3304 +T1,ASC(B1$)+128 :POKE3304
+T2,ASC(B2$)+128
1270 B$(A)="**":B1$=B1$+"(SPACE2)" +B2$
1280 B$=B2$:GOTO1330
1290 AA=AA+1:READT1,T2
1300 POKE3304 +T1,ASC(B2$)+128:POKE3304
+T2,ASC(B1$)+128
1310 B$(A)="**":B1$=B2$+"(SPACE2)" +B1$
1320 B$=B1$
1330 PRINTBB$:BD=BD-1
1340 BD$=STR$(BD)
1350 IFLEN(BD$)<3THENBD$="(SPACE)" +BD$:G
OTO1350
1360 PRINT"(HOME DOWN3 RVSON)"SPC(29)BD$
1370 FORZ=1TO999:NEXT:GOTO780
1380 IFSD>BDTHENPRINT"(HOME DOWN2)ICH(SP
ACE)HABE(SPAC)GEWONNEN!(SPACE4)"
1390 IFSD<BDTHENPRINT"(HOME DOWN2)SIE(SP
ACE)HABEN(SPAC)GEWONNEN!(SPACE3)"
1400 IFSD=BDTHENPRINT"(HOME DOWN2)UNENTS
CHIEDEN!(SPACE8)"
1410 PRINT"NOCH(SPAC)EIN(SPAC)SPIEL?(S
PACE RVSON FLASHON)J/N(FLASHOFF RVSOFF)"
1420 BETA$: IFA$="" THEN1420
<118>
<211>
<188>
<97>
<23>
<158>
<44>
<53>
<146>
<209>
<182>
<121>
<248>
<45>
<249>
<76>
<187>
<247>
<84>
<153>
<214>
<116>
<254>
<93>
<204>
<90>
<162>
<99>
<98>
<84>
<79>
<2>
<47>
<18>
<20>
<108>
<0>
1430 IFA$="J"THENGOSUB1690:RUN350
1440 IFA$="N"THENSYS65529
1450 CLR
1460 FORA=1TO14:C1$=MID$(C$(A),13,1):C2$
=MID$(C$(A),15,1)
1470 IFC1$=B$ORC2$=B$THEN1490
1480 NEXT:GOTO1510
1490 GOSUB930:PRINT"(HOME DOWN2)SIE(SPAC
E)KOENNEN(SPAC)SETZEN!(SPACE)":FORT=500
TO800STEP50:SOUND1,T,5:NEXTT
1500 FORA=1TO999:NEXT:GOTO780
1510 IFUU=1THEN1380
1520 GOTO1150
1530 PRINT"(CLEAR BLUE)":COLOR4,3,3
1540 PRINTTAB(13)"SPIELREGELN"
1550 PRINTTAB(13)"(WHITE) (BLU
E)"
1560 PRINTSPC(13)"(WHITE DOWN) (BLU
E)"
1570 PRINT"(RIGHT)WIR(SPAC)SPIELEN(SPAC
E)WHITE RVSON)DOMINO(RVSOFF BLUE)!"
1580 PRINT"(RIGHT DOWN2)LEGEN(SPAC)SIE(
SPACE)IHRE(SPAC)STEINE(SPAC)SOLANGE(SP
ACE)AB,(SPACE)BIS
1590 PRINT"(RIGHT DOWN)SIE(SPAC)KEINEN(
SPACE)STEIN(SPAC)MEHR(SPAC)HABEN(SPAC
)ODER(SPAC)KEI-
1600 PRINT"(RIGHT DOWN)NEN(SPAC)STEIN(S
PACE)MEHR(SPAC)LEGEN(SPAC)KOENNEN.(SPA
CE)SIEGER
1610 PRINT"(RIGHT DOWN)IST,(SPACE)WER(SP
ACE)AM(SPAC)ENDE(SPAC)DIE(SPAC)WENIGS
TEN(SPAC)STEINE
1620 PRINT"(RIGHT DOWN)ODER(SPAC)KEINEN
(SPAC)MEHR(SPAC)HAT!
1630 PRINT"(RIGHT DOWN2)KOENNEN(SPAC)SI
E(SPAC)NICHT(SPAC)ABLEGEN,(SPACE)GEBEN
(SPAC)SIE
1640 PRINT"(WHITE RIGHT DOWN) ** (BLUE S
PACE)EIN.(SPACE3 WHITE RVSON)616(RVSOFF
BLUE SPACE)BEGINNT!
1650 PRINTTAB(26)"(WHITE DOWN) (BLU
E)"
1660 PRINT"(RIGHT)KOENNEN(SPAC)WIR(SPAC
E)BEGINNEN?(SPACE3)<(WHITE RVSON FLASHON
)RETURN(FLASHOFF RVSOFF BLUE)>"
1670 BETA$: IFA$="" THEN1670
1680 GOTO330
1690 SCNCLR:PRINT"(DOWN12 RIGHT13)BITTE(
SPACE RVSON FLASHON)WARTEN(FLASHOFF RVSO
FF)":RETURN
ENDE DES LISTINGS

```



## REFUSAL-BASICLADER

Wird dieser kleine Basiclader in ein eigenes Programm mit eingebunden, so wird ein radikaler List- und Saveschutz erzeugt, den man mit einer Autostart-Routine noch verbessern kann.

Deshalb sollte sich der Basiclader gleich am Anfang eines Programms befinden, wodurch nun gleich durch das Starten des Programms eine Maschinensprache

Routine im Systemspeicher abgelegt und aktiviert wird. Versucht man nun das Programm zu listen oder abzuspeichern wird auf den ersten Anschein nur ein harmloser RESET durchgeführt, von dem sich eingefleischte Profis nicht abschrecken lassen. Diese versuchen nun ihre altbewährten OLD-Routinen anzuwenden, die aber sprichwörtlich nur ein Schlag ins Leere erzeugen. Der ge

samte Basic-Speicher wurde nämlich mit Nullen aufgefüllt und das vorhandene Programm ist bis auf die Maschinen-Routine des List- und Saveschutzes ständig zerstört.

Es hat auch keinen Sinn vor einem List- und Saveversuch die List- und Savevektoren zu verändern, da diese ständig, durch eine eingebaute Interrupt-Routine, umgestellt werden.

Achtung: Die Daten richtig abtippen, sonst wird beim Erzeugen der Maschinenroutine ein normaler RESET verursacht. Dieser ist jedoch durch POKE4098, 1:SYS34892 leicht wieder aufzuheben. Ist die Maschinenroutine einmal erzeugt worden, kann sie immer wieder durch SYS1630 aktiviert werden. Beim Abtippen können selbstverständlich die REMARKS weggelassen werden.

```

10 REM "*****"
12 REM " "
14 REM " LIST & SAVE "
16 REM " "
18 REM " REFUSAL "
20 REM " "
22 REM " WRITTEN BY "
24 REM " "
26 REM " GUNTER MASSAR "
28 REM " "
30 REM "*****"
32 REM
34 FOR K=1630 TO 1684:READ A:S=S+A
35 POKE K,A:NEXT K
36 IF S<>5455 THEN SYS 65526:ELSE SYS 1630
37 REM
38 REM " DATAS UM NEUEN IRQ-VEKTOR "
40 REM "EINZUSTELLEN (HEX 065E-066A) "
42 REM

```

```

44 DATA 120,162,107,160,6,142,20,3,140
45 DATA 21,3,88,96
46 REM
48 REM " NEUE IRQ-ROUTINE: "
50 REM "EINSTELLEN VOM NEUEN LIST UND"
52 REM " SAVE-VEKTOR (HEX 066B-067D) "
54 REM
56 DATA 162,126,160,6,142,6,3,142,48,3
57 DATA 140,7,3,140,49,3,76,14,206
58 REM
60 REM " NEUE LIST- UND SAVE-ROUTINE "
62 REM " (HEX 067E-0694) "
64 REM
66 DATA 169,0,160,16,170,129,55,198,55
67 DATA 208,250,196,56,240,5,198,56
68 DATA 76,131,6,76,246,255

```

READY.

## HARDCOPY-BLITZ

Das vorliegende Maschinenprogramm druckt eine Hardcopy des C-16 Bildschirmes innerhalb von 20 Sekunden!

Das Programm wird mit dem entsprechenden Basicprogramm geladen und mit SYS 818 gestartet. Zu beachten ist, daß die Maschinenroutine im Kassettenpuffer abgelegt wurde, um den im Graficmodus ohnehin nicht sehr großen Speicher des C-16 zu schonen. Mit Hilfe des Assemblerli-

stings sollte ein Verschieben in einen anderen Bereich ohne weiteres möglich sein. Des weiteren muß darauf geachtet werden, daß bei Aufruf der Routine, der Kanal 1 geöffnet ist, da die Datenübertragung zum Drucker darüber stattfindet.

Das Programm ändert die Seitenlänge des Papiers und führt nach jeder Hardcopy ein Formfeed aus, um zu vermeiden, daß der Ausdruck durch ein irrtümliches

internes Formfeed des Druckers in zwei Teilen erscheint.

Die eigentliche Druckroutine ist als reines Maschinenprogramm geschrieben, um sie möglichst schnell zu machen; es kann mit dem Monitor des C-16 bequem eingegeben werden. Das Ladeprogramm in Basic dient lediglich dem Aufrufen der Grafic, es löscht sich nach Beendigung selbst und sollte daher direkt nach dem Eingeben abgespeichert wer-

den. Das gilt im Prinzip auch für das Maschinenprogramm, das sich bei Abstürzen durch falsche Eingabe unter Umständen selbst zerstört.

Das Programm wurde für den Citizen 120 D entwickelt. Es läuft aber auf allen Epon-kompatiblen Druckern. Da die Drucker-Initialisierung im ersten Teil des Programms stattfindet, ist eine Anpassung an andere Druckertypen durchaus möglich.

### C-16 LISTING + CHECKSUMMEN (DC V1.0)

```

10 REM***** <203>
20 REM BILDSCHIRM HARDCOPY <112>
30 REM COPYRIGHT ANDREAS GLOECKNER <104>
40 REM AM SONNENSTEIN 46 <78>
50 REM 5804 HERDECKE <182>
60 REM***** <253>
70 GRAPHIC2,1 <208>

```

```

80 GRAPHICO <142>
90 PRINT" {HOME}LOAD"+CHR$(34)+"!HARDCOPY <55>
{SPACE}B18"+CHR$(34)+" ,B,1"
100 PRINT" {DOWN}NEW" <32>
110 POKE1319,19:POKE1320,13:POKE1321,13:
POKE1322,147:POKE239,4 <178>
120 END <248>
ENDE DES LISTINGS

```

READY.

MONITOR

PC SR AC XR YR SP  
; FF00 00 00 FF 00 FB

```

>0332 A2 01 20 C9 FF A0 00 A9
>033A 1B 20 D2 FF A9 43 20 D2
>0342 FF A9 2A 20 D2 FF A9 1B
>034A 20 D2 FF A9 41 20 D2 FF

```

```

>0352 A9 08 20 D2 FF A9 07 B5
>035A DB A9 3E B5 D9 A9 10 B5
>0362 DA A9 1E B5 DB A9 38 B5
>036A DF A9 01 B5 E0 B5 E1 A9
>0372 00 B5 E2 A9 28 B5 DC A9
>037A 1B 20 D2 FF A9 2A 20 D2
>0382 FF A9 05 20 D2 FF A9 C8
>038A 20 D2 FF A9 00 20 D2 FF
>0392 A9 19 B5 DD A9 09 B5 DE
>039A C6 DE D0 0C C6 DD D0 1D

```

```

>03A2 C6 DC D0 29 20 E4 03 60
>03AA B1 DB 20 D2 FF 38 A5 DB
>03B2 E5 E1 B5 DB A5 D9 E5 E2
>03BA B5 D9 4C 9A 03 38 A5 DB
>03C2 E5 DF B5 DB A5 D9 E5 E0
>03CA B5 D9 4C 96 03 18 A5 DB
>03D2 65 DA B5 DB A5 D9 65 DB
>03DA B5 D9 A9 0D 20 D2 FF 4C
>03E2 79 03 A9 0D 20 D2 FF A9
>03EA 0C 20 D2 FF 20 CC FF 60

```

# AUTO- START

Mit dem vorliegenden Programm werden Ihre Programme nach dem Laden von Kassette automatisch gestartet!

# tips & tricks / checksummer

Laden-Starten-dann wird nach dem Filenamem des Programms gefragt, welches automatisch geladen und gestartet werden soll. Nach der Eingabe des Filenamens (max. 15 Zeichen), wird dieser nach \$025E (Bereich für Filenamem) geschrieben und die SAVE-ROUTINE mit SYS1630 ausgeführt. In dieser wird der Basic-Warmstart-Vektor (Originaladres-

se \$8712) auf \$028A gesetzt (Dies ist die Anfangsadresse der zweiten Assembler-Routine). Danach wird der Bereich von \$028A bis \$0304 auf Kasette gespeichert. (Siehe Assembler-Listing) Jetzt wird der Originalwert (\$8712) in den Basic-Warmstart-Vektor geschrieben. Der Computer ist somit wieder im Direktmodus. Ein Neustart des Programms

bewirkt, daß weitere Autostart-Vorspanne auf Kasette gespeichert werden können. Nach dem Autostart-Vorspann wird dann das eigentliche Basic-Programm abgespeichert.

Beim Laden des Programms startet automatisch (durch Änderung des Basic-Warmstart-Vektors) die Assembler-Routine (\$028A) und das nachfolgende Programm wird

geladen. Durch die Ausgabe von 'RUN' und 'RETURN' startet das Programm. (Siehe Assembler-Listing)

Um reine Assembler-Programme damit starten zu können, kann der Bereich ab \$02B9 entfallen. Stattdessen setzt man dort ein 'JMP Startadresse des Assembler-Programms' ein.

\*\*\*

## C-16 LISTING + CHECKSUMMEN (OC V1.0)

```

1 SCNCLR:COLOR4,1:COLOR0,1:COLOR1,2,6 <194>
2 DATA7,BA,8D,02,03,A9,02,8D <178>
3 DATA03,03,A9,01,A2,01,A0,01 <180>
4 DATA20,BA,FF,A9,00,A2,5E,A0 <117>
5 DATA02,20,8D,FF,A9,8A,85,8B <62>
6 DATAA7,02,85,D9,A9,8B,A2,04 <98>
7 DATAA0,03,20,8B,FF,A9,12,8D <197>
8 DATA02,03,A9,87,8D,03,03,20 <11>
9 DATA12,87,60,EA,EA,EA,EA,EA <173>
10 FORT=1630 TO 1694 <117>
11 READA#:A=DEC(A#):POKET,A:NEXT <225>
12 CLR:RESTORE13 <131>
13 DATAA9,00,8D,15,FF,8D,19,FF <245>
14 DATAA9,90,20,D2,FF,A9,93,20 <5>
15 DATAD2,FF,A9,00,A2,01,A0,01 <185>
16 DATA20,BA,FF,A9,00,20,8D,FF <208>
17 DATAA9,00,20,D5,FF,A9,12,8D <180>
18 DATA02,03,A9,87,8D,03,03,A9 <70>
19 DATA13,8D,27,05,A9,0D,8D,2B <127>

```

```

20 DATA05,A9,02,85,EF,A9,12,8D <132>
21 DATA00,0C,A9,15,8D,01,0C,A9 <106>
22 DATA0E,8D,02,0C,AD,35,03,85 <40>
23 DATA2D,AD,36,03,85,2E,4C,12 <58>
24 DATA87,00,8D,00,AD,35,03,18 <193>
25 FORT=650 TO 739 <146>
26 READA#:A=DEC(A#):POKET,A:NEXT <240>
27 PRINT"⟨CLEAR DOWN⟩":INPUT"FILENAME"; <1>
S# <82>
28 S=LEN(S#) <82>
29 IFS>15THEN27:ELSEPOKE1650,S <10>
30 FORI=1TOS:POKE605+I,ASC(MID$(S#,I,1)) <135>
31 NEXT:SYS1630 <67>
32 REM ***** <143>
33 REM * AUTOSTART C-116/C-16/PLUS4 * <136>
34 REM * COPYRIGHT BY D. SCHEEL 1986 * <185>
35 REM * JEGLICHE AENDERUNG DES COPY- * <115>
36 REM * RIGHTS IST VERBOTEN ! * <148>
37 REM ***** <148>
ENDE DES LISTINGS

```

## Checksummer VC 1.0 für VC-20

Ebenso wie der C-64-Checksummer besteht auch das VC-20-Programm aus einer kleinen Maschinenroutine. Aus Speicherplatzgründen konnte jedoch dieses Programm nicht »unter« dem ROM abgelegt werden, sondern

hier mußte der bekannte Kassettenpuffer erhalten. Dies hat den Vorteil, daß kein Basic-Speicher verloren geht. Der Nachteil besteht darin, daß nach der Aktivierung des Checksummers keine Kassettenoperationen durchführbar sind, da diese den Kassettenpuffer benötigen. Um die Kassettenoperationen wieder zu ermöglichen, muß der Checksummer durch den Befehl »SYS 58459«,

sowie durch das Betätigen der Tasten STOP/RESTORE, abgeschaltet werden. Aktiviert wird der Checksummer mit SYS 828.

Die Funktion und Arbeitsweise stimmt mit dem C-64-Checksummer überein und kann dort nachgelesen werden.

```

1 REM ***** <176>
2 REM CHECKSUMMER VC 1.0 <126>
3 REM VC 20 VERSION <8>
4 REM COPYRIGHT FRANK BRALL <126>
5 REM T R O N I C - V E R L A G <228>
6 REM ***** <181>
7 REM <150>
10 PRINT"⟨CLEAR DOWN SPACE⟩CHECKSUMMER⟨SPACE⟩VC⟨SPACE⟩1.0⟨SPACE⟩" <216>

```

```

20 PRINT"⟨DOWN⟩COPYRIGHT⟨SPACE⟩FRANK⟨SPACE⟩BRALL" <221>
30 PRINT"⟨DOWN⟩-----⟨SPACE⟩03.01.86⟨SPACE⟩CE)-----" <152>
35 PRINT"⟨DOWN SPACE4⟩TRONIC-VERLAG" <123>
60 DA=0:FORI=828 TO 1020 <158>
61 READ DA:SU=SU+DA <163>
62 POKE I,DA <64>
63 NEXT I <83>
65 IF SU<>22919 THEN PRINT"⟨DOWN⟩FEHLER⟨SPACE⟩IN⟨SPACE⟩DATA-ZEILEN⟨SPACE⟩!":END <14>
70 PRINT"⟨DOWN⟩" <222>
71 PRINT"AKTIVIEREN⟨SPACE⟩=SYS⟨SPACE⟩828" <9>
72 PRINT"ABSCHALTEN⟨SPACE⟩=SYS⟨SPACE⟩58459" <74>
73 PRINT"VOR⟨SPACE⟩BETRIEB⟨SPACE⟩DER⟨SPACE⟩DATA-" <40>
74 PRINT"SETTE⟨SPACE⟩DEN⟨SPACE⟩CHECKSUMMER" <38>
75 PRINT"ABSCHALTEN⟨SPACE⟩UND⟨SPACE⟩DIE" <80>
76 PRINT"TASTEN⟨SPACE⟩STOP/RESTORE" <198>
77 PRINT"BETAETIGEN⟨SPACE⟩!" <19>
80 POKE B16,116:POKE B18,116:POKE B17,19 <151>
6:POKE B19,196:REM SAVE/LOAD AUS

```

```

90 SYS 828:PRINT"⟨DOWN⟩CHECK⟨SPACE⟩1.0⟨SPACE⟩IST⟨SPACE⟩AKTIV⟨SPACE⟩!" <99>
100 DATA 169,71,141,2,3,169,3,141,3,3,96 <85>
,32,105,3,134,122,132
101 DATA 123,32,115,0,170,240,243,162,25 <100>
5,134,58,144,8,134,255,32,121
102 DATA 197,76,225,199,162,1,134,255,76 <161>
,156,196,166,255,224,1,240,3
103 DATA 76,96,197,169,0,141,248,3,141,2 <211>
49,3,141,250,3,160,2,24
104 DATA 177,95,200,113,95,141,250,3,160 <143>
,3,200,177,95,240,44,201,34
105 DATA 208,10,173,248,3,73,1,141,248,3 <94>
,177,95,174,248,3,208,4
106 DATA 201,32,240,228,238,249,3,174,24 <31>
9,3,24,177,95,109,250,3,141
107 DATA 250,3,202,208,244,76,138,3,56,3 <203>
2,240,255,142,246,3,140,247
108 DATA 3,162,0,160,0,24,32,240,255,169 <232>
,91,32,9,225,169,0,174
109 DATA 250,3,32,205,221,169,93,32,9,22 <21>
5,32,63,203,32,63,203,172
110 DATA 247,3,174,246,3,24,32,240,255,1 <121>
62,0,134,255,76,111,3,0
111 DATA 0,0,0,0,32,0 <118>

```

Der VC-20-Checksummer kommt frei Haus unter folgenden Bestellnummern: VV10K (Kassette) oder VV10D (Diskette)

# Steuercodes und Checksummer

**Neu: Sonderzeichen werden durch Klartext ersetzt!**

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

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

### Tastenbezeichnungen für Steuercodes: C-16/116 u. Plus 4 (TRONIC-NORM)

SPACE	LEERZEICHEN	LEERTASTE (GROESSTE TASTE)	ORNG	ORANGE	Commodore-Taste + 1
F1	FUNKTIONSTASTE	FUNKTIONSTASTE F1	BRN	BRAUN	Commodore-Taste + 2
F3	FUNKTIONSTASTE	FUNKTIONSTASTE F3	YL.GRN	GELB-GRÜN	Commodore-Taste - 3
F5	FUNKTIONSTASTE	FUNKTIONSTASTE F5	DINK	ROSA	Commodore-Taste + 4
F7	FUNKTIONSTASTE	FUNKTIONSTASTE F7	BL.GRN	Blau-Grün	Commodore-Taste + 5
F2	FUNKTIONSTASTE	FUNKTIONSTASTE F2	L.BLU	Hellblau	Commodore-Taste + 6
F4	FUNKTIONSTASTE	FUNKTIONSTASTE F4	D.BLU	Dunkelblau	Commodore-Taste + 7
F6	FUNKTIONSTASTE	FUNKTIONSTASTE F6	L.GRN	Hellgrün	Commodore-Taste + 8
F8	FUNKTIONSTASTE	FUNKTIONSTASTE F8			
BLACK	SCHWARZ	CONTROL-TASTE & 1	DOWN	Cursor unten	Taste neben INST DEL
WHITE	WEISS	CONTROL-TASTE & 2	UP	Cursor hoch	3. Taste neben DOWN von rechts oben
RED	ROT	CONTROL-TASTE & 3	RIGHT	Cursor rechts	4. Taste von rechts oben
CYAN	TUERKIS	CONTROL-TASTE & 4	LEFT	Cursor links	5. Taste von rechts oben
PURPLE	PURPUR	CONTROL-TASTE & 5	CLR	Schirm löschen	Taste über Return zusammen mit Shift
GREEN	GRUEN	CONTROL-TASTE & 6	HOME	Cursor in Ecke	Taste über Return
BLUE	BLAU	CONTROL-TASTE & 7	DEL	Delete	Taste ganz rechts oben
YELLOW	GELB	CONTROL-TASTE & 8	Flash on	Blinken ein	Controll-Taste + ,
RVSON	INVERSE EIN	CONTROL-TASTE & 9	Flash off	Blinken aus	Controll-Taste + .
RVSOFF	INVERSE AUS	CONTROL-TASTE & 0			
CTRL	CONTROL	CONTROL-TASTE ZUSAMMEN			

MIT NACHFOLGENDEN ZEICHEN.  
Z.B. CTRLA ENTSPRICHT CTRL & A)

### Tastenbezeichnungen für Steuercodes

### VC-20 und Commodore 64

(TRONIC-NORM)

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

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



## ***SUPER! – Checksummer für CPC 464***

CHECKSUMMER 1.0 CPC, ist für den CPC 464/664 entwickelt worden.

Wie funktioniert unser Checksummer?

Tippen Sie zunächst das Programm in Ihren Computer ein. Bei der Eingabe ist zu beachten, daß auch alle Zwischenräume (Leerzeichen) mit dem Listing übereinstimmen. Die REM-Zeilen müssen ebenfalls mitübernommen werden.

### **Sichern Sie das Programm auf KASSETTE oder Diskette bevor Sie es starten!**

Nachdem Sie es gespeichert haben, aktivieren Sie das Programm mit „RUN 65200“. Es erscheint die Meldung: CHECKSUMMER AKTIV FUNKTIONSTASTE 0. Dies bedeutet, wenn Sie die Taste 0 des Zehnerblocks betätigen, wird der Checksummer aufgerufen. Speichern Sie das Programm jetzt mit dem Befehl SAVE "CHECKV1", A auf Kasette

oder Diskette ab. Nur so können Sie es später mit Merge an Ihre Programme anhängen.

### **Wie benutzt man den Checksummer?**

Wenn Sie ein Programm mit CHECK überprüfen wollen, laden Sie zunächst das zu prüfende Programm (wenn Sie's nicht ohnehin schon im Speicher haben), dann geben Sie den Befehl MERGE "CHECKV1" um das Prüfprogramm anzuhängen. Wichtig ist, daß das zu prüfende Programm keine Zeilennummer besitzt, die größer als 65199 ist. Jetzt aktivieren Sie den Checksummer mit RUN 65200. Nun kann jederzeit die Prüfung mit der Funktionstaste 0 aufgerufen werden. Vergleichen Sie jeweils die im Heft abgedruckten Prüfsummen mit denen, die Ihr Programm ausgibt. Falls Sie einmal einen Fehler feststellen, müssen Sie nicht das Programm neu laden, sondern können ihn normal berichtigen

und dann wieder mit der Funktionstaste 0 das Check-Programm starten. Möchten Sie den Checksummer löschen, geben Sie einfach den Befehl DELETE 65200-ein.

### **Anmerkung zu Checksummer 1.0 CPC:**

Groß-Kleinschreibung beachten! Leerzeichen innerhalb einer Zeile müssen dem ORIGINAL-Listing entsprechen! Leerzeichen am Ende einer Zeile werden ignoriert! Umgewandelte TOOKEN werden automatisch in den Ursprungswert zurückgesetzt! REM-Zeilen müssen eingegeben werden!

*Unser Spezialist Frank Brall arbeitet zur Zeit an einem Checksummer für plus 4. Also plus 4-Fans achtet auf die nächste Compute mit-Ausgabe!*

**ACHTUNG: Einige Listings enthalten das Zeichen "–", dieses Zeichen entspricht dem Pfeil nach links (Taste ganz links oben). Das Zeichen "∧" steht für den Pfeil nach oben. Das Zeichen "∖" steht für das (engl.) Pfund-Symbol. (Taste neben Restore).**

```
65200 *** Tronic CPC Checksummer ***
65210 *** Autor Frank Brall 1985 ***
65220          VERSION 1.0
65230 KEY 128,"65530 REM TRONIC"+CHR$(13)
)+"run 65250"+CHR$(13)
65240 PRINT:PRINT"CHECKSUMMER AKTIV FUNKTIONSTASTE 0":PRINT:END
65250 CLS:PRINT"Tronic-Verlag Checksummer Version 1.0"
65260 PRINT:PRINT:PRINT:PRINT"<B>ILDSCHIRM <D>RUCKER"
65270 E$=INKEY$:E$=LOWER$(E$):IF E$="" THEN 65270
65280 IF E$="b" THEN a=0:GOTO 65310
65290 IF E$="d" THEN a=8:GOTO 65310
65300 GOTO 65200
65310 ADRESS=368:GCHECK=0:CLS:PRINT#A,"CHECK V1: PRUEFSUMMEN: ":PRINT#A:PRINT#A,"ZEILENR.: SUMMEN"
65320 LAENGE=(PEEK(ADRESS+1)*256)+PEEK(ADRESS)
65330 ZNUM=(PEEK(ADRESS+3)*256)+PEEK(ADRESS+2)
65340 IF ZNUM=65200 THEN IF F=0 THEN 65480 ELSE PRINT #A, USING "#####";ZNUM1;:PRINT#A,"-";:PRINT#A,": ";:PRINT #A, USING "#####";ZCHECK:F=0:GOTO 65480
65350 IF ZNUM=65500 THEN 65510
65360 ZCHECK=0:FOR I=0 TO LAENGE-6
```

```
65370 WERT=PEEK(I+ADRESS+4)
65380 IF WERT=32 THEN 65410
65390 ZCHECK=ZCHECK+WERT+(I*WERT)
65400 IF ZCHECK>65535 THEN ZCHECK=ZCHECK-65536:GOTO 65400
65410 NEXT I
65420 IF F=0 THEN ZCHECK1=ZCHECK:ZNUM1=ZNUM:ADRESS=ADRESS+LAENGE:F=1:GOTO 65320
65430 F=0:ZNUM2=ZNUM
65440 PRINT #A,USING "#####";ZNUM1;:PRINT#A,"-";:PRINT #A,USING "#####";ZNUM2;:PRINT#A,": ";:PRINT #A,USING "#####";ZCHECK+ZCHECK1
65450 GCHECK=GCHECK+ZCHECK+ZCHECK1
65460 IF GCHECK>65535 THEN GCHECK=GCHECK-65536:GOTO 65460
65470 ADRESS=ADRESS+LAENGE:GOTO 65320
65480 PRINT#A, "-----"
--"
65490 PRINT #a,"GESAMTCHECKSUMME: "GCHECK
65500 GCHECK=0:A=0:PRINT#A:PRINT#A,"CHECKSUMMER SELBSTTEST !":GOTO 65360
65510 IF GCHECK=62078 THEN PRINT#A,"--ENDE--":END
65520 PRINT "FEHLER IM CHECKSUMMER !":SO UND 1,100,500
65530 REM TRONIC
```

„Fix und fertig“ ist der Checksummer 1.0 CPC natürlich auch erhältlich: Einfach anfordern unter: Bestellnummer: SRV 10K (Kassette) . . . . 10 DM SRV 10D (Diskette) . . . . 20 DM

## FILL-UTILITY für den C-16/116

Verglichen mit dem C-64 verfügt der C 16/116 zwar bereits über ein sehr starkes BASIC, aber auch bei ihm fehlen ein paar nützliche Befehle, so z.B. eine Routine zum Füllen beliebiger Bildschirmausschnitte mit einem bestimmten Zeichen oder einer bestimmten Farbe. Genau dafür ist die FILL-UTILITY gedacht.

Nach dem Eingeben und Starten des BASIC-Loaders (das Listing) wird das Maschinenprogramm zunächst eingelesen. Nach kurzer Zeit meldet sich das Loader-Programm mit der Frage, ob das reine Maschinenprogramm „FILL-UTILITY“ auf Diskette oder auf Kassette gespeichert werden soll (der Loader wird danach nicht mehr benötigt und kann gelöscht werden). Sie können die Maschinenroutine dann, wenn Sie sie brauchen, mit **LOAD „FILL-UTILITY“,8,1** (bzw. **.1,1** von Kassette)

einladen (die Sekundäradresse darf nicht fehlen) und mit **POKE 55,0: POKE 56,62** vor Überschreiten durch BASIC schützen. Die Routine wird gestartet mit dem Befehl **SYS 16128,X1,Y1,X2,Y2,F(L),Z**.

X1/Y1 und X2/Y2 sind dabei die Koordinaten zweier sich diagonal gegenüberliegender Eckpunkte des rechteckigen Bildschirmausschnittes, der gefüllt werden soll (X muß im Bereich von 0 bis 39 liegen, Y im Bereich von 0 bis 24). Der Parameter F bestimmt, ob der Bildschirmausschnitt mit einem Zeichen **F = 0**) oder einer Farbe (**F > 0**) gefüllt werden soll. Wird mit einer Farbe gefüllt, dann muß zusätzlich noch die **Luminanz L** der Farbe hinter F angegeben werden. Der Parameter Z schließlich bestimmt das Zeichen (siehe Bildschirm-Codes im Handbuch) bzw. die Farbe (entsprechend den im Handbuch bei COLOR

angegebenen Farbwerten), mit dem bzw. mit der der gewählte Bildschirmbereich gefüllt werden soll. Addiert man zu einem Farbwert noch **128** dazu so erscheinen die im betreffenden Bildschirmbereich befindlichen Zeichen blinkend (FLASH-Modus).

Zwei Beispiele:

**SYS 16128,1,138,23,0,1** füllt bis auf einen 1 Zeichen breiten Rahmen an allen vier Seiten den ganzen Bildschirm mit dem Buchstaben 1.

**SYS 16128,1,1,38,23,1,3,6+128** füllt den eben mit dem Buchstaben A gefüllten Bildschirmbereich mit der Farbe grün und läßt alle in diesem Bereich befindlichen Zeichen blinkend erscheinen.

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

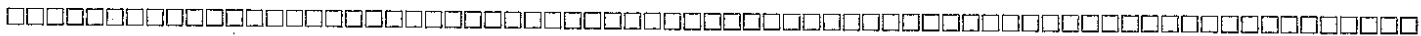
### C-16 LISTING + CHECKSUMMEN (DC V1.0)

10 REM *****	<121>	155 IFA#="D" THEN POKE 16294,8:GOTO 170	<83>
15 REM *	<112>	160 IFA#="K" THEN POKE 16294,1:GOTO 170	<18>
20 REM * FILL-UTILITY	<42>	165 GOTO 145	<0>
25 REM *	<122>	170 SYS 16293:END	<249>
30 REM * EINGABEFORM:	<236>	175 :	<233>
35 REM *	<132>	180 :	<238>
40 REM * SYS 16128,X1,Y1,X2,Y2,F(L),Z	<46>	185 :	<243>
45 REM *	<142>	190 :	<248>
50 REM * SCHÜTZEN VOR ÜBERSCHREIBEN	<196>	195 DATA 32,216,157,224,40,144,3,76,28,15	
55 REM * DURCH BASIC:	<122>	3,134,216,32,216,157,224,25,176,244	<226>
60 REM *	<157>	200 DATA 134,217,32,216,157,224,40,176,23	
65 REM * POKE 55,0 : POKE 56,62	<172>	5,134,218,32,216,157,224,25,176,226	<102>
70 REM *	<167>	205 DATA 134,219,165,219,24,229,217,16,13	
75 REM * 1986 BY DANIEL DURSTEWITZ	<167>	,165,217,72,165,219,133,217,104,133	<203>
80 REM * REICHENBERGER STR. 26	<144>	210 DATA 219,76,39,63,133,219,165,218,24,	
85 REM * 4460 NORDHORN	<166>	229,216,16,13,165,216,72,165,218	<78>
90 REM * TEL. 05921/79200	<20>	215 DATA 133,216,104,133,218,76,61,63,133	
95 REM *	<192>	,218,32,216,157,160,12,224,0,240	<100>
99 REM *****	<210>	220 DATA 15,32,216,157,224,8,176,164,138,	
105 :	<163>	10,10,10,10,170,160,8,134,221,166	<5>
110 :	<168>	225 DATA 217,165,216,132,217,202,48,9,24,	
115 :	<173>	105,40,144,248,230,217,208,244,133	<177>
120 :	<178>	230 DATA 216,32,216,157,138,5,221,166,219	
125 FORT=16128TO16325:READA:POKET,A:F=P+A:NEXT	<232>	,232,164,218,200,145,216,136,16,251	<51>
130 IFF=27239GOTO 145	<106>	235 DATA 72,165,216,24,105,40,144,2,230,2	
135 PRINT "DOWN" FEHLER (SPACE) IN (SPACE) DE		17,133,216,104,202,16,232,76,220	<193>
N (SPACE) DATAS! (SPACE) BITTE (SPACE) UEBERPR		240 DATA 139	<82>
UEFEN"	<221>	245 :	<47>
140 PRINT "SIE (SPACE) DIE (SPACE) ZEILEN (SPA		250 :	<52>
CE) 195-275 (SPACE) NOCH (SPACE) EINMAL!":END	<195>	255 REM SAVE-ROUTINE	<253>
145 PRINT "DOWN" SOLL (SPACE) DAS (SPACE) PRO		260 :	<63>
GRAMM (SPACE) AUF (SPACE) RYSON) D (RVS OFF) ISK	<23>	265 :	<68>
ETTE (SPACE) ODER"		270 DATA 162,1,160,1,32,49,244,169,12,162	
150 INPUT "AUF (SPACE) RYSON) K (RVS OFF) ASSET		,95,160,16,32,12,244,169,0,133,216	<59>
TE (SPACE) ABGESPEICHERT (SPACE) WERDEN":A#	<112>	275 DATA 169,63,133,217,169,216,162,165,1	
		60,63,76,148,241	<183>
		ENDE DES LISTINGS	

# 6502-Assembler-Kurs

## Teil 17

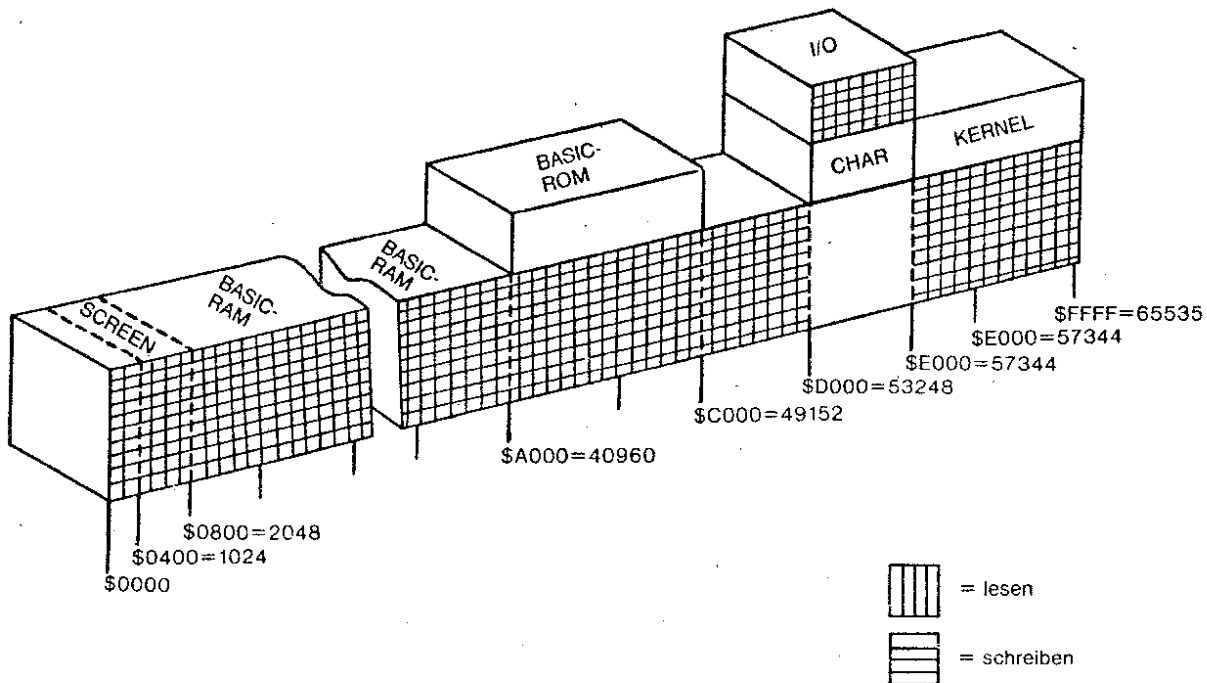
Im letzten Teil unseres Kurses haben wir bereits begonnen, die Belegung des C64 Speicher's im Zusammenhang mit dem Prozessor-Port zu betrachten. Ich möchte dies nun fortsetzen, und anhand einiger Skizzen erläutern.



LORAM=1 HIRAM=0 CHAREN=1 EXROM=? GAME=1 POKE 1,5  
 \$0000-\$CFFF Gesamte Bereich auf RAM geschaltet  
 \$0000-\$D7FF Videocontroller und Soundchip  
 \$D800-\$DBFF Farbram  
 \$DC00-\$DFFF I/O-Bausteine CIA 1/2  
 \$E000-\$FFFF Betriebssystem

umgeschaltet. Für Basic Programme oder Befehlerweiterungen kann dieser Modus nicht verwendet werden, da die Basic- sowie die ROM-routinen nicht mehr zugänglich sind. Einige Programme erreichen allerdings durch geschickte Umschaltung der Betriebsarten, eine Art Bank-Switching. Auf diese Weise lassen sich ROM und RAM quasi gleichzeitig nutzen. Ansonsten ist diese Betriebsart sehr gut zum betreiben eines zweiten Prozessors geeignet. Dieser kann beispielweise, vom Modulport aus, direkt auf das RAM sowie die I/O Bausteine des C64 zugreifen.

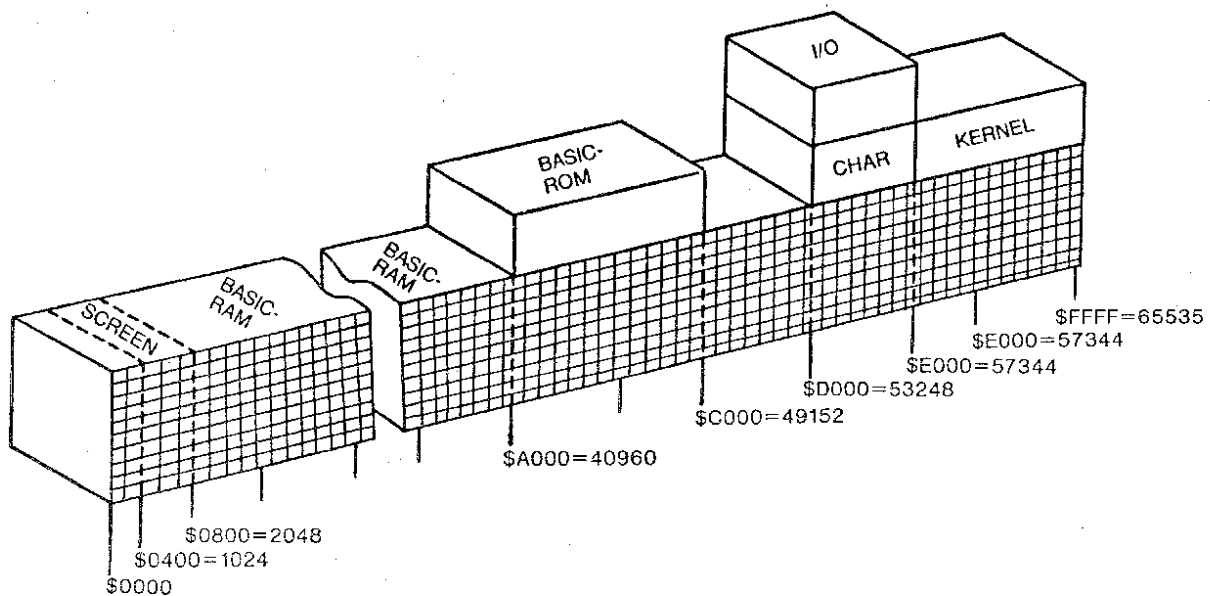
In dieser Betriebsart wird fast der gesamte Speicher des C16 auf RAM



LORAM=0 HIRAM=0 CHAREN=? EXROM=1 GAME=1 POKE 1,0  
 \$0000-\$FFFF Gesamter Bereich wird auf RAM geschaltet

Basic- und das System-ROM ausgeblendet, sondern zusätzlich auch die I/O Bausteine. Dadurch wird praktisch der gesamte Adressraum des Microprozessors 6510 durch RAM unterlegt. Dies ist übrigens die einzige Betriebsart, welche auch erlaubt, den RAM-Speicher unter dem Zeichengenerator sowie den I/O Bausteinen zu nutzen.

Dies ist sicherlich der stärkste POKE. Denn hier wird nicht nur das



LORAM=1 HIRAM=1 CHAREN=1 EXROM=0 GAME=0  
 POKE 1,7 (Externes Signal)  
 \$0000-\$7FFF RAM Speicher für Basicprogramme und Systemvektoren  
 \$8000-\$BFFF Externes Modul, beispielsweise Programmiersprache  
 \$C000-\$CFFF Freier Speicherbereich für MC-Programme  
 \$D000-\$D7FF Videocontroller und Soundchip  
 \$D800-\$DBFF Farbram  
 \$DC00-\$DFFF I/O Bausteine CIA 1/2  
 \$E000-\$FFFF Betriebssystem

Durch das Einstecken einer Befehls-erweiterung oder einer anderen Programmiersprache kommt diese Konfiguration zustande. Der 16KByte Bereich von \$8000 bis \$BFFF wird nun vom Modul belegt. Der darunter liegende RAM-Bereich läßt sich nicht mehr ansprechen.



LORAM=0 HIRAM=1 CHAREN=1 EXROM=0 GAME=0  
 POKE 1,6 (Externes Signal)  
 \$0000-\$9FFF RAM Bereich  
 \$A000-\$BFFF 8kByte ROM Einschub  
 \$C000-\$CFFF Freier Speicherbereich für MC-Programme  
 \$D000-\$D7FF Videocontroller und Soundchip  
 \$D800-\$DBFF Farbram  
 \$DC00-\$DFFF I/O Bausteine CIA 1/2  
 \$E000-\$FFFF Betriebssystem

Dies ist die typische Spielekonfiguration, wenn ein Modul (z.B. Soccer) eingesteckt ist.



LORAM=1 HIRAM=1 CHAREN=1 EXROM=0 GAME=1 POKE 1,7 (Externes Signal)  
 \$0000-\$7FFF RAM Bereich für Basic Programme  
 \$8000-\$9FFF ROM Modul, beispielsweise Basic-Erweiterung  
 \$A000-\$BFFF Basicinterpreter  
 \$C000-\$CFFF Freier Speicherbereich für MC-Programme  
 \$D000-\$D7FF Videcontroller und Soundchip

\$D800-\$DBFF Farbram  
 \$DC00-\$DFFF I/O Bausteine CIA 1/2  
 \$E000-\$FFFF Betriebssystem

Auch diese Konfiguration kommt durch das Einstecken eines Moduls zustande. Diesmal belegt das Modul nur 8 kByte, von \$8000-\$9FFF. In der Regel handelt es sich hier um eine Basic-Erweiterung oder um ein ähnliches Programm. Durch eine bestimmte Code-Sequenz am Anfang des Moduls, wird dieses automatisch nach dem Einschalten des Rechners aktiviert. Somit stehen dem Anwender sofort die Funktionen der Erweiterung zur Verfügung.

Wir haben nun alle möglichen Systemkonfigurationen des C64 kennengelernt. Wir haben an der Anzahl sowie an dem Aufbau der Speicher-Organisation erkannt wie flexibel der C64 eigentlich aufgebaut ist. Um jedoch alle diese Fähigkeiten, welche besonders bei der Grafik-Programmierung nützlich sind, ausnutzen zu können, bedarf es sehr guter Kenntnisse der Maschinensprache. Wir wollen uns deshalb vorerst nur mit der Einschalt-Konfiguration beschäftigen.



## Arbeitsweise von Betriebssystem und Basicinterpreter

Nach dem Einschalten des Rechners oder nach einem Reset springt der Rechner grundsätzlich an die Adresse, welche in den Speicherstellen \$FFFC angegeben ist. Das bedeutet, daß die Adresse im Betriebssystem kontrolliert wird. Bei dem C64 steht in diesen beiden Bytes die Adresse \$FCE2. Ab dieser Adresse steht die sogenannte RESET-Routine im Betriebssystem. Diese Routine ist dafür verantwortlich, daß alle Vektoren, Register und I/O Bausteine in den richtigen Zustand gebracht werden. Ohne diese Routine wäre der C64 praktisch tot, das heißt man könnte, wenn man von der Kontrollampe absieht, nicht feststellen, ob der Rechner ein- oder ausgeschaltet ist. Nach dem nun Vektoren, I/O Bausteine durch diese Routine initialisiert wurden, wird der sogenannte Basic-Interpreter aufgerufen. Dieser übernimmt nun, in Zusammenarbeit mit dem Betriebssystem, die gesamte Steuerung und Tastaturabfrage. Das bedeutet, daß nach der Ausgabe der Einschaltmeldung alle Eingaben direkt zum Basic-Interpreter gelangen. Dieser erlaubt beispielsweise das Eingeben, Löschen oder Editieren von Basic-Zeilen. Auch die direkte Eingabe von Basic-Befehlen wird hier abgearbeitet.



# programme

## Was nutzt uns der Basic-Interpreter für Maschinenspracheprogramme ?

Diese Frage ist relativ einfach zu beantworten. Der Basic-Interpreter sowie das Betriebssystem sind aus zahlreichen kleinen und größeren Unterprogrammen zusammengesetzt. Diese Unterprogramme lassen sich oft auch sinnvoll von eigenen Entwicklungen aus nutzen. Beispielsweise gibt es Routinen zur Ausgabe einer Zahl oder einer Meldung auf dem Bildschirm. Nutzt man kräftig alle Routinen dieser beiden ROM's, so kann man sein MC-Programm stark verkürzen. Nachteil dieser Methode ist jedoch, daß Programme durch die Verwendung von ROM-Routinen nur sehr schwer auf andere Rechner zu übertragen sind. Da jedoch viele Programme ohnehin stark auf den C64 zugeschnitten sind, läßt sich dieser Nachteil jedoch in Kauf nehmen. Wir werden im Verlaufe unseres Kurses alle interessanten ROM-Routinen kennenlernen. Neben dieser Möglichkeit, die ROM's auszunutzen, gibt es noch eine weitere interessante Möglichkeit. Für viele Funktionen des Betriebssystems oder des Basic-Interpreters, werden freie Speicherstellen im RAM, sprich Variablen, gebraucht. Eine Eigenheit des C64 ist es, diese durch zahlreiche Adressierungsarten vorwiegend in der Zero-Page abzulegen. Diese lassen sich somit vom Basic- sowie auch von Maschinensprache direkt beeinflussen. Besonders interessant ist dies, wenn man kleinere Befehlserweiterungen oder ähnliches in den Basic-Interpreter einbinden möchten. Um die sogenannte ZERO-PAGE, sowie die darüber liegenden Vektoren, nutzen zu können, beginnen wir hier mit der Beschreibung der einzelnen Adressen sowie deren Belegung und Funktion. Im Anschluß daran werden wir mit einem kurzen Programm demonstrieren, wie sich dies Adressen ausnutzen lassen.

\* \* \*

### Die Belegung der C64 Zero-Page

Adresse HEX	Dezimal	Belegung
\$00	0	Datenrichtungsregister für Prozessorport
\$01	1	Prozessorregister
\$02	2	ungenutzt
\$03-\$04	3-4	Vektor zur Umwandlung Fließkomma nach Fest
\$05-\$06	5-6	Vektor zur Umwandlung Fest nach Fließkomma
\$07	7	Suchzeichen
\$08	8	Hochkomma-Flag
\$09	9	Spalte bei TAB-Befehl
\$0A	10	Flag 0=Load 1=Verify
\$0B	11	Zeiger in Eingabepuffer
\$0C	12	Flag für DIM-Befehl
\$0D	13	Typeflag \$00=numerisch, \$FF=String
\$0E	14	Flag \$80=Integer, \$00=Real
\$0F	15	Hochkomma-Flag bei List
\$10	16	Flag für FN
\$11	17	Flag für INPUT \$00, GET \$40, READ \$98
\$12	18	Vorzeichen bei ATN

\$13	19	aktives I/O-Gerät
\$14-\$15	20-21	Integer-Adresse, Zeilennummer
\$16	22	Zeiger auf Stringstack
\$17-\$18	23-24	Zeiger auf zuletzt verwendeten String
\$19-\$21	25-33	Stringstack
\$22-\$25	34-37	Universal Zeiger
\$26-\$2A	38-42	Register für Funktionsauswertung
\$2B-\$2C	43-44	Zeiger auf Basic-Programmstart
\$2D-\$2E	45-46	Zeiger auf Start der Variablen
\$2F-\$30	47-48	Zeiger auf Start der Arrays
\$31-\$32	49-50	Zeiger auf Ende der Arrays
\$33-\$34	51-52	Zeiger auf Start der Strings
\$35-\$36	53-54	Hilfszeiger
\$37-\$38	55-56	Zeiger auf Basic-RAM Ende
\$39-\$3A	57-58	Basic-Zeile welche abgearbeitet wird
\$3B-\$3C	59-60	vorige Basic-Zeile
\$3D-\$3E	61-62	Zeiger auf nächsten Basic-Befehl
\$3F-\$40	63-64	Data-Zeile welche abgefragt wird
\$41-\$42	65-66	Zeiger auf nächste Data-Zahl
\$43-\$44	67-68	Zeiger auf Herkunft der Eingabe
\$45-\$46	69-70	Variablenname
\$47-\$48	71-72	Variablenadresse
\$49-\$4A	73-74	Variablenzeiger für FOR/NEXT
\$4B-\$4C	75-76	Zwischenspeicher für Programmzeiger
\$4D	77	Maske für Vergleichsoperationen
\$4E-\$4F	78-79	Zeiger für FN-Funktion
\$50-\$53	80-83	Stringdescriptor
\$54	84	Konstante \$4C für JMP-Befehl
\$55-\$56	85-86	Sprungvektor für Funktionen
\$57-\$5B	87-91	Arithmetik Akku 3
\$5C-\$60	92-96	Arithmetik Akku 4
\$61-\$65	97-101	Fließkommaakku 1, FAC

An dieser Stelle unterbrechen wir die Beschreibung der Zero-Page. Im nächsten Teil setzen wir diese fort, und zeigen an einem Beispiel deren Anwendung. Frank Brall

### Verwendete Literatur:

- 6502 Microcomputer-Programmierung, Peter Heuer, Hofacker Verlag
- 64 Intern, Angershausen, Becker, Englisch, Gerits, Data Becker Buch
- C64 Programmieren in Maschinensprache, Winfried Kassera, Frank Kassera, Markt & Technik
- Profihandbuch, H.L. Schneider, Werner Eberl, Markt und Technik



**In der nächsten Ausgabe:**  
**Ein weiterer Kurs!**  
**Interruptprogrammierung des C-64**

## Checksummer C Version 1.0 für Commodore 64

Checksummer v1.0 ist ein Prüfsummen-Programm, das die Eingabe von Programm-Listings zum wahren Vergnügen macht. Tippfehler werden schon während des Eingabens der einzelnen Programmzeilen erkannt. Dieses System, zusammen mit einem neuen Druckverfahren, gewährleisten, daß unsere Listings zu 99,9% fehlerfrei abgedruckt werden.

**Arbeitsweise und Aufbau unseres Checksummers:**

Unser Checksummer besteht aus einem

kleinem Maschinenprogramm, welches als Basic-Loader abgedruckt ist.

Tippen Sie diesen Loader ein und speichern ihn auf Kassette oder Diskette ab; Sie können ihn zukünftig immer wieder benutzen. Der Start erfolgt durch den Befehl „RUN“. Nach kurzer Zeit meldet sich der Rechner mit der Meldung „TRONIC ..“. Der Checksummer ist nun aktiv. Nun kann man ein beliebiges Tronic-Listing eingeben. Nachdem eine Zeile mit RETURN abgeschlossen wird, erscheint links oben auf dem Schirm eine Prüfwahl. Vergleichen Sie diese mit der Zahl, welche im Heft hinter diese Zeile abgedruckt ist. Stimmen die Zahlen überein, so ist die Zeile richtig eingegeben; ansonsten muß sich noch ein Fehler in Ihre Eingabe eingeschlichen haben, und Sie müssen die Zeile korrigieren.

Auf diese Weise können Sie das gesamte Listing schnell und fehlerfrei eingeben. Interessant ist auch, daß bei der Eingabe von Zeilen die üblichen Abkürzungen benutzt werden können, ohne die Checksumme zu verändern. Leerzeichen außerhalb von Anführungszeichen werden ignoriert, da diese auf die Ausführung der einzelnen Befehle keinen Einfluß haben. Sie können Ihr Programm auch starten, denn der Checksummer und Ihr Programm beeinflussen sich nicht gegenseitig. Wollen Sie den Checksummer abschalten, so geben Sie einfach „POKE 1,55“ ein oder betätigen die Tasten-Kombination „RUNSTOP und RESTORE“. Aktivieren können Sie den Checksummer jederzeit (auch nach RESET) mit „POKE 1,53“.

### C64 LISTING MIT CHECKSUMMEN (C V1.0)

0 REM ***** TRONIC CHECKSUMMER *****	<116>	15 DATA73,67,45,86,69,82,76,65,71,32,67,	<129>
1 REM VERSION C 1.0 BEI FRANK BRALL	<10>	72,69,67,75,83,85,77	
2 REM -----	<148>	16 DATA77,69,82,32,35,35,35,139,13,13	<147>
3 FOR I=832 TO 1008:READ A:S=S+A	<68>	32,32,32,32,32,32,32	
4 POKE I,A:NEXT I:IF S<>16397 THEN PRINT		17 DATA32,32,32,32,32,86,69,82,83,73,79,	<79>
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE		78,32,67,32,49,46,48	
!{SPACE}(10-19)":STOP	<32>	18 DATA13,13,32,32,32,32,32,40,67,41,	<94>
5 SYS 832:S=0:FOR I=58464 TO 58603:READ		32,70,82,65,78,75,32	
A:S=S+A	<21>	19 DATA66,82,65,76,76,32,40,49,48,46,56,	<12>
6 POKE I,A:NEXT I:IF S<>18919 THEN PRINT		53,41,13,0	
"FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE		29 DATA169,0,141,107,191,141,108,191,141	<7>
!{SPACE}(29-36)":STOP	<219>	,109,191,160,2,24,177,95,200,113	
7 S=0:FOR I=48979 TO 49005:READ A:S=S+A	<33>	30 DATA95,141,109,191,160,3,200,177,95,2	<91>
8 POKE I,A:NEXT I:IF S<>2888 THEN PRINT"		40,44,201,34,208,10,173,107,191	
FEHLER{SPACE}IN{SPACE}DATA-ZEILEN{SPACE}	<23>	31 DATA73,1,141,107,191,177,95,174,107,1	<128>
!{SPACE}(46-47)":STOP		91,208,4,201,32,240,228,238,108	
9 PRINT"({DDHN2})AKTIVIEREN{SPACE}:{SPACE}		32 DATA191,174,108,191,24,177,95,109,109	<244>
POKE{SPACE}1,53{SPACE18}AUSSCHALTEN:{SPA	<26>	,191,141,109,191,202,208,244,76,120	
CE}POKE{SPACE}1,55":NEW		33 DATA228,56,76,183,228,72,32,201,255,1	<101>
10 DATA169,0,133,254,162,1,189,137,3,133	<176>	70,104,144,1,138,96,32,240,255	
,255,160,0,177,254,145,254,136		34 DATA142,105,191,140,106,191,162,0,160	<89>
11 DATA208,249,230,255,165,255,221,139,3	<147>	,0,24,32,240,255,169,91,32,12	
,208,238,202,16,230,169,96,141,49		35 DATA225,169,0,174,109,191,32,205,189,	<239>
12 DATA165,169,228,141,50,165,169,53,133	<225>	169,93,76,83,191,173,33,208,145	
,1,169,141,133,254,162,3,134,255		36 DATA243,96,105,2,164,145,200,208,4,19	<202>
13 DATA160,0,177,254,240,7,32,202,241,20	<140>	7,161,208,247,96	
0,76,120,3,169,2,141,32,208		46 DATA32,12,225,32,63,171,32,63,171,172	<213>
14 DATA96,160,224,192,0,147,17,32,32,35,	<158>	,106,191,174,105,191,24,32,240	
35,35,35,32,84,82,79,78		47 DATA255,76,128,164,24,0,0,5,200	<65>
		ENDE DES LISTINGS	

# Checksummer für COMMODORE 64

Wer den Checksummer nicht eingeben möchte, kann diesen auch unter der folgenden Bestellnummer beziehen:

Bestell-Nr. CV10K/Kassette 10 DM, Bestell-Nr. CV10D/Diskette 15 DM

## Zeitschriften und Programme von 1985/86

### Heft 5/85

**Commodore**  
 Bestell-Nr. COM CK 1/1 16,- DM  
 Bestell-Nr. COM CD 1/1 20,- DM  
 Senso (C-16/116), Jungle (C-64), Auto-start (C-64), Sequenzer (C-64), Sufo (C-64), Olympia (VC-20), Flowers (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 1/1 16,- DM

Cobra, Krümi, Haunted Castle, Q-Bert, ROM-CALLS

### Heft 9/85

**Commodore**  
 Bestell-Nr. COM CK 5/1 16,- DM  
 Bestell-Nr. COM CD 5/1 20,- DM  
 Superoni 64 (C-64), Galaxy (C-64), Atlantis (C-64), Shaolin (C-16), Hospital (VC-20), Flipper-Ball (VC-20), Alien (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 5/1 16,- DM  
 Bestell-Nr. COM SD 5/1 39,- DM  
 Duell, Crash, Citylander, Klondike, REM-Killer

### Heft 1/86

**Commodore**  
 Bestell-Nr. COM CK 1/2 16,- DM  
 Bestell-Nr. COM CD 1/2 20,- DM  
 Checksummer 1.0 C, Astronaut Battle (C-64), Hueycobra (C-16/116), Hardcopy-routine (C-16/116), Helmut und die Zeitmaschine (C-16/116), Locos (VC-20), Rock'eme (VC-20), Charly der Raketenmann (VC-20), Interrupt-Programmierung (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 1/2 16,- DM  
 Bestell-Nr. COM SD 1/2 39,- DM  
 Defender, Boulder Dash, Basic-Befehle auf den Funktionstasten

### Heft 5/86

**Commodore**  
 Bestell-Nr. COM CK 5/2 16,- DM  
 Bestell-Nr. COM CD 5/2 20,- DM  
 Space-Devil (C-64), Disk-Eddi (C-64), ABC-Legen (V-20), 3-D-Labyrinth (VC-20), Tron (VC-20), Invaders of Space (C-16/116), Submarine (C-16/116), Balkendiagramm (C-16/116)

**Schneider**  
 Bestell-Nr. COM SK 5/2 16,- DM  
 Bestell-Nr. COM SD 5/2 39,- DM  
 Startank, Grafik-Designer, Cross-Reference, Itext

### Heft 9/86

**Commodore**  
 Bestell-Nr. COM CK 9/2 16,- DM  
 Bestell-Nr. COM CD 9/2 20,- DM  
 Moonpatrol (VC-20), Falcons Lander (VC-20), Frogger (VC-20), The Gammas (C-64), Cave of Death (C-64) Sprites and Sound mit dem UMH-System (C-64), Diggy (C-16/116), Tapedirectory (C-16/116), Reversi (C-16/116)

### Heft 6/85

**Commodore**  
 Bestell-Nr. COM CK 2/1 16,- DM  
 Bestell-Nr. COM CD 2/1 20,- DM  
 Mampfman (C-64), Softlanding (C-64), Karate (C-64), Einbrecher (VC-20), Tanker (VC-20), Jupiter Rescue (VC-20), SOS-Schiffbruch (C-16)

**Schneider**  
 Bestell-Nr. COM SK 2/1 16,- DM

Tastaturbelegung, Cave-Runner, Old-Shurehand

### Heft 10/85

**Commodore**  
 Bestell-Nr. COM CK 6/1 16,- DM  
 Bestell-Nr. COM CD 6/1 20,- DM  
 The Castle of Dr. Creep (VC-20), Catchgo (VC-20), Sub-Zone (VC-20), The Maze Graze Game (C-64), Super-Breakout (C-64), Planet of Death (C-64), Funktionstasten-Programmierung (C-64), Car Race (C-16/116), Key-Helper (C-16/116)

**Schneider**  
 Bestell-Nr. COM SK 6/1 16,- DM  
 Bestell-Nr. COM SD 6/1 39,- DM  
 Labyrinth, Catch, Mini Car Race, Inka Sogra, Single Disk Backup

### Heft 2/86

**Commodore**  
 Bestell-Nr. COM CK 2/2 16,- DM  
 Bestell-Nr. COM CD 2/2 20,- DM  
 Robo (C-64), Basic-Befehlsweiterung (C-64), Raketenstart (C-16/116), Dreikampf (C-16/116), Headline (C-16/116), Defender (VC-20), Formel 1 (VC-20), Memory-Display (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 2/2 16,- DM  
 Bestell-Nr. COM SD 2/2 39,- DM  
 Alien 2001, Diamond Hunter, Directory

### Heft 6/86

**Commodore**  
 Bestell-Nr. COM CK 6/2 16,- DM  
 Bestell-Nr. COM CD 6/2 20,- DM  
 Music-Master (C-64), The little Puck goes home (C-64), EDDY (C-64), Invers-Utility (C-16/116), Checksummer (C-16/116), Affenwurf (C-16/116), Data-Maker (VC-20), Karate-Kid (VC-20), Q-Bert (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 6/2 16,- DM  
 Bestell-Nr. COM SD 6/2 39,- DM  
 Diskort, Mini-Forth, Starcommand

### Heft 7/85

**Commodore**  
 Bestell-Nr. COM CK 3/1 16,- DM  
 Bestell-Nr. COM CD 3/1 20,- DM  
 Sea-Attak (C-64), Sace-Age (C-64), Micro-Tools (C-64), Disky 1.0 (C-64), Stuntman (VC-20), Sepp (VC-20), Balloon-Shoot (VC-20), Farbenfroh (C-16)

**Schneider**  
 Bestell-Nr. COM SK 3/1 16,- DM  
 Bestell-Nr. COM SD 3/1 39,- DM  
 Secret Valley, Smileyamat, Super-File

### Heft 11/85

**Commodore**  
 Bestell-Nr. COM CK 7/1 16,- DM  
 Bestell-Nr. COM CD 7/1 20,- DM  
 Interpreter Music-Language (C-64), Firelegion (C-64), Funktionstastenprogrammierung (C-64), Schatztaucher (C-16), Dangerous Planet (VC-20), Geo-Zaxxon (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 7/1 16,- DM  
 Bestell-Nr. COM SD 7/1 39,- DM  
 Etis, Alien-Hunter, Demon-Attack, Datenverwaltung, Discshow

### Heft 3/86

**Commodore**  
 Bestell-Nr. COM CK 3/2 16,- DM  
 Bestell-Nr. COM CD 3/2 20,- DM  
 Minor-Jumper (C-16/116), Hero (C-16/116), Data-Maker (C-16/116), Befehlsweiterung (C-64), Starchip (C-64), Mini-Car-Race (C-64), Text-Scroller (C-64), Text-Editor (C-64), Checksummer (VC-20), Robin at Ravenwreck (VC-20), Mini-Karate (VC-20), Autonumber-Routine (VC-20)

**Schneider**  
 Bestell-Nr. COM SK 3/2 16,- DM  
 Bestell-Nr. COM SD 3/2 39,- DM  
 Smily, Screpi, Die verrückte Fabrik, Disc-Help

### Heft 7/86

**Commodore**  
 Bestell-Nr. COM CK 7/2 16,- DM  
 Bestell-Nr. COM CD 7/2 20,- DM  
 VC-20-Sound-Demo, Mad Rush (VC-20), Ghost-Hell (VC-20), Roadrunner (C-64), Basic Extension (C-64), Meteoritensturm (C-64), Checksummer C-16/116, Airwolf (C-16/116), Break-Out (C-16/116)

**Schneider**  
 Bestell-Nr. COM SK 7/2 16,- DM  
 Bestell-Nr. COM SD 7/2 39,- DM  
 Poker, Cave-Walker, Diskettenarchiv

### Heft 8/85

**Commodore**  
 Bestell-Nr. COM CK 4/1 16,- DM  
 Bestell-Nr. COM CD 4/1 20,- DM  
 Drucker-List (VC-20), Flowers II (VC-20), Dino Eggs (VC-20), Piano (C-16/116), Tank-Duell (C-64), Vampir (C-64), Memory (C-64), Raster Register, Interrupt (C-64)

**Schneider**  
 Bestell-Nr. COM SK 4/1 16,- DM  
 Bestell-Nr. COM SD 4/1 39,- DM  
 Copter Patrol, Suchrätsel, Tower

### Heft 12/85

**Commodore**  
 Bestell-Nr. COM CK 8/1 16,- DM  
 Bestell-Nr. COM CD 8/1 20,- DM  
 Freggias Nightmare (VC-20), Wächter des heiligen Tempels (VC-20), Jumping Joe (C-16/116), City-Runner (C-16/116), Mission Rescue (C-64), Turbo-Jet (C-64), Print-Generator (C-64), Datagenerator (C-64)

**Schneider**  
 Bestell-Nr. COM SK 8/1 16,- DM  
 Bestell-Nr. COM SD 8/1 39,- DM  
 Checksummer 1.0 CPC, Two Cherry Chaps, Erwin im Zauberschloß, Drucksteuerung NLQ 401

### Heft 4/86

**Commodore**  
 Bestell-Nr. COM CK 4/2 16,- DM  
 Bestell-Nr. COM CD 4/2 20,- DM  
 Shuttle (VC-20), Josef in den Nöhlen (VC-20), Merge-Routine (VC-20), Shoot it (VC-20), Fast Trust (C-64), Data-Text 64 (C-64), Gary, der schnelle Kabelverleger (C-16/116), SDI (C-16/116), Beep-Routine (C-16/116)

**Schneider**  
 Bestell-Nr. COM SK 4/2 16,- DM  
 Bestell-Nr. COM SD 4/2 39,- DM  
 Space-Rescue, Texteditor, Starfight, Programm-Retter

### Heft 8/86

**Commodore**  
 Bestell-Nr. COM CK 8/2 16,- DM  
 Bestell-Nr. COM CD 8/2 20,- DM  
 Death Desert (C-64), 3-D-Labyrinth (C-64), Klick (C-64), Bahnhof (VC-20), Omega Attack (VC-20), Sound (VC-20), Plattenlayout (C-16/116), Balloon Shoot (16/116), Hardcopy (C-16/116)

**Schneider**  
 Bestell-Nr. COM SK 8/2 16,- DM  
 Bestell-Nr. COM SD 8/2 39,- DM

### Aus diesem Heft:

**Commodore**  
 Bestell-Nr. COM CK 10/2 16,- DM  
 Bestell-Nr. COM CD 10/2 20,- DM  
**Schneider**  
 Bestell-Nr. SK 10/2 16,- DM  
 Bestell-Nr. SD 10/2 39,- DM

Alle hier aufgeführten Kassetten und Disketten können jederzeit nachbestellt werden. Bei den Zeitschriften sind die Ausgaben 5, 6, 7, 8 (86) und 5, 9, 10, 12 (85) noch vorhanden.

# software-service

Programm      Preis Kass./Disk.      Bestell-Nr.

## C=64

Multi-Key/S-Tool 64/Interrupt-Programme	16,--/21,--	UC 2/85
Sprithilfe/Diskloader/Directory	21,--/ -	UC 3.1/85
Tape-Directory/Asmon/Data-Generator	15,--/21,--	UC 3.2/85
Fast Load	10,--/ -	UC 4/84
Diskmonitor/Zeichensatz	20,--/ -	UC 5/85
Reassembler/Maskengenerator	15,--/20,--	UC 6/85
Del 64/Treiberprogramm für 1526/MPS 802	14,--/19,--	UC 7/85
Deceleator	14,--/19,--	UC 7/85
Supertapedirectory/Renew/Kontrollabfrage	15,--/ -	UC 8/85
Video-Utilities, Multi-Basic V3, Bildeditor	17,--/23,--	UC 9/85
Better Basic/Autounumber/Hardcopy/Terminal-programm	17,--/23,--	UC 10/85
Speicher-Oszilloskop/Makro-Assembler/Change Type	49,--/54,--	UC 1/86
Reactor/Concentration/Datenbank	17,50/23,50	HC/C-1
Warlords/Caverns of Death	14,50/19,50	HC/C-2
Alien-Destroyer/Duell	24,50/29,50	HC/C-3
Ocean Game/Tennis	17,50/23,50	HC/C-4
The Caves	17,50/23,50	HC/C-5
Gardener	17,50/23,50	HC/C-6
Ufo/Skateboard Sam	24,50/29,50	HC/C-1-4
Olympic Biathlon/Struggle for Life	24,50/29,50	HC/C-2-4
Pole Position/Save your teeth/Push	24,50/29,50	HC/C-3-4
Explorer 2/Programmname?	24,50/29,50	HC/C-4-4
Mauern/Widerstand	8,--/15,--	C 41
Space-Comets/Erdsalte/Sprite-Data	15,--/23,50	C 51
Autostart/Bestellschein/Roadpainter	16,50/23,50	C 61
Hardcopy/Space-Fighter/Data-Generator	15,50/19,50	C 71
Monster-Attack/Block-Painter/Epson-Drucker	16,50/23,50	C 81
Projekt/Datenbank	16,--/23,50	C 91
Spiders/The Basic	16,50/23,50	C 101
High Noon/Skeet/Grafik-Designer	17,50/23,50	C 121
Painter/Star-Baddle/Editor	17,50/23,50	C 22
Wüstenrally/Jet-Pac/Black Moore Castle	17,50/23,50	C 32
Brieftaube/Cadelon	19,50/24,50	C 42
Ritter Erik/Grand Prix Sprithilfe	19,50/24,50	C 52
Firebird/High Music	24,50/29,50	C 62
Moonsweeper/Scotti	24,50/29,50	C 72
Tron/Mercurious	24,50/29,50	C 13
Interceptor-Basel/Schotter	24,50/29,50	C 23
Jump-Man/Space-Taxi/The Way	24,50/29,50	C 33
Die Festung/Tanksimulation	24,50/29,50	HC/C-5-4
Fight Night/Monsterjagd/Collussus Mühle	24,50/29,50	C 63

## Neu in unserer Software-Bestelliste der

## C=16

Programm      Preis Kass./Disk.      Bestell-Nr.

Fantasy-Country/Horror-Castle	18,--/23,--	0 23
Moon Fighter/Goldrausch	18,--/23,--	0 33
Zeichendesigner	14,--/ -	UO 8/85
OLD-Routine-Merge-Routine/Change Type	14,--/19,--	UO 1/86
Frogger/Humor	18,--/23,--	0 43

Programm      Preis Kass./Disk.      Bestell-Nr.

## VIC-20

Bestellschein/Glücky	8,50/15,--	V 61
Multigraph/All Rammer	11,--/15,50	V 71
Zyklus/Meteorit	11,--/19,50	V 81
Garten/Schloß Gruselstein	14,--/19,50	V 91
Fressman/Outlaw	14,--/19,50	V 101
Prost/Bufalo Bill	14,--/19,50	V 121
Joy Man/Powerpack	14,--/19,50	V 22
Der rasende Malocher/Frankie goes to Pharao	14,--/19,50	V 32
Matron/Obst	14,--/19,50	V 42
Race On/Cagy	14,--/19,50	V 52
Nürburg 3D/Düsi	14,--/19,50	V 62
Breaker/Expulsion	14,--/19,50	V 72
Brückenbau/Jango	16,--/21,50	V 13
Inka-Schatz/Miner der Fassadenstreicher	16,--/21,50	V 23
Booby in Action/Bergshooting	16,--/21,50	V 33
Zyklus/Geldsammler	11,50/ -	HC/V-1
Star-Wars/Punktefresser	14,50/ -	HC/V-2
Catch the Fish/Mister Jump	14,50/ -	HC/V-3
Ghost's Eggs/Fox Hunt	14,50/ -	HC/V-4
Collector/Break Out	14,50/ -	HC/V-5
Q-Bert/Salvage Crew	16,50/ -	HC/V-6
Dog Fight/Schwitzbad	16,50/ -	HC/V-1-4
Earth Rescue	14,50/ -	HC/V-2-4
Tank	14,50/ -	HC-8-3-4
Terra 2/Pac-Ball	16,50/21,50	HC/V-4-4
Sprites/Space-Battle	13,--/19,--	UV 2/85
Grafik-Painter	10,--/ -	UV 5/85
VIC-Clock	9,--/ -	UV 6/85
Decelerator	9,--/ -	UV 7/85
Joypaint	12,--/ -	UV 8/85
Disassembler, Old-Programme	10,--/ -	9/85
Directory/Farb-JRQ	10,--/ -	UV 10/85
Mèn Rescue/Economy	14,50/ -	HC/V-5-4

Anleitungen zu unseren Programmen erhalten Sie aus unseren Heften. Bei ausverkauften Heften erhalten Sie zum Heftpreis Fotokopien der Spielbeschreibungen.

Programm      Preis Kass./Disk.      Bestell-Nr.

## Schneider

Autorennen	11,--/ -	US 3/85
Universal-Datei V1	9,--/ -	US 4/85
Computerschrift/Symbol-Swap	10,--/ -	US 5/85
Keyboard Toolkit/Farbdecodierung	11,--/ -	US 6.1/85
Aladin	9,--/ -	US 6.2/85
Painter/Box-Befehl	14,--/24,--	US 7/85
Maschinensprache-Monitor/Disk-Hilfe	15,--/25,--	US 8/85
Basic, Maschinen-Kit	12,--/22,--	US 9/85
Backup/Directory/Ellipse/Disk	/27,--	US 10/85
Data-Generator/Grafik Toolkit/ASCII List (Turbo Pascal)	15,--/25,--	US 1/86
Fallschirmspringer	9,--/ -	HC/SR-3
Geisterschloß	9,--/ -	HC/SR-4
Zeichendesigner	12,50/ -	HC/SR-5
Mini Car Race/Interceptor 3 D	17,50/26,50	HC/SR-6
Secret Agent	12,50/24,50	HC/SR-1-4
Space Invaders/Mini Pac/Otti's „Sound“	17,50/27,50	HC/SR-2-4
Samson und die Kisten/Die verflixte 15	17,50/27,50	HC/SR-3-4
Moon-Landing/Panzer	17,50/26,50	HC/SR-4-4
Super-Miner	14,--/24,--	SR 42
GPC-Bert	14,--/24,--	SR 52
Conan's Castle	15,50/25,50	SR 62
Snider's Mace	16,50/26,50	SR 72
Schneider Panik/Killership	18,50/28,50	SR 13
Horror-Caves/Midnight	18,50/28,50	SR 23
Cosmis-Castle/Sprinter-Mace/Traumland	18,50/28,50	SR 33
Mühle	12,50/24,50	HC/SR-5-4

# software service

Der Tronic-Verlag bietet Spitzensoftware für Jedermann. Überzeugen Sie sich von unseren Angeboten und bestellen Sie noch heute. Alle Bestellungen werden innerhalb nur

## 1 Woche

nach Eingang bearbeitet. Telefonische Bestellungen unter der Rufnummer

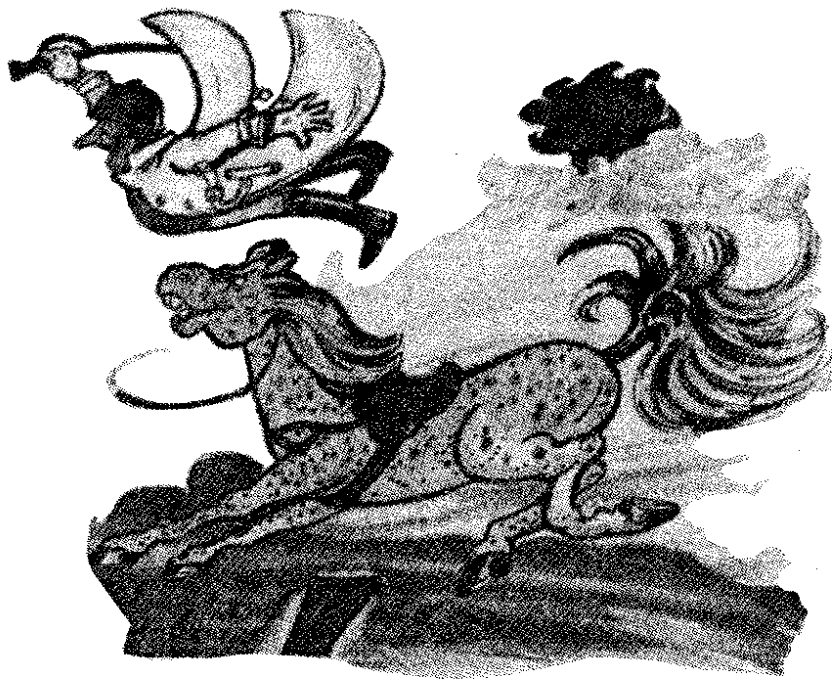
• 05651/30011



# Horse Games – Pferderennen

## Das Top-Spiel

C64



### HORSE GAMES

Verwetten Sie nicht Ihren ganzen Einsatz! Der C-64 ist wirklich stark!

„Horse Games“, das ist die Pferderennbahn auf Ihrem Computermonitor! Sie messen Ihre Disk-

jockeybegabung mit dem Computer. Wenn Sie das Titelbild sehen, gelangen Sie mit dem Feuerknopf in das erste Bild. Hier werden Ihre Fähigkeiten als Springreiter getestet. Ihre Aufgabe ist es, den Parcours zu meistern, ohne die Hindernisse zu berühren. Dazu lenken Sie das Pferd

statt mit den Zügeln nun mit dem Joystick, wobei ein Druck nach rechts Antraben und ein Druck nach links Stehenbleiben bedeutet. Joystick nach oben bewirkt einen Sprung.

Haben Sie alle 14 Hindernisse mehr oder weniger gut bewältigt,

erscheint die Anzeige mit der Auswertung.

Nach erneutem Drücken des Feuerknopfes beginnt die zweite Disziplin, das Galopp-Reiten. Hierbei treten Sie gegen den Computer an. Das Computerpferd erscheint immer in der Bild-

mitte; Ihr Pferd bewegt sich also immer in Relation zu diesem.

Obwohl die Steuerung dieser Disziplin sehr einfach ist, sind doch überlegte Strategien erforderlich. Joystick nach rechts bedeutet schneller, nach links langsamer. Am unteren Bildrand werden 3 Werte angezeigt. Die zurückgelegte Strecke, die Kraftreserven und die abgelaufene Zeit. Die Kraftreserven verändern sich, je nachdem ob Sie schneller oder langsamer reiten. Sinkt die Zahl unter 50, so werden Sie immer langsamer und müssen erst wieder Kräfte sammeln. Zu guter letzt legt das Computerpferd auf den letzten 300 Metern noch einen Endspurt ein - da müssen Sie schon genügend Kraftreserven haben, um da mithalten zu können. Nach Beendigung des Laufes wird Ihre Zeit angezeigt und Sie können nach Knopfdruck wieder neu beginnen.

Eingabehinweise:

Zunächst müssen die Programmteile HORSE1 bis HORSE5 einzeln abgetippt und gespeichert werden.

Danach gibt man NEW: POKE 56,80 ein und startet die Programme HORSE1 bis HORSE5 nacheinander. Wenn ein Programm fertig ist, so wird dies angezeigt und Sie müssen das nächste laden und starten. Nachdem alle Programme einmal durchgelaufen sind, geben Sie folgende Zeile ein: POKE56,160: POKE44,94: POKE 46,156: SAVE"HORSE DATA",8,1 (RETURN)

Dadurch wird das Programm "HORSE DATA" erzeugt, das später vom Hauptprogramm nachgeladen wird. Von nun an brauchen Sie nur noch das Programm HORSE GAMES zu laden und zu starten, da dieses das Maschinenprogramm nachlädt. Bei Datensettenbetrieb sind natürlich die Deviceadressen entsprechend zu ändern.

## Listing 1

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

1 AD=30000:F0R1=1000T01095:READA#,PS:C=0
:FORN=1TOLN(A#)STEP2 <36>
2 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A#,N
+1,1))-48:A=A+7*(A>9):B=B+7*(B>9) <161>
3 POKEAD,16*A+B:C=C+PEEK(AD):AD=AD+1 <176>
4 NEXT:PRINTI;"(SPACE)";:IFC>PSTHENPRIN
T"ERROR":STOP <157>
5 PRINT"OK":NEXT <143>
6 PRINT"FERTIG":END <136>
1000 DATA 7BA97F8D0DDCA9018D1AD0A9038502
ADDC02BD12D0A918BD11D0A9568D1403, 3357 <180>
1001 DATA A9758D15035860AD19D08D19D02901
F0034CEAD,16*A+B:C=C+PEEK(AD):AD=AD+1 <253>
1002 DATA A6028DC5028D21D0BDC028D2D08D
BB028D23D0ADEFO2F00AC901F0034CDB, 3875 <97>
1003 DATA 764CB376AD00DC2908F00AAD00DC29
04F00B4CAF75A9018DD8024CAF75A900, 3258 <163>
1004 DATA 8DB802ADD802D0034CDE75ADCF02F0
034CCC75E000F0034CCC75CED502CED5, 4102 <212>
1005 DATA 02ADD002F0034CDE75E001F0034CDE
75CED602EAE003F00CE000F008E001F0, 3998 <171>
1006 DATA 07E002F04F4CE47618ADD502C9D080
0AA9D78DD502200A76A9D78D16D04CE9, 3945 <200>
1007 DATA 76AED202A028BD006099FCFC2BD0061
99F7C2BD0062991FC3BD00639947C3BD, 3985 <147>
1008 DATA 006499698F3C3BD00639947C3BD008
EED20260CED602ADD602C9D0B00AA9D7, 4446 <223>
1009 DATA 8DD602205576A9D78D16D04CE976AE
D302A028BD00689973FC1BD00699967C1, 3806 <225>
1010 DATA BD0066998F8C1BD0067997B7C1BD006A
9917C199EFC0CAB8D0DBEED30260E002, 4387 <150>
1011 DATA F0034CE476CED602CED602ADD602C9
D0B00AA9D78DD60220A776A9D78D16D0, 4306 <74>
1012 DATA 4CE976AED302A028BD00689973FC1BD

```

```

00699767C1BD0066998F8C1BD00679987, 3867 <108>
1013 DATA C1BD006A9917C199EFC0CAB8D0DBEE
D302604C3B76E002F0034CE476AD7003, 4185 <222>
1014 DATA D0F1A9D28D16D0EAA602BDC0028D12
D08AF00668A868AA6840ADEF02F007C9, 4311 <2>
1015 DATA 01F01E4C4977204C7720E07720FB77
208E78201A7820357920947920D0794C, 2956 <174>
1016 DATA 4977203780204C7720E077205D7C20
F17C20097D20647D20BC7D20EA7D200C, 2858 <74>
1017 DATA 7E20287E20467E20677E4C31EAADD9
02D00160EEDA02ADDA02C901F018C902, 3389 <80>
1018 DATA F026C903F034C904F042C905F050A9
008DDA024CCE77A2148EFC3E8BEF9C3, 4327 <3>
1019 DATA E8BEFAC3E8BEFBC360A21C8EF8, 5639 <220>
1020 DATA C3E8BEF9C3E8BEFAC3E8BEFBC360A2
208E78201A7820357920947920D0794C, 2956 <78>
1021 DATA 60A2248EFC3E8BEF9C3E8BEFAC3E8
BEFBC360A228BEF8C3E8BEF9C3E8BEFA, 5661 <171>
1022 DATA C3E8BEFBC360ADD802D00160A5A229
03F006A9008DD90260A9018DD90260A9008DD802
1023 DATA D802F006A9018DD90260A9008DD802
A210BEFBC3E8BEF9C3E8BEFAC3E8BEFB, 4648 <159>
1024 DATA C360ADDC02D00160ADDD02F00FC901
F01DC902F02BC903F0394C7C78A22C8E, 3768 <187>
1025 DATA F8C3E8BEF9C3E8BEFAC3E8BEFBC360
A230BEFBC3E8BEF9C3E8BEFAC3E8BEFB, 6053 <149>
1026 DATA C360A234BEF8C3E8BEF9C3E8BEFAC3
E8BEFBC360A238BEF8C3E8BEF9C3E8BE, 5638 <122>
1027 DATA FAC3E8BEFBC360A23CBEF8C3E8BEF9
C3E8BEFAC3E8BEFBC360ADDC02D014AD, 5517 <182>
1028 DATA 00DC2901F00160A9018DDC028DD802
20F3786020C578A5A22907F00160EEDD, 3505 <116>
1029 DATA 02ADD02C905F00160A9008DDC028D
DD0260ADDD02C9029005C903B00E60CE, 3377 <87>
1030 DATA 01D0CE03D0CE05D0CE07D060AD01D0
C9BC900160EE01D0EE03D0EE05D0EE07, 4166 <236>
1031 DATA D060ACD302015F035C028F031C03C
F02DC04BF029C053F025C063F021C07B, 4232 <244>
1032 DATA F01DC095F019C0A7F015C0BCF011C0
CBF00DC0D2F009C0E3F005C0FBF00160, 4875 <132>
1033 DATA A9018DD80260ACD302C02C016F035C029
F031C03DF02DC04CF029C054F025C064, 3977 <59>
1034 DATA F021C07CF01DC096F019C0A8F015C0
BDF011C0CCF00DC0D3F009C0E4F005C0, 4882 <26>
1035 DATA FCF00160A5A22907F00160EE002AD
DE02F006A9008DD80260A9018DDFC02AE, 3748 <88>
1036 DATA E002A9019DE00260ADDF02D00160EE
DF02ADDF02C919F021A740BD04D4A941, 3762 <102>
1037 DATA 8D04D4A91C8D02D48D03D4A9608D05
D4A98BD06D4A90F8D01D4A9008DDDF02, 3673 <191>
1038 DATA A940BD04D460ADDC02F00160ADDB02
D00160A5A2290FF0034CFE79A9018DD0C, 3517 <84>
1039 DATA D4A9008D0DD4A9B08D0BD4A9818D0B
D4A9508D08D4A5A2E905290FF00160A9, 3807 <40>
1040 DATA 018D0CD4A9008D0DD4A9808D0BD4A9
818D0BD4A9508D08D460A200A000EAEA, 3720 <217>
1041 DATA EAC8D0FAE8D0F5A210BEF8C3E8BEF9
C3E8BEFAC3E8BEFBC320937E20277FA9, 5469 <208>
1042 DATA 148DD202A90A8DD302A9008DEF02A0
0F79DF028BD0FAA90FBD18D4203075AD, 3535 <181>
1043 DATA D302C905F0034C647A78A2FBA9209D
FFBF9DF9C09DF3C19DFFD79DF9D89DF3, 5137 <219>
1044 DATA D99DECD9A9DECC2CAD0E5A9038D21D0
A9008DE102A200A9C085FBA92085FAA0, 4809 <199>
1045 DATA 0AB9EF0291FAB8D0F88A1869A2C9AB
B007A00F91FA4CC77A3BE90AA00F91FA, 4243 <111>
1046 DATA 8BA9A291FA18A5FA691485FAA90065
FB85FB8DE102F00DA006B93B0391FAB8, 4327 <221>
1047 DATA D0F84CF07AA009B9410391FAB8D0F8
18A5FA691485FAA90065FB85FB8E00E, 4625 <71>
1048 DATA D0A0A200A9A18D96C28D97C28DE102
D0034C307BAD97C21869048D97C2C9AB, 4214 <130>
1049 DATA B0034C307BEE96C238AD97C2E90A8D

```

```

97C2E8E00ED0D7A20BBD4A039D8BC2CA, 4332 <155>
1050 DATA D0F7AD00DC2910D0F9A2FBA90F9DF
D79DF9D89DF3D99DECDACAD0EFFFFFF02, 5564 <51>
1051 DATA 5BA2FBA9209DFBF9DF9C09DF3C19D
ECC2CAD0EFA9008DEB03A9FF8D15D0BD, 5215 <200>
1052 DATA 1CD0A9A8DAEC38DAFC3A9A38D9CC3
A9A58D9DC3A9A18D9EC38D9FC38D9BC3, 5056 <242>
1053 DATA A9A18DB7C38DB8C38DB9C38DBAC3AD
9BC3C9A0D0F9A900BD04D48D0BD48D12, 4802 <209>
1054 DATA D47BA204BDB6C39D6103CAD0F7A900
8D21D0A920A2F89DFBF9DF9C09DF3C1, 4937 <123>
1055 DATA 9DECC2CAD0F1AD00D0CDB0B01CEE
6303AD630318C9ABF0034C1D7CEE6203, 4066 <185>
1056 DATA AD6303E90ABD6C39D6103CAD0F7A900
630318C9A1F0034C1D7CCE6203AD6303, 3013 <11>
1057 DATA 18690ABD6303A209BD65039D9BC1CA
D0F7AD6203BDA3C1AD6303BDA4C1AD64, 3822 <225>
1058 DATA 03BDA6C1AD6303BDA7C1A2FBA9019D
FFD79DF9D89DF3D99DECDACAD0F1AD00, 5165 <92>
1059 DATA DC2910D0F94C217AAD5B03D00160EE
5C03AD5C03C901F018C902F026C903F0, 3529 <65>
1060 DATA 34C90A402C905F050A9008D5C034C
DF7CA2148EFC3E8BEFDC3E8BEFDC3E8, 4565 <141>
1061 DATA BEFFC360A2188EFC3E8BEFDC3E8BE
FEC3E8BEFFC360A21C8EFC3E8BEFDC3, 5629 <155>
1062 DATA EBFEFDC3E8BEFFC360A2208EFC3E8
BEFDC3E8BEFEC3E8BEFFC360A224BEFC, 5683 <17>
1063 DATA C3E8BEFDC3E8BEFEC3E8BEFFC360A2
28BEFDC3E8BEFDC3E8BEFEC3E8BEFFC3, 6020 <40>
1064 DATA 60AD5D03D00160A5A22903F006A900
8D5B0360A9018D5B0360AD00C2908F0, 2970 <60>
1065 DATA 08AD00DC2904F02560AD02D0C9FA90
0DC5E03CE5E03CE9FC3CE9FC3A0EE00, 3870 <232>
1066 DATA D0EE06D0EE04D0EE02D0CE5E0360AD
00D0C90AB00EE5E03A5A22907F00160, 3786 <36>
1067 DATA EE9FC360CE00D0CE02D0CE04D0CE06
D0EE5E03EE5E0360AD5E03C723B005C9, 4010 <101>
1068 DATA 19902760A91E8D5E03EEAFC3ADAFDC3
C9ABF00160A9A18DAFC3EEAEC3ADAEC3, 4591 <83>
1069 DATA C9ABF00160A9AABDAEC360A91E8D5E
03CEAFC3ADAFDC3C9A0F00160A9AABDAF, 4563 <61>
1070 DATA C3CEAEC3ADAEC3C9A0F00160A9A18D
AEC360ADAE3C9A6900160C9A390034C, 4683 <7>
1071 DATA D27DC9A1F00E4CD67DCE5F0360CE5F
03CE5F0360CE5F03CE5F03CE5F03CE5F, 3683 <120>
1072 DATA 0360AD5F03C903B00160A9008D5F03
AD00D0C90AB00160CE00D0CE02D0CE04, 3160 <33>
1073 DATA D0CE06D060A5A22903F00160AD00D0
C90AB00160CE00D0CE02D0CE04D0CE06, 3757 <66>
1074 DATA D060A204CE9FC3BD9BC3C9A190034C
427EDE9AC3BD9BC369A9D99BC3CAD0E8, 4720 <213>
1075 DATA 60A204EEBAC3EEBAC3BDB6C3C9ABBO
034C637EFEB5C3BDB6C3E90A9DB6C3CA, 5093 <58>
1076 DATA D0E860AD9CC3C9A1F00160AD9DC3C9
A4900160A5A22902F00160AD00D0C90A, 4189 <8>
1077 DATA B00160CE00D0CE02D0CE04D0CE06D0
60AD00DD29FC8D00DDA9088D18D08D25, 3814 <251>
1078 DATA D0A9C08DB802A9098D27D08D28D08D
29D08D2AD08D22D08D2BD08D2CD08D2D, 3938 <172>
1079 DATA D08D2ED0A9038D23D0A9008D26D0A9
0F8D15D08D1CD0A9D28D16D0A9738D08, 3829 <224>
1080 DATA D08D0C0A98B8D0AD08D0E0D0A9A8D
09D08D0E0A9BF8D0DD08D0FD0A9648D, 4152 <131>
1081 DATA 00D08D04D0A97C8D02D08D06D0A9BC
8D01D08D03D0A9D18D05D08D07D0A900, 3876 <187>
1082 DATA 8D20D0A9058D21D060A2F8BD47719D
FFBFBD41729DF9C0BD3B739DF9DFD79DF9D89DF3D9
9DECDACAD0D7A90F8D18D4A9008DE903, 5096 <30>
1084 DATA 78A9F08D1AD0A9008D0DDCA9A48D14
03A97F8D150358AD00C2910D0F9A2FB, 3813 <149>
1085 DATA A9209DFBF9DF9DF9C09DF3C19DECC2CA
D0EF78A9318D1403A9E8D150358A900, 4558 <75>
1086 DATA 8D04D48D0BD48D12D460CEE803ADE8

```

```

03F0034C31EAA90A8DEB03A9008D18D4, 3740. <89>
1087 DATA A9A8D05D48D0CD4ED13D4A9AF8D06
D48D0DD48D14D4A9C88D02D48D09D48D, 4007 <4>
1088 DATA 10D4A2028E03D4E8BE0AD4E8BE11D4
A9408D04D48D0BD48D12D4A9418D04D4, 3864 <207>
1089 DATA 8D0BD48D12D4AEE903E080F0034C0C
B0A2008EE903BD3C6E8D01D4BDBE6EBD, 3839 <36>
1090 DATA 00D4BD406F8D08D4BDC26F8D07D48D
44708D0FD4BDC6708D0ED4EEE9034C31, 3993 <74>
1091 DATA EA60CEEA03ADEA03F00160A9068DEA
03A9468D05D48D0CD48D13D4A9A8D06, 3894 <31>
1092 DATA D48D0DD48D14D4A9A08D02D48D09D4
8D10D4A2028E03D4E8BE0AD4E8BE11D4, 4086 <99>
1093 DATA A9408D04D48D0BD48D12D4A9418D04
D48D0BD48D12D4AEEB03E05FF0034C98, 3853 <159>
1094 DATA 80A2008EEB03E05BD026C8D01D4BD616C
8D00D4BDC06C8D0BD48D1F6D8D07D48D, 3798 <91>
1095 DATA 7E6D9D0FD4BDD6D8D0ED4EEEB0360
B481, 2370 <150>
ENDE DES LISTINGS

```

## Listing 2

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

1 AD=34176;FORI=1000TD1099;READA#,PS:C=0 <243>
:FORN=1TOLEN(A#)STEP2
2 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A#,N <161>
+1,1))-48:A=A+7*(A>9):B=B+7*(B>9)
3 P=PEEK(AD),16*A+B:C=C+P:AD=AD+1 <176>
4 NEXT:PRINTI;"(SPACE)";:IFC<>PSTHENPRIN <157>
T"ERROR":STOP
5 PRINT"OK":NEXT <143>
6 PRINT"FERTIG":END <136>
1000 DATA 000000000000000000000000000000 <2>
00000000000000000000000000000000F00, 18
1001 DATA 000F0000D00000300000300000300 <90>
000300000300000300000300000300000300
1002 DATA A90000000000000000000000000000 <36>
00000000000000000000000000000000F000, 409
1003 DATA 004000004000800000800002A0FF0A <70>
9BF00AABC06AAAC09A2A6AA6066AA950, 3075
1004 DATA 5A8B00000AAAAA2B2AA202AAA000A <95>
AA000AA0006AA0006B00006B0000600, 1914
1005 DATA 00060000600000600000600000600 <253>
000A0000600000600000000000000000, 52
1006 DATA 00000000005A8B009A8B00BA8B00 <9>
BA8B00BA8B00AA0002A00000A00000, 2290
1007 DATA 600000600000600000600000600000 <19>
600000A0000020000020000000000000, 800
1008 DATA 000000000000000000000000000000 <118>
000000000000000000000000000000, 0
1009 DATA 000000000000000000000000000000 <10>
000000000000000000000000000002A, 42
1010 DATA 0802AA0AAAA0AAAA000000000000 <138>
000000000000000000000000000000, 870
1011 DATA 00000000000300000FC0200D00A0 <130>
0502A03C0A9B3F2AABFFFAAF09A0AFA, 2390
1012 DATA A556DAAB0056AB0096AB0000000AAA <97>
000AAA000AA002AA0002AA0002A000, 2255
1013 DATA 0AA0000A0002BA0002BA0000A2800 <86>
02BA0000B20000B20000B20000000000, 1320
1014 DATA 000000000000000000000000000000 <201>
AB009A8B00BA000BA000BA0000AB, 1866
1015 DATA 000AB00000A00008A0000B200028A <100>
000A2B0000A0000000000000000000, 956
1016 DATA 000000000000000000000000000000 <126>
0000000000000000000000000000, 0
1017 DATA 000000000000000000000000000000

```

00000000000000000000000000000000, 0	<127>	02800002A00000AA00000A0000020000, 1240	<202>
1018 DATA 000000000000002A0002AA0AAAAA2A		1055 DATA 02000000000000000000000000000000	
AAAA000000000000000000000000000000, 946	<227>	00000000000000000000A6AB009AAB00BAA0, 1004	<93>
1017 DATA 000000000000000000000000000000003		1056 DATA 00BAA000AE000000000000000000000000	
00000FC0200D00A00502AB3F0A983FEA, 1112	<75>	0002BB000A0B000AA2B000AAA00008B000, 2148	<223>
1020 DATA AAF06AAAF09A06FAA552DAAB0056AB		1057 DATA 80000000000000000000000000000000	
0076AB00000000AAA000AAA000AAA0002, 30B3	<82>	0000000000000000000000000000000000, 128	<133>
1021 DATA AA0002AA0002A0000AA0000AB0002A		1058 DATA 000000000000000000000000000000000	
80002BB0002BA00000BA00000000000000, 1598	<190>	000000000000000000000000000000000, 0	<169>
1022 DATA 000B2B000A0B00000000000000000000		1059 DATA 000000000000000000000000000000000	
0000000000000000A6AB009AAB00BAA000, 106B	<56>	00AA200AAA2AAAA02AAAA0000000000, 1106	<159>
1023 DATA BAAB00BEAB0002AA0000AAB000A2B0		1060 DATA 000000000000000000000000000000000	
002BB0002BB0000A0000000000000000, 2214	<64>	000000000000000000000000000000000, 0	<222>
1024 DATA 028000008000000000000000000000000		1061 DATA A00F02A03F0A9B3FEAABF0EAAAF0BA	
000000000000000000000000000000000, 258	<209>	2AFAA5565AAB0056AB0096AB0000000A, 3480	<163>
1025 DATA 000000000000000000000000000000000		1062 DATA AA000AAA000AAA000AAA0002AA0002	
00000000000000000000000000000000, 0	<136>	A000290000A40000A40002900002900, 1420	<32>
1026 DATA 000000000000000000000000000000002		1063 DATA 000900000A00000280000280000080	
AA000AAA00AAAA0AAAA000000000000, 1042	<2>	000000000000000000000000000000000, 407	<239>
1027 DATA 000000000000000000000000000000000		1064 DATA A6AB009AAB00BAA000BAA000AE000	
000000000300000FC0000D0000200500A, 420	<240>	00AB0000AB0000AA0000A00002A2B0020, 2450	<214>
1028 DATA 0C02A03F0A9BFFFAABF0EAAAF09A2A		1065 DATA AB000AB0000A000000000000000000000	
FAA4065AA95056AB0096AB0000280AAA, 3687	<14>	000000000000000000000000000000000, 316	<5>
1029 DATA 200AAA000AAA0002AA0002AA0002AA0		1066 DATA 000000000000000000000000000000000	
000AA0002AB0002AB000A20000A20000, 1732	<173>	000000000000000000000000000000000, 0	<177>
1030 DATA 820000820000A2B00020B0000000000		1067 DATA 000000000000000000000000000000000	
0000000000000000000000000000000A6, 876	<73>	000000000000000000000000200002A0000AA, 214	<94>
1031 DATA AB009AAB00BAA0000BAA0000AE00000		1068 DATA 0002AA000000000000000000000000000	
AB0000AB00002A00002A00002A00000A, 1988	<7>	000000000000000000000000000000000, 172	<31>
1032 DATA 00000A00002AB00022800020B00000		1069 DATA 0000200030A000FEA000DA9B035AAB	
80000000000000000000000000000000, 630	<145>	0FEAAA3FFA2A3EAD565AAB005AAB0096, 3046	<0>
1033 DATA 000000000000000000000000000000000		1070 DATA AB00A6AB000002AAA00AAA02B2AA	
00000000000000000000000000000000, 0	<144>	0202AA0202AA0202AB0002A000029000, 2184	<228>
1034 DATA 000000000000000000000000000000000		1071 DATA 0A40000A40000A0000090000280000	
0000000000000000AA000AAA02AAAA2A, 734	<56>	240002000002000002000000000000000, 339	<34>
1035 DATA AAAA0000000000000000000000000000000		1072 DATA 000000000000000000000000000000000	
00000000000000000000000000000000F, 358	<170>	AAA0BE06A0B001A00001B0000680001A, 2128	<98>
1036 DATA C0000D00B00500803C02A03F0A9BFF		1073 DATA 00001B000000000000000000000000000	
CAABF06AAAF09A2AFAA606DAA95056AB, 3633	<144>	000000000000000000000000000000000, 24	<224>
1037 DATA 0076AB00002B0AAA000AAA000AAA000		1074 DATA 000000000000000000000000000000000	
0AAA002AAA002AB002AA0002AA0002AB, 1974	<167>	00000000000000000000000000000000, 0	<185>
1038 DATA 0002B0000000000000000A00000A00000800		1075 DATA 000000000000000000000000000000000	
000B000000000000000000000000000000, 168	<98>	000000000000000000000000000000002, 2	<186>
1039 DATA 0000000000000000A6AB009AAB00BAA0		1076 DATA 80AA0AAAAA0B2AAA0B02AA000000000	
00BEAB00AAAB0002AA0002BA00020A00, 2030	<190>	000000000000000000000000000000000, 1048	<241>
1040 DATA 020280000000000000000A0000000002028002B		1077 DATA 000000000000000000000000000000000	
20000820000A20000200000000000000, 674	<43>	00200030A000FEA00FD69B3FDAABFFFF, 1994	<118>
1041 DATA 000000000000000000000000000000000		1078 DATA 6A7EAA6A5AAB45AAB1096AB000008	
00000000000000000000000000000000, 0	<152>	02AA0002AA0002AA0002AA000AAA040A, 2318	<73>
1042 DATA 000000000000000000000000000000000		1079 DATA A0096A400AA9000AA40000000000000000	
000000000000000000000000000000000, 0	<153>	000000000000000000000000000000000, 692	<177>
1043 DATA AA000AAA2AAAAA20AAAA0000000000000		1080 DATA 000000000000000000000000000000000	
000000000000000000000000000000000, 1104	<125>	0076AB009AAB00BAA00BAA000AEAA0, 1822	<200>
1044 DATA 000000000300000FC0000D0000000000500		1081 DATA 0006A00001A00001A000064B0002A00	
803C02803F0A0B0FFEA60F0EAA0F09AAB, 2528	<148>	000000000000000000000000000000000, 760	<148>
1045 DATA FAA4AB5AA95B56AB0096AB00000000A		1082 DATA 000000000000000000000000000000000	
AA000AAA000AAA0002AA0002AA0002A0, 2547	<179>	000000000000000000000000000000000, 0	<193>
1046 DATA 0002A0000AA0000A20000A20000A200002AB		1083 DATA 000000000000000000000000000000000	
0000BB0000AA00002200000000000000, 926	<190>	00000000000000000000000000000002A0, 162	<157>
1047 DATA 000000000000000000000000000000000		1084 DATA 000B2B00000AA0000AAA002AAA902	
A6AB009AAB00BAA000BEA000AAA0002, 1684	<131>	AA1A76AA000000000000000000000000, 1417	<95>
1048 DATA A00002A0000280000AB0000A000002A		1085 DATA 000000000000000000000000000000000	
0000AA00008B00000B00000800000000, 964	<74>	000000000000000000000000000000000, 502	<243>
1049 DATA 000000000000000000000000000000000		1086 DATA 0B40000F00203FC0A0FFF2A05EB69B	
00000000000000000000000000000000, 0	<160>	9AA6AB96A9AA0002AAAA002AAA000AAA, 3335	<17>
1050 DATA 000000000000000000000000000000000		1087 DATA 0000AA000002000000000000000000000	
00000000000000000000000000000000, 0	<161>	000000000000000000000000000000000, 172	<66>
1051 DATA 00000000000000AA2B0AAA2AAAAA		1088 DATA 000000000000000000000000000000000	
02AAAA00000000000000000000000000, 1114	<186>	000000000000000000000000000000000, 1004	<2>
1052 DATA 000000000000000000000000000000000		1089 DATA AB00EAA000BAA00002AAB0002AB0000	
0FC0000D00200500A03C02A03F0A9BFF, 1122	<156>	2B00000A000006000002400001B00000, 1539	<221>
1053 DATA EAABF06AAAF09A2AFAA5565AAB0056		1090 DATA 80000A00000600000200000200000	
AB0096AB0000000AAA000AAA000AAA00, 3225	<139>	000000000000000000000000000000000, 448	<37>
1054 DATA 02AA0002AA0002A000028000028000		1091 DATA 000000000000000000000000000000000	

```

00000000000000000000000000000000, 0 <202>
1092 DATA 00000000000000000000000000000000 <110>
00002A00A2AA0AAAAA0A0AAA00000000, 914 <110>
1093 DATA 00000000000000000000000000000000 <169>
00000000000000000000000000000000C00003, 195 <169>
1094 DATA F0000340000D40200F00A03FC2A0FF <173>
FA7BFC2AAB5AA6AA9A7A9AA6AA560000, 3304 <173>
1095 DATA 02AA0002AA00002AA0000AA0000AA000 <165>
002B00002B00002700000A0000090000, 996 <165>
1096 DATA 0900002B00002400002100000000000 <98>
00000000000000000000000000000000, 118 <98>
1097 DATA 009A8009A8000EA0000FAB000BAB0 <27>
0002B00002A00000A00000A00000A000, 2502 <27>
1098 DATA 002B00002B00002B00002B00000B00 <138>
000B0000000000000000000000000000, 176 <138>
1099 DATA 00000000, 0 <218>
ENDE DES LISTINGS

```

```

1019 DATA 800000802BA00000000000000000000 <84>
00000C37F5FDF7F5F5F5F5D0C0705CD7, 3080 <84>
1020 DATA F5DDDD7D700000000C0705CD7DFDDDD <56>
DDFDDDD7D7D7777F777575F5755D5F, 4876 <56>
1021 DATA 5D5DDDFDDDC0705CD7757D7775D7D7 <171>
F7FFF7F5350D755D5F5D5DDDFDDDD7D7, 5076 <171>
1022 DATA D7D7777F777575F5755D5F5D5DDDF7 <80>
DDDDDDDDDD7D7DC0C0C0C0C0C0C0C0C0C0C, 5137 <80>
1023 DATA 20000000020000D7D7372F03000002 <93>
75F5755D5FDD3D0DFDDDD7D7D7D7777F, 3113 <93>
1024 DATA 7D7D7DFD7D5D5D5DC0C0C0C0C0C0C8C0 <237>
C077F5350D030000005DDDFDDDDDDDD3F, 4302 <237>
1025 DATA 0CC2C8CAC0C2C80000802A02880288 <118>
2000000F0F0F0A2AAABA000000C0C480, 2593 <118>
1026 DATA A0208A3505111111112220C0010000 <106>
100000003C3D3C2BAAAAAAD7141414, 1906 <106>
1027 DATA 1414280055FF7FDFF7FD550055FDF7 <174>
DF7FFF55000010000000400100004000, 2775 <174>
1028 DATA 100000040000000440000400010515 <123>
14505051504050541405030547FFF7FF, 986 <123>
1029 DATA 5D5D5D5D5D05050545050505050000 <221>
3C3C3F2BAAAABE55FF7FDFF7FD550000, 2645 <221>
1030 DATA 000000100000000004000000000000 <78>
000000040000000004000000001000, 164 <78>
1031 DATA 0000000400001000000000000000010 <182>
0114055DDFFFFF10451113F7F7FFF7, 2260 <182>
1032 DATA 004504477FFFFF0410005DDFFFFF <166>
FF01004477FFF7FFF003C0547FFF7FF, 3990 <166>
1033 DATA FFFFFFFFDF3F4100FFFF7FFFFFF5D <235>
0100FFFFFFFFFD10004FFFFFFFFFFFF4, 5874 <235>
1034 DATA 4001FFFF7FFFFFFC5000FFDFFFFFFF <71>
4D4400003F3333333333F0003030303, 3325 <71>
1035 DATA 030303003F030303F030303F003F0303 <238>
3F03033F003333333F030303003F0303, 885 <238>
1036 DATA 3F03033F003F03033F33333F003F03 <174>
030D03030003F33333F3333F003F33, 1215 <174>
1037 DATA 333F03033F00000000000000000000 <41>
00000000F0F000000000000030300D0, 1653 <41>
1038 DATA D0C0C0FFFFFF03030333033F3F30000 <72>
00F333F333F3003030F333333F30000, 3624 <72>
1039 DATA 003F303F033F0030303F3333333300 <15>
FCFCFCFC3C3C3C0000000000C0C0F0F0, 3159 <15>
1040 DATA 00000003030300030000000F303F333 <50>
F3000000F3303030F0F0C0D0F3C0C0C0, 2865 <50>
1041 DATA D0C0C0C0F0C0C0C0F00000A0A02B2B <59>
0A0A000002020A0A2B8B0000B0B00000, 2732 <59>
1042 DATA 00000A0A2A2B2BA0A0A0AB82A0A0A <114>
0202020A0B0B0B8B8B8B2A2222222, 1860 <114>
1043 DATA 22A2B220A2B220002000A0A0B0A0 <120>
AB0BABA00B2AA2AAB0A22A0B0000B0B0, 2634 <120>
1044 DATA 0B0000000020A0A2B20A0B000A0AB <147>
0A0202000B0B0B0A020280A0200AA2A, 1692 <147>
1045 DATA 02020AAA0A2A200A2A200A0A2A2 <247>
A2A2A2A2A2222AAAAA222222222020A, 2264 <247>
1046 DATA 2B2A20280A020082A2A200A2B200, <152>
912
ENDE DES LISTINGS

```

## Listing Teil 3

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

1 AD=37560:FORI=1000T01046:READA#,FS:C=0 <64>
:FORN=1TOLEN(A#)STEP2 <64>
2 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A#,N <161>
+1,1))-48:A=A+7*(A>9):B=B+7*(B>9) <161>
3 POKEAD,16*A+B:C=C+16*A+B:AD=AD+1 <42>
4 NEXT:PRINTI;"(SPACE)":;IFC<>PSTHENPRIN <157>
T"ERROR":STOP <157>
5 PRINT"OK":NEXT <143>
6 PRINT"FERTIG":END <136>
1000 DATA 0001053D3D3D3F3F00404040404040 <205>
D0F47D4F43404040D0000145CDFD3D3F, 2522 <205>
1001 DATA 3F3D3D3D3D3D3D3D00000000000000 <26>
00003D3D7DFDFD3D3F3FF3FF3FF3FF, 3021 <26>
1002 DATA FFFF01057DFDFD3D3D3D4040404040 <84>
404040FFFFFFFFFFF3FF3FF3FF3FF, 5157 <84>
1003 DATA FFFFFFF3D3D7DFDFD3D3D3D3D3D3D3D <194>
3D3D3F3F47D4F43404040403D2C08B0, 3205 <194>
1004 DATA 200280004040008B200208803D3D3D <2>
3D3D3D8D2D0008B0000000000812800, 1501 <2>
1005 DATA 200020000040B0008B220002803D3D <186>
3D3D3D8D2D0920822000002200008020, 1492 <186>
1006 DATA 00A00800000001053D3D3D3D3D00 <76>
000000000000B000000000000020000, 670 <76>
1007 DATA 02082B802B225B80C8A2088A20000 <249>
808BA20200000200000000020000800, 1516 <249>
1008 DATA 0800022008B22008B2080802008 <243>
0002002000000020B020020000B00002, 1044 <243>
1009 DATA 00002000200002000000B80B000008 <112>
002000010FB2F8F0F0F4040404040, 1013 <112>
1010 DATA 40A000020020000000000000000000 <135>
2002002002000000B200020F2208222A, 719 <135>
1011 DATA 208BA88820A208022008A00000B080 <51>
2800820820000200000000002002220A, 1422 <51>
1012 DATA 20082228B22008B000082080020022 <103>
B00B00B000B00B00A000000000A000B0, 1464 <103>
1013 DATA 0000000005055A5A5A5505001455 <175>
69AA9655410005555AAAA55550004155, 2190 <175>
1014 DATA 96AA695514000303030303030314 <0>
54F4FDFFF7F4F40000000040D0F43D00, 2722 <0>
1015 DATA 00000000003435F4F4FDFFF7F4F4F4 <154>
0F03004092F43D0F40D0F43D0F030040, 3287 <154>
1016 DATA 000000040D0F43D3430303832328B <29>
00F6FDFFF7F6F0000003A040DBF43D0F, 3267 <29>
1017 DATA 03D2F43D0F230040D00F0B0040D0F4 <224>
3DBF55D5FD3D3D3D7DF43D0F030000, 2829 <224>
1018 DATA 000030040D0F43D0F03FD3D3D3D <32>
3D7DFD00000008200A8020FD3D3D3DBC, 2562 <32>

```

## Aufrufprogramm

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

0 POKES6,100:PRINTCHR$(8) <28>
1 IFA=0THENA=1:PRINT"(CLEAR)PLEASE(SPACE <34>
)WAIT(SPACE)2(SPACE)MINUTES":LOAD"HORSE(<34>
SPACE)DATA",8,1 <34>
5 FORX=832T0865:READQ:POKEX,0:NEXT <19>
6 DATA120,169,51,133,1,169,0,133,95,133, <30>
90,133,88,169,208,133,96,169,240,133 <30>
7 DATA89,169,224,133,91,32,191,163,169,5 <69>
5,133,1,88,96 <69>

```



```

8 SYS832:POKEB50,160 <206>
10 FORI=699T0718:READQ:POKEI,Q:NEXT <51>
15 DATA3,3,6,1, <176>
20 DATA040,0,146,056, <88>
30 DATA0,5,5,14, <187>
40 DATA0,0,0,0, <139>
50 FORI=752T0761:READQ:POKEI,Q:NEXT:DATA <163>
8,9,14,4,5,18,14,9,19,19 <62>
60 FORI=828T0853:READQ:POKEI,Q:NEXT <151>
61 DATA6,5,8,12,5,18,7,5,19,3,8,1,6,6,20 <117>
70 FORI=870T0878:READQ:POKEI,Q:NEXT <12>
71 DATA9,8,18,5,32,26,5,9,20 <201>
95 FORI=34176T037247:POKEI+16000,PEEK(I) <249>
:NEXT <78>
96 FORI=37560T039500:POKEI+20000,PEEK(I) <146>
:NEXT <14>
97 FORT=57552T057559:READQ:POKET,Q:NEXT <15>
98 DATA126,14,28,56,112,112,126,0 <252>
99 FORT=57600T057607:POKET,Q:NEXT <185>
100 SYS31265 * <143>
180 ***** * <187>
181 * * <212>
182 * HORSE GAMES * <189>
183 * * <6>
184 * (C) 1986 BY * <191>
185 * * <120>
186 * RADDISH-SOFT * <193>
187 * * <194>
188 * MUSIC (C) BY SKINNY * <50>
189 * * <196>
190 * * <11>
191 * TEL. 02365 / 59008
192 *
195 *****
ENDE DES LISTINGS

```

```

20202034202020203420203536373839, 1207 <129>
1010 DATA 343420343420202034202034202020 <228>
20342020203420202020202034203434, 1212
1011 DATA 203420343420343434676869343420 <84>
20202034203420342020203434202020, 1468
1012 DATA 202034202020342053542020342020 <86>
342020202020205354203420202020, 1298
1013 DATA 20342020342020202020203420202020 <84>
34202034202020343637383934342034, 1266
1014 DATA 342020203420203420202020342020 <206>
2034202020202020202020342020202034, 1132
1015 DATA 20202020202020202020203420281D1E1C <83>
20203420203420202020203420202020, 1091
1016 DATA 203420202020342020202034202020 <10>
20202020202034202034203420202020, 1112
1017 DATA 1F29211D232420202020203434202020 <7>
3420203434202020202020202020342020, 1125
1018 DATA 2020203A3B3C3D3E3F404120203434 <35>
20203420203420203420203420202020, 1348
1019 DATA 202020342020202020203420202020 <66>
20556A6B6C6D6E6F3420342020343420, 1560
1020 DATA 34342020202034202020202020342020 <15>
55565758592020342020342020342055, 1460
1021 DATA 565758593420343420203420202020 <53>
3420342020343420202034202034202034203A, 1440
1022 DATA 3B3C3D3E3F40412020343420203420 <200>
20342020342020342020202020202034, 1342
1023 DATA 3420203420203420203420203420342034 <191>
202020201F29211D232420202020203434, 1165
1024 DATA 202020342020343420202020202020 <195>
2034202020202020342020202034202020, 1112
1025 DATA 203420202034342020A2B2C29272434 <161>
203434202034342034202020203420, 1265
1026 DATA 202020203220203420342020204243 <187>
20444546473420203420342034202020, 1349
1027 DATA 20202020202020342020203420203420 <191>
3420342020343432325A6E6F70717273, 1709
1028 DATA 343420342020202034203420342020 <6>
20202020202034205A5B5C5D5E5F2020, 1495
1029 DATA 342020343420205A5B5C5D5E5F2034 <181>
20202034202034203420343420203420, 1555
1030 DATA 202034202034202020424320444546 <10>
47342020342034203420202020202020, 1331
1031 DATA 202020342020203420342020202020 <123>
3420202020342020202020342020203420203420, 1174
1032 DATA 27243420343420202034342034342020 <235>
202034202020202020342020202020342020, 1203
1033 DATA 20202034202034203420342034203420 <203>
20342D2E2F3024202034203434202020, 1254
1034 DATA 203420342020342020202034203420 <184>
202020203120202048494A4B4C4D3434, 1404
1035 DATA 202020202020342020202034202020 <104>
34203420202020342034203420203434, 1172
1036 DATA 313274757677787932342034202034 <244>
34342020203420342020202020203420, 1724
1037 DATA 203132606162202020342034203420 <17>
3432326061622020203420203420342034, 1593
1038 DATA 202034202034202034202020342034 <207>
3220202048494A4B4C4D343420202020, 1405
1039 DATA 20203420202020342020202034203420 <172>
2020203420342034203420342034203434, 1232
1040 DATA 20342020342D2E2F30242020342034 <20>
34202020203420342020342020202020, 1214
1041 DATA 20203420203420342020202020342020 <88>
3420202034202034203420342D313234, 1240
1042 DATA 203420322020342020202020203420 <88>
20202020342020203420203220203420, 1148
1043 DATA 20342020342020202034202034202020 <176>
20342020202020342020203420202020, 1132
1044 DATA 202020342020202034202020747A7B <186>
7C7D2034202020202020202020342020, 1522
1045 DATA 202034203420203420203220636465 <243>
20203420202020202034312063646520, 1535
1046 DATA 2020342020202034203420342020342020

```

## Listing Teil 4

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

1 AD=24576:FORI=1000T01117:READA#,FS:C=0 <80>
:FORN=1TOLEN(A#)STEP2
2 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A#,N <161>
+1,1))-48:A=A+7*(A>9):B=B+7*(B>9)
3 POKEAD,16*A+B:C=C+PEEK(AD):AD=AD+1 <176>
4 NEXT:PRINTI;"(SPACE)";:IFC<>PSTHENPRIN <157>
T"ERROR":STOP
5 PRINT"OK":NEXT <143>
6 PRINT"FERTIG":END <136>
1000 DATA 202020202020202020202020202020 <233>
1B1C2020202020202020202020202020, 983
1001 DATA 202020202020202020202020202020 <202>
20202020203334203420203420202020, 1071
1002 DATA 20202020203420202020202020203420 <54>
202020203420202020202020342020, 1072
1003 DATA 206620342034203420202020342034 <226>
20202020203434202020342020343420, 1262
1004 DATA 20342020202034202020203420202020 <110>
2034202020203420202020202034, 1132
1005 DATA 342020342020342020342020203420 <244>
203334203420203420202020202020, 1171
1006 DATA 20342020202020202020203420202020 <246>
2020202020202020202020202020, 1032
1007 DATA 2020201B1C20202020202020202020 <168>
20202020202020202020202020202020, 983
1008 DATA 20202020202020202020202034202020 <218>
2020342020203420281D1E1C20203420, 1071
1009 DATA 20342020202020342020202020203420

```

# programme

2020342020203420202034202020342020, 1152 <145> 93948F90919293948F90919293948F90, 4509 <168>  
1047 DATA 34202020342020203420202020342020 <126> 1084 DATA 919293948F90919293948F90919293 <145>  
20202034202020342020203420202034, 1152 <126> 948F90919293948F90919293948F9091, 4510 <145>  
1048 DATA 2020203420202034202020342020342034 <188> 1085 DATA 9293948F90919293948F9091929394 <27>  
2D313234203420322020342020202020, 1218 <188> 8F90919293948F90919293948F909192, 4511 <27>  
1049 DATA 20342020202020202034202020342020 <49> 1086 DATA 93948F90919293948F90919293948F <225>  
20208688878786888786888286878782, 2483 <49> 90919293948F90919293948F90919293, 4512 <225>  
1050 DATA 87888788888288828887868D86898A <145> 1087 DATA 948F90919293948F90919293948F90 <227>  
868788868D86828786888786868D8687, 4191 <145> 919293948F90919293948F90919293948F90, 4513 <227>  
1051 DATA 87878D868788828688868786888882 <62> 1088 DATA 8F90919293948F90919293948F9091 <223>  
8782868D868286888788868D86878882, 4176 <62> 9293948F90919293948F90919293948F, 4508 <223>  
1052 DATA 868688868786868688888688878888 <227> 1089 DATA 90919293948F90919293948F909192 <174>  
86888887878786878688878787878886, 4184 <227> 93948F90919293948F90919293948F90, 4509 <174>  
1053 DATA 8688868786887888887868788878782 <87> 1090 DATA 919293948F90919293948F90919293 <46>  
888D868D8886898A86878782888D8682, 4192 <87> 948F90919293948F90919293948F90919293, 5058 <46>  
1054 DATA 88828682868D8882868D8888878782 <196> 1091 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF <17>  
878886888687888882888687868D8886, 4186 <196> FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF <17>  
1055 DATA 8788888D88868786878286878D8882 <201> 1092 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF <202>  
87868782888687878787872043406363, 3868 <201> FFFFFFFFFFEFEFOCFFFF88FFFF004043, 6877 <202>  
1056 DATA 8682868D8882868D88888787888788 <162> 1093 DATA 0000006343000043003C0000430000 <150>  
86888687888882888687868D88868788, 4191 <162> 00000000000000000000000000000000, 360 <150>  
1057 DATA 888D8886888386878786858E8584858E <238> 1094 DATA 000000006300000000630000000000 <73>  
87828886878787878720888687878786, 4077 <238> 000000000040004000000000FFFFBFBFFF, 1473 <73>  
1058 DATA 888687838686878386848584858E85 <36> 1095 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF <21>  
8E8584858E858B8C848584858E858E86, 4184 <36> FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, 7905 <21>  
1059 DATA 84868886888386878784858E8584858E <19> 1096 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF <244>  
858485848584858E858E858E858E8584, 4181 <19> FFFFFFFFFFFFFFFFFEFOCFFFF88FF, 7511 <244>  
1060 DATA 8584858E8584858E85848584858485 <13> 1097 DATA FF004043000000436300004300BE00 <246>  
848584858485848584858485848584858485, 4128 <13> 00430000000000000000000000000000, 876 <246>  
1061 DATA 848584858485848584858485848584 <19> 1098 DATA 000000000000000000000000006300 <91>  
858485848584858E858E858E858485848584, 4151 <19> 00000000000000000000400040000000FF, 581 <91>  
1062 DATA 8584858E858E858E858E858E858E85 <88> 1099 DATA FFBFBFFFFF0D060D060D060D050B05 <133>  
8E858E8788878888878888687878788, 4225 <88> 0B050B050B050A050A050A050A040904, 1358 <133>  
1063 DATA 838888888683888687868883888687 <216> 1100 DATA 09040904090409060D060D060D050B <17>  
86888388868387838886878388868787, 4159 <216> 050B050B050B050A050A050A050A0409, 248 <17>  
1064 DATA 888686204140636358E858E858E85 <169> 1101 DATA 040904090409060D060D060D05 <30>  
8E878887888887888868787888388, 3890 <169> 0B050B050B050A050A050A050A04, 243 <30>  
1065 DATA 888886838886878688838886878688 <162> 1102 DATA 090409040904090A050A050A050ACF <112>  
83888683878388868783888687878886, 4162 <162> 9ECF9ECF9ECF9E7BF77BF77BF77BF7E2, 3653 <112>  
1066 DATA 862095969798999A95969798999A95 <75> 1103 DATA C4E2C4E2C4E2C4E2C4850A850A850A <175>  
969798999A95969798999A9596979899, 4557 <75> CF9ECF9ECF9ECF9E7BF77BF77BF77BF, 4974 <175>  
1067 DATA 9A95969798999A95969798999A9596 <170> 1104 DATA E2C4E2C4E2C4E2C4E2C4850A850A85 <233>  
9798999A95969798999A95969798999A, 4699 <170> 0ACF9ECF9ECF9ECF9E7BF77BF77BF77B, 4953 <233>  
1068 DATA 95969798999A95969798999A959697 <87> 1105 DATA F7E2C4E2C4E2C4E2C4D201A22201D <225>  
98999A95969798999A95969798999A95, 4694 <87> 1A1A1D201A2722201D1A1A1A1A1A1D, 2575 <225>  
1069 DATA 969798999A95969798999A95969798 <255> 1106 DATA 201D1D1D1D1D1A18132727272727 <12>  
999A95969798999A95969798999A9596, 4695 <255> 272727272727272727272727272727, 1104 <12>  
1070 DATA 9798999A95969798999A9596979899 <227> 1107 DATA 2727272727272727272727271A181A1D20 <35>  
9A95969798999A95969798999A959697, 4696 <227> 1D1A201D1A171A1D1D1A1715151A1A1A, 945 <35>  
1071 DATA 98999A95969798999A95969798999A <3> 1108 DATA 1A1A1A13131313131313131345DB14CF <80>  
95969798999A95969798999A95969798, 4697 <3> DB45141445DB1412CFDB4514141414, 1994 <80>  
1072 DATA 999A95969798999A95969798999A95 <95> 1109 DATA 1445DB454545454545149D8912121212 <200>  
969798999A95969798999A9596979899, 4698 <95> 121212121212121212121212121212, 1327 <200>  
1073 DATA 9A95969798999A95969798999A9596 <176> 1110 DATA 12121212121212121212121212149D14 <129>  
9798999A95969798999A95969798999A, 4699 <176> 45DB4514DB45143B144545143BEDED14, 1888 <129>  
1074 DATA 95969798999A95969798999A959697 <132> 1111 DATA 141414141489898989898989898989 <14>  
A09B9C9D9E9FA09B9C9D9E9FA09B9C9D, 4803 <132> 00000000000000000000000000000000, 1196 <14>  
1075 DATA 9E9FA09B9C9D9E9FA09B9C9D9E9FA0 <177> 1112 DATA 0000000000000000000000000000001A1D <211>  
9B9C9D9E9FA09B9C9D9E9FA09B9C9D9E, 4883 <177> 201A22201D1A1A1D201A2722201D1A1A, 533 <211>  
1076 DATA 9FA09B9C9D9E9FA09B9C9D9E9FA09B <233> 1113 DATA 1A1A1A1A1D201D1D1D1D1A181320 <144>  
9C9D9E9FA09B9C9D9E9FA09B9C9D9E9F, 4884 <233> 1D20222722202722201D20222201D1A, 932 <144>  
1077 DATA A09B9C9D9E9FA09B9C9D9E9FA09B9C <164> 1114 DATA 1A2222222222221D1A181A1D22201D <221>  
9D9E9FA09B9C9D9E9FA09B9C9D9E9FA0, 4885 <164> 00000000000000000000000000000000, 459 <221>  
1078 DATA 9B9C9D9E9FA09B9C9D9E9FA09B9C9D <119> 1115 DATA 000000000000000000000000000000 <163>  
9E9FA09B9C9D9E9FA09B9C9D9E9FA09B, 4880 <119> 1445DB14CFDB45141445DB1412CFDB45, 1684 <163>  
1079 DATA 9C9D9E9FA09B9C9D9E9FA09B9C9D9E <251> 1116 DATA 14141414141445DB454545454545 <64>  
9FA09B9C9D9E9FA09B9C9D9E9FA09B9C, 4881 <251> 89DB45DBCF12CFDB12CFDB45DBCF12CFDB, 3590 <64>  
1080 DATA 9D9E9FA09B9C9D9E9FA09B9C9D9E9F <187> 1117 DATA 451414CF12CF12CF12CF12CF12CF <114>  
A09B9C9D9E9FA09B9C9D9E9FA09B9C9D, 4882 <187> DB45, 2181 <114>  
1081 DATA 9E9FA09B9C9D9E9FA09B9C9D9E9FA0 <183> ENDE DES LISTINGS  
9B9C9D9E9FA09B9C9D9E9FA09B9C9D9E, 4883 <183>  
1082 DATA 9FA09B9C9D9E9FA09B9C9D9E9F9B9B <210>  
9B9B9B9B9B9B9B9B9B9B9B9B9B9B9B9B, 4713 <210>  
1083 DATA 90919293948F90919293948F909192



# Listing Teil 5

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

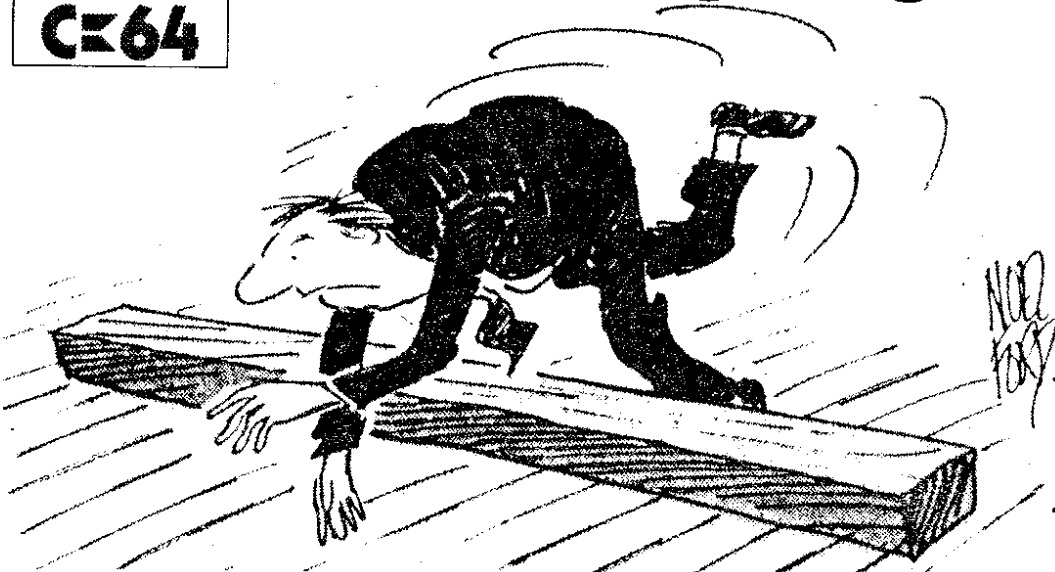
1 AD=29000:FORI=1000T01032:READA#,FS:C=0
:FORN=1T0LEN(A#)STEP2 <150>
2 A=ASC(MID$(A#,N,1))-48:B=ASC(MID$(A#,N
+1,1))-48:A=A+7*(A>9):B=B+7*(B>9) <161>
3 P%KEAD,16*A+B:C=C+PEEK(AD):AD=AD+1 <176>
4 NEXT:PRINTI;"(SPACE)";:IFC<>P$THENPRIN
T"ERROR":STOP <157>
5 PRINT"OK":NEXT <143>
6 PRINT"FERTIG":END <136>
1000 DATA 202020202020202020202020202020AB
AC20ADAD20AD202020202020202020202020, 1694 <113>
1001 DATA 20202020202020202020202020202020
202020202020202020AEAFB0B1CFB2B320, 2034 <97>
1002 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <65>
1003 DATA 20B4B5B6B7B8B9BA2020202020202020
20202020202020202020202020202020, 2049 <253>
1004 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <67>
1005 DATA 20202020202020202020202020202020
BBBCBD202020202020202020C6C720202020, 1793 <17>
1006 DATA 20202020202020202020202020202020
202020202020202020BEBFC0C1C2C3C4, 2119 <16>
1007 DATA C52020C8C9CACBCCDCEC320202020
20202020202020202020202020202020, 2517 <76>
1008 DATA 2020BBBCBD2020202020202020C6C720
20202020202020202020202020202020, 1793 <164>
1009 DATA 20202020202020202020202020BEBFC0C1
C2C3C4C52020C8C9CACBCCDCEC32020, 3644 <175>
1010 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <73>
1011 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <74>
1012 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <75>
1013 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <76>
1014 DATA 20202020208186878288888828888887
88878688888878D87888868286898A8788, 3670 <121>
1015 DATA 87878D868888886868D868287862020
84858E858783878E858485848584858E, 3976 <6>
1016 DATA 8584858E85888888C848584858E858488
878883878387888872020202020202020, 3355 <141>
1017 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <80>
1018 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <81>
1019 DATA 2020202020202020202020202020203420
203420202020342020203420202034, 1092 <117>
1020 DATA 202020203420202020342020202034
20202020203420343420202034202020, 1132 <206>
1021 DATA 342020203420345354203420203420
203420202020342020342020202034, 1275 <48>
1022 DATA 342020342020342020202034203420
20565758592020342034203420203420, 1394 <90>
1023 DATA 20202034202020203420342034203420
342034342034343434345A5B5C5D5E5F20, 1595 <110>
1024 DATA 343420342034202034342020343420
20203434343420342034343420343420, 1352 <183>
1025 DATA 343434343460616263203434203420
342034203420342020203420202034, 1534 <80>
1026 DATA 2020342020203420342020342020342020
34646534202034202034203420203434, 1349 <221>
1027 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <91>
1028 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <92>
1029 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <93>
1030 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <94>
1031 DATA 20202020202020202020202020202020
20202020202020202020202020202020, 992 <95>
1032 DATA 20202020202020202020202020202020, 256 <255>
ENDE DES LISTINGS

```

# Auf Kontrollgang im Wald

C64

Patrouille



Nach einem Einblick in die Spielanleitung muß man in einem Wald patrouillieren. Dabei müssen Sie verschiedene Hindernisse überwinden oder beseitigen, wie z.B.: Gräben, Holzstapel oder Felsbrocken. Viele dieser Hindernisse haben ihre ganz speziellen Tücken, die Sie erst im Laufe des Spiels kennenlernen. Zusatzpunkte gibt es, wenn ein entgegenfliegender Vogel abgeballert wird. Die Steuerung erfolgt über Joystick am CTRL-Port 2. Das Spiel ist vollständig in Maschinensprache geschrieben und ermöglicht so einen fließenden Ablauf.

## Bitte benutzen Sie den C64-Checksummer (S. 31)

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

100 REM *PATROUILLE* (C) RALF PROBST <19
86> <27>
120 FORT=49152T052603 <139>
140 READA:POKET,A:NEXT <116>
160 SYS49152 <74>
300 DATA32,230,201,32,156,198,169,32,141
,179,2,169,19,141,218,2,141,220 <216>
301 DATA2,141,225,2,141,229,2,169,22,141
,219,2,141,221,2,169,25,141 <225>
302 DATA223,2,141,224,2,141,226,2,169,29
,141,222,2,141,228,2,169,70 <48>
303 DATA141,182,2,141,183,2,169,33,141,1
84,2,141,227,2,169,160,141,230 <249>
304 DATA2,141,234,2,169,227,141,235,2,16
2,247,142,236,2,232,142,237,2 <33>
305 DATA169,98,141,238,2,169,121,141,239
,2,169,111,141,240,2,169,100,141 <207>
306 DATA241,2,169,80,141,0,208,169,178,1
41,1,208,169,194,141,3,208,169 <252>
307 DATA180,141,5,208,169,159,141,7,208,
169,175,141,11,208,169,253,141,28 <81>
308 DATA208,169,7,141,37,208,141,46,208,
169,0,141,38,208,169,67,141,29 <144>
309 DATA208,169,8,141,23,208,169,2,141,3
9,208,169,9,141,41,208,169,1 <175>
310 DATA141,42,208,141,43,208,141,44,208
,169,12,141,45,208,141,248,2,162 <81>
311 DATA10,142,32,208,142,33,208,169,255
,141,164,2,173,197,0,201,64,208 <125>
312 DATA249,169,147,32,210,255,238,134,2
,32,178,200,32,239,199,169,15,141 <28>
313 DATA214,0,169,13,141,211,0,32,16,229
,160,0,185,25,201,32,210,255 <10>
314 DATA200,192,11,208,245,173,197,0,201
,34,240,14,201,64,208,7,169,16 <184>
315 DATA44,0,220,208,203,76,59,193,160,0
,140,4,212,140,187,2,32,25 <229>
316 DATA199,32,22,200,162,0,142,198,0,23
4,174,198,0,240,250,172,187,2 <167>
317 DATA200,200,192,8,144,227,76,205,192
,169,147,32,210,255,169,0,162,9 <107>
318 DATA157,198,2,202,208,250,162,240,16
9,160,157,247,6,169,12,157,247,218 <209>
319 DATA202,208,243,169,140,141,77,7,141
,81,7,169,133,141,78,7,141,80 <147>
320 DATA7,141,90,7,141,91,7,141,106,7,16
9,150,141,79,7,169,138,141 <62>
321 DATAB9,7,169,147,141,93,7,169,144,14
1,101,7,141,92,7,169,149,141 <39>
322 DATA102,7,169,142,141,103,7,169,139,
141,104,7,169,148,141,105,7,169 <255>
323 DATA48,141,146,7,160,0,169,7,162,6,1
57,100,219,152,157,140,219,202 <95>
324 DATA169,7,157,76,219,157,100,219,157
,88,219,152,157,140,219,202,208,238 <35>
325 DATA142,16,208,142,251,0,142,32,208,
140,131,219,140,119,219,169,51,141 <180>
326 DATA131,7,232,142,119,7,142,21,208,1
42,168,2,142,169,2,142,232,2 <156>
327 DATA142,233,2,232,142,187,2,142,199,
2,162,15,142,40,208,142,33,208 <161>
328 DATA162,17,142,18,212,162,8,142,14,2
12,160,0,162,3,32,243,199,162 <149>
329 DATA255,142,230,2,142,231,2,238,250,
0,208,67,238,171,2,173,171,2 <83>
330 DATA41,1,240,57,238,119,7,160,10,140
,200,2,173,187,2,10,205,199 <236>
331 DATA2,144,22,173,199,2,10,201,128,20
8,6,206,248,2,76,83,194,141 <183>

```

```

332 DATA199,2,169,2,141,251,0,141,187,2,
13,199,2,13,251,0,141,251 <4>
333 DATA0,206,182,2,206,182,2,173,170,2,
208,3,32,153,200,173,250,0 <47>
334 DATA41,1,240,110,169,12,141,134,2,16
0,0,162,18,32,203,196,173,250 <227>
335 DATA0,41,2,240,66,32,204,200,169,13,
141,134,2,162,6,32,203,196 <194>
336 DATA202,208,250,172,198,2,174,184,2,
224,35,208,14,173,198,2,73,40 <1>
337 DATA141,198,2,168,162,33,142,184,2,1
38,153,159,4,232,232,152,24,105 <87>
338 DATA40,168,224,39,144,241,238,184,2,
173,27,212,41,1,141,233,2,174 <181>
339 DATA232,2,173,233,2,240,3,202,208,5,
232,224,9,240,248,142,232,2 <108>
340 DATA189,233,2,141,247,6,32,31,198,17
3,30,208,141,253,0,41,1,240 <119>
341 DATAB,77,253,0,45,170,2,240,26,173,2
53,0,174,170,2,208,4,41 <151>
342 DATA248,240,3,76,244,196,162,0,142,1
78,2,142,160,2,76,67,195,173 <168>
343 DATA253,0,45,160,2,208,6,238,160,2,7
6,67,195,173,178,2,208,226 <178>
344 DATA173,253,0,77,21,208,9,1,141,21,2
08,32,204,197,32,190,197,169 <173>
345 DATA1,141,21,208,141,178,2,206,131,7
,174,131,7,224,48,208,3,76 <84>
346 DATA191,198,169,200,141,248,7,206,18
3,2,208,66,169,255,77,21,208,45 <71>
347 DATA27,212,45,251,0,141,255,0,169,2,
162,0,44,255,0,240,31,13 <173>
348 DATA16,208,141,16,208,13,21,208,141,
21,208,169,63,157,2,208,173,27 <10>
349 DATA212,45,182,2,9,30,141,183,2,76,1
38,195,10,232,232,201,128,208 <230>
350 DATA213,169,1,141,183,2,169,64,44,21
,208,240,47,173,250,0,41,1 <202>
351 DATA240,40,173,169,2,174,168,2,240,4
,10,76,165,195,74,141,169,2 <29>
352 DATA41,62,208,11,173,168,2,73,1,141,
168,2,76,192,195,169,141,24 <173>
353 DATA109,169,2,141,13,208,169,128,44,
21,208,240,33,173,46,208,73,8 <139>
354 DATA141,46,208,173,14,208,24,105,4,2
05,14,208,144,6,141,14,208,76 <56>
355 DATA232,195,173,21,208,41,127,141,21
,208,169,8,44,0,220,208,14,173 <180>
356 DATA0,208,24,105,1,205,0,208,144,3,1
41,0,208,169,4,44,0,220 <38>
357 DATA208,13,173,0,208,56,233,1,201,24
,144,3,141,0,208,173,179,2 <144>
358 DATA201,32,208,12,174,177,2,240,48,1
69,1,44,0,220,208,47,41,1 <245>
359 DATA240,31,173,179,2,74,240,12,141,1
79,2,24,105,135,141,1,208,76 <214>
360 DATAB3,196,162,0,142,177,2,169,32,14
1,179,2,76,83,196,238,179,2 <51>
361 DATA76,83,196,238,1,208,238,1,208,16
9,16,44,0,220,208,34,169,128 <137>
362 DATA44,21,208,208,27,173,0,208,201,2
20,176,20,24,105,48,141,14,208 <137>
363 DATA174,1,208,142,15,208,169,128,13,
21,208,141,21,208,173,31,208,41 <126>
364 DATA1,240,9,141,177,2,206,1,208,76,1
43,196,238,1,208,173,200,2 <167>
365 DATA41,7,24,109,203,2,141,203,2,162,
0,32,217,196,173,200,2,41 <95>
366 DATAB,24,109,203,2,162,0,142,200,2,3
2,217,196,160,0,162,5,189 <50>
367 DATA202,2,24,105,48,153,141,7,200,20

```

2,208,243,172,248,2,32,22,198	<144>	406 DATA2,238;189,2,173,193,2,74,141,192	<14>
368 DATA76,17,194,32,240,255,169,32,32,2	<191>	,2,74,208,67,232,236,191,2	
10,255,169,20,32,210,255,96,201	<167>	407 DATA208,54,224,63,208,1,96,238,191,2	<193>
369 DATA10,144,251,56,233,10,157,203,2,1	<29>	,238,191,2,238,191,2,173,189	
89,204,2,24,105,1,157,204,2	<148>	408 DATA2,24,105,16,201,240,208,27,160,2	<117>
370 DATA224,3,240,232,232,76,217,196,173	<25>	54,200,200,185,250,0,24,105,240	
,21,208,41,127,141,21,208,173,170	<169>	409 DATA153,250,0,185,251,0,105,0,153,25	<22>
371 DATA2,208,25,168,32,168,198,173,253,	<8>	1,0,192,0,240,233,169,0,141	
0,41,120,240,24,201,32,208,3	<103>	410 DATA189,2,169,128,141,192,2,169,192,	<232>
372 DATA76,150,197,201,16,208,16,76,101,	<9>	141,193,2,45,192,2,141,192,2	
197,41,1,208,27,238,170,2,32	<37>	411 DATA138,168,177,254,160,0,45,193,2,2	<79>
373 DATA175,198,76,250,194,141,170,2,200	<97>	40,153,205,192,2,240,7,176,8	
,74,208,252,169,210,153,247,7,140	<132>	412 DATA160,7,76,232,199,172,203,2,152,1	<170>
374 DATA200,2,76,250,194,173,170,2,73,25	<116>	72,189,2,76,101,199,160,0,162	
4,45,21,208,141,21,208,45,16	<253>	413 DATAB,140,250,0,160,11,140,211,0,232	<213>
375 DATA208,141,16,208,172,170,2,192,9,2	<207>	,142,214,0,32,16,229,172,250	
40,45,192,33,208,3,32,190,197	<43>	414 DATA0,185,219,200,32,210,255,200,152	<51>
376 DATA162,205,32,158,198,162,0,142,170	<180>	,41,15,208,244,192,48,144,222,96	
,2,76,250,194,160,207,140,252,7	<40>	415 DATA238,164,2,208,14,162,0,189,36,20	<107>
377 DATA174,14,208,142,8,208,174,15,208,	<192>	1,240,91,32,210,255,232,76,29	
142,9,208,162,6,142,200,2,76	<212>	416 DATA200,162,7,142,211,0,162,3,142,21	<132>
378 DATA250,194,162,203,173,23,208,41,8,	<212>	4,0,32,16,229,162,0,189,170	
201,8,208,2,162,209,73,8,141	<35>	417 DATA201,32,210,255,232,224,6,208,245	<153>
379 DATA23,208,142,251,7,76,88,197,169,3	<17>	,174,164,2,224,2,240,15,176,26	
2,44,29,208,208,9,13,29,208	<230>	418 DATA169,176,141,250,0,169,201,141,25	<125>
380 DATA141,29,208,76,177,197,44,23,208,	<30>	1,0,76,112,200,169,188,141,250,0	
208,14,13,23,208,141,23,208,160	<239>	419 DATA169,201,141,251,0,76,112,200,169	<61>
381 DATA3,32,246,197,76,250,194,160,0,76	<215>	,200,141,250,0,169,201,141,251,0	
,39,197,169,67,141,29,208,169	<122>	420 DATA160,0,177,250,240,7,32,210,255,2	<50>
382 DATA223,45,23,208,141,23,208,96,160,	<8>	00,76,114,200,162,6,142,214,0	
0,140,4,212,162,240,142,6,212	<17>	421 DATA169,0,141,211,0,32,16,229,162,0,	<167>
383 DATA162,50,160,33,140,4,212,142,39,2	<230>	189,210,201,240,7,32,210,255	
08,142,1,212,138,160,10,32,22	<43>	422 DATA232,76,140,200,96,238,231,2,208,	<212>
384 DATA198,170,232,224,99,208,238,169,2	<180>	20,162,0,142,4,212,162,245,142	
10,141,248,7,160,25,140,254,0,162	<40>	423 DATA231,2,238,230,2,208,5,162,244,14	<3>
385 DATA20,142,1,212,162,129,142,4,212,1	<192>	2,230,2,169,15,141,24,212,160	
69,16,172,254,0,32,22,198,170	<212>	424 DATA33,140,5,212,140,6,212,174,230,2	<70>
386 DATA202,138,141,24,212,208,242,96,16	<233>	,189,230,1,141,1,212,140,4	
0,255,162,255,202,208,253,136,208,248	<35>	425 DATA212,96,160,30,185,32,50,89,191,5	<221>
387 DATA96,162,0,160,1,169,4,141,254,0,1	<17>	2,153,32,50,136,208,244,96,176	
52,10,168,232,232,201,128,240	<230>	426 DATA105,32,32,32,32,32,32,32,32,1	<255>
388 DATAB2,44,21,208,240,237,201,64,240,	<30>	05,174,174,32,32,171,107,117	
14,201,16,208,13,173,252,7,201	<215>	427 DATA174,123,176,105,117,105,174,174,	<147>
389 DATA204,240,6,238,254,0,238,254,0,18	<96>	174,125,125,117,105,93,32,106,189,173	
9,0,208,141,255,0,56,237,254	<147>	428 DATA93,32,106,107,106,177,173,173,17	<37>
390 DATA0,205,255,0,176,6,157,0,208,76,3	<239>	3,106,92,147,144,137,133,140,133,142	
5,198,72,152,77,21,208,13	<215>	429 DATA132,133,142,133,149,133,147,65,7	<57>
391 DATA16,208,141,21,208,41,4,240,21,17	<17>	8,76,69,73,84,85,78,71,32,63	
3,21,208,41,16,240,25,152,73	<230>	430 DATA19,144,83,73,69,32,72,65,66,69,7	<133>
392 DATA255,45,16,208,141,16,208,104,76,	<30>	8,32,68,73,69,32,65,85	
90,198,96,173,250,7,73,26,141	<239>	431 DATA70,71,65,66,69,32,68,73,69,83,69	<180>
393 DATA250,7,76,111,198,169,204,141,252	<215>	,78,32,74,69,69,80,32	
,7,169,180,141,9,208,76,118,198	<96>	432 DATA68,85,82,67,72,32,69,73,78,69,78	<236>
394 DATA162,200,138,157,48,7,232,224,210	<147>	,32,87,65,76,68,32,90	
,208,247,96,162,20,169,10,76,179	<200>	433 DATAB5,32,83,84,69,85,69,82,78,46,14	<231>
395 DATA198,162,220,169,5,142,1,212,141,	<54>	1,68,65,66,69,73,32,77	
24,212,169,129,141,4,212,96,162	<209>	434 DATAB5,69,83,83,69,78,32,83,73,69,32	<246>
396 DATA255,169,6,157,144,217,202,208,25	<169>	,86,69,82,83,67,72,73	
0,160,19,169,160,153,154,5,153,194	<133>	435 DATA69,68,69,78,69,32,72,73,78,68,69	<2>
397 DATA5,153,234,5,136,208,244,140,21,2	<147>	,82,45,32,32,78,73,83	
08,140,198,0,162,2,185,11,201	<230>	436 DATAB3,69,32,85,69,66,69,82,87,73,78	<240>
398 DATA157,194,5,192,5,176,9,153,68,6,1	<147>	,68,69,78,46,32,87,73	
85,20,201,153,62,6,200,232	<200>	437 DATA69,32,90,46,66,46,58,0,69,73,78,	<182>
399 DATA232,224,20,208,230,169,63,141,74	<54>	69,78,32,72,79,76,90	
,6,169,16,44,0,220,240,14,173	<209>	438 DATA83,84,65,80,80,69,76,0,70,69,76,	<224>
400 DATA197,0,201,64,240,242,201,39,208,	<169>	83,66,82,79,67,75,69	
3,32,226,252,76,3,192,96,162	<133>	439 DATA78,0,69,76,69,70,65,78,84,69,78,	<192>
401 DATA0,142,254,0,142,255,0,185,248,7,	<116>	0,18,84,65,83,84,69	
170,173,254,0,24,105,64,141		440 DATA17,157,157,157,68,82,85,69,67,75	<91>
402 DATA254,0,173,255,0,105,0,141,255,0,		,69,78,146,0,120,169,49,141	
202,208,236,185,39,208,141,203		441 DATA1,0,169,0,141,98,0,141,100,0,169	<49>
403 DATA2,169,147,32,210,255,169,128,141		,208,141,99,0,169,56,141	
,250,0,169,216,141,251,0,169,128		442 DATA101,0,162,16,160,0,177,98,145,10	<56>
404 DATA141,252,0,169,4,141,253,0,160,3,		0,200,208,249,238,99,0,238,101	
140,191,2,162,0,142,189,2		443 DATA0,202,208,240,169,55,141,1,0,169	<180>
405 DATA76,190,199,145,250,200,145,250,1		,31,141,24,208,88,162,0,189	
36,169,160,145,252,200,145,252,238,189		444 DATA94,202,157,224,56,232,224,16,208	



```

,245,189,94,202,157,248,56,232,224 <189>
445 DATA64,208,245,169,0,141,250,0,169,5 <90>
0,141,251,0,169,158,141,252,0
446 DATA169,202,141,253,0,160,0,162,4,17 <22>
7,252,145,250,200,208,249,238,251
447 DATA0,238,253,0,202,208,240,96,248,2 <79>
48,0,224,224,0,0,0,24,24
448 DATA24,24,24,0,0,0,1,1,1,3,7,15,7,15 <239>
,128,128,128,192
449 DATA224,240,224,240,15,31,63,31,63,6 <83>
3,127,127,240,248,252,248,252,252
450 DATA254,254,255,255,255,127,3,1,1,1, <62>
255,255,255,254,192,128,128,128
451 DATA0,0,0,0,0,0,3,240,0,15,8,0,15,8, <168>
0,51,2,0
452 DATA51,2,0,195,0,128,195,0,128,170,1 <200>
70,160,169,85,104,150,170,168
453 DATA170,170,171,171,170,235,175,235, <14>
251,47,235,251,13,195,112,13,195,112
454 DATA15,195,240,15,195,240,3,0,192,17 <129>
3,31,255,248,31,255,248,15,255
455 DATA248,15,255,240,15,255,224,7,255, <69>
224,3,255,224,1,255,224,0,255
456 DATA240,0,255,240,0,255,240,1,255,24 <213>
0,3,255,224,1,255,192,0,255
457 DATA128,0,63,0,0,63,0,0,28,0,0,0,0,0 <74>
,0,0,0
458 DATA0,240,0,0,0,0,0,0,0,0,0,0,0,0,0 <158>
0,0,0
459 DATA0,0,0,0,0,0,0,0,40,0,0,150,0,0 <249>
,150,0,0
460 DATA150,0,2,170,128,9,105,96,9,105,9 <94>
6,9,105,96,42,170,168,150
461 DATA150,150,150,150,150,150,150,150, <44>
40,40,40,32,0,0,192,0,0,192
462 DATA0,0,192,3,0,192,3,0,192,3,192,19 <139>
2,3,204,192,3,207,192
463 DATA3,255,192,3,255,192,3,255,192,3, <53>
255,192,3,255,192,3,255,192
464 DATA3,255,192,3,255,192,3,255,192,3, <236>
255,192,3,255,192,15,255,240
465 DATA63,255,252,170,0,60,0,0,255,0,3, <2>
251,192,3,250,192,3,250
466 DATA192,15,250,240,15,250,240,15,250 <143>
,240,15,254,176,15,254,176,63,254
467 DATA188,63,254,188,63,254,188,63,254 <218>
,188,63,254,188,63,254,188,63,254
468 DATA188,63,254,188,63,254,188,63,254 <139>
,188,63,254,188,83,0,255,0,3
469 DATA85,192,13,85,112,53,105,92,53,17 <60>
0,92,53,170,92,53,170,92,53

```

```

470 DATA170,92,53,170,92,53,170,92,53,17 <130>
0,92,53,170,92,13,105,112,13
471 DATA85,112,13,85,112,3,85,192,3,85,1 <190>
92,0,215,0,0,215,0,0
472 DATA60,0,0,60,0,197,0,0,0,0,0,0,0,0, <75>
0,0,0,0
473 DATA0,0,0,0,0,0,0,160,0,10,160,0,42, <27>
170,160,174,170,168
474 DATA170,170,170,170,170,170,174,170, <88>
170,240,170,170,128,170,170,128,170,168
475 DATA128,160,40,128,160,40,0,160,40,0 <42>
,160,40,0,240,60,197,0,0
476 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <4>
0,0
477 DATA0,160,192,0,171,255,192,42,255,2 <106>
40,171,255,192,160,192,0,0,0
478 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <6>
0,0
479 DATA0,0,0,0,0,0,0,171,0,0,0,0,0,0,0, <96>
0,0,0
480 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <8>
0,0
481 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,20 <49>
,0,0
482 DATA85,0,2,150,128,2,170,128,10,170, <217>
160,10,170,160,42,170,168,173
483 DATA0,63,255,0,234,171,0,234,188,12, <117>
234,192,59,170,255,218,170,171
484 DATA59,170,255,12,234,192,0,234,188, <15>
0,234,171,0,63,255,0,0,0
485 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <13>
0,0
486 DATA0,0,0,0,0,0,0,0,0,151,0,0,0,48,0 <91>
,240,60,3
487 DATA192,55,3,204,55,13,204,53,205,19 <16>
2,13,119,0,13,87,0,3,85
488 DATA195,0,215,12,0,213,240,63,85,192 <16>
,15,87,0,0,220,0,0,215
489 DATA0,3,127,3,207,0,15,15,0,15,3,0 <213>
,48,0,0,48,0
490 DATA0,162,1,0,64,5,65,80,4,65,16,2,0 <73>
,128,2,0,128,2
491 DATA0,128,2,0,128,12,195,48,15,195,2 <42>
40,3,0,192

```

ENDE DES LISTINGS



## Die C64-Werkstatt: Tips & Tricks

### Sprite Turn und Character Switch: Zwei nützliche Routinen

#### Sprite Turn-Demo

Problemlösungen an einem C-64 waren die Vorreiter für die beiden Maschinensprachen-Utilities.

Bei der 1. Routine handelt es sich um ein Utility, welches ab 49152 (\$C000) im Speicher liegt, und somit keinen Basicspeicher verbraucht. Die Routine wird nach dem Einpoken mit SYS 49152 (CR) gestartet. Nach dem Aufruf bewirkt ein Drücken der ↑-Taste ein Um-

schalten des Zeichensatzes. Dieses geschieht im Klein-, wie auch im Großschritmodus. Der Zeichensatz, zu dem umgeschaltet werden kann, liegt ab 16384 im Speicher. Ein weiterer Druck auf die ↑-Taste bewirkt ein Zurückschalten auf den alten Zeichensatz. Dieses kann nun zur Anwendung kommen, z.B. in Textverarbeitungsprogrammen, wo schnell ein weiterer Zeichensatz zugeschaltet werden muß.

Bei dem 2. Programm handelt es

sich um Sprite Turn, welches es ermöglicht einen Sprite schnell zu drehen. Dieses kommt sehr häufig in Spielen vor, wenn ein Richtungswechsel vollzogen werden muß. Es braucht also nicht mehr mühselig ein zweiter Sprite entwickelt zu werden, der nie ganz so aussieht

wie der Erste. Nach dem Einpoken ab 49152 (\$C000) in den Speicher kann 'Sprite Turn' mit SYS 49152, Block (CR) gestartet werden. Wo bei Block den Spriteblock angibt, in dem sich der zu drehende Sprite befindet.

#### C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 REM ***** <185>
20 REM * CHARAKTER SWITCH * <129>

```

```

30 REM * ----- * <214>
40 REM * (C) 1985 BY * <13>
50 REM * JOERG TUTTAS * <117>
60 REM * * <157>
70 REM ***** <245>
80 : <138>
90 FOR T=49152 TO 49370:READ Q:S=S+Q <203>
100 POKE T,Q:NEXT <153>
110 IF S<>29039 THEN PRINT"FEHLER{SPACE} <50>
IN{SPACE}DATAS":STOP <178>
120 :
1000 DATA120,165, 1,141, 60, 3, 41,251,1 <144>
33, 1,169, 0,133,250,169,208,133
1010 DATA251,169, 0,133,252,169, 48,133, <6>
253,162, 0,160, 0,177,250,145,252
1020 DATA200,240, 3, 76, 30,192,232,230, <158>
251,230,253,224, 16,208,235,173, 60
1030 DATA 3,133, 1,173, 24,208, 41,241, <15>
9, 12,141, 24,208,169, 76,141, 20
1040 DATA 3,169,192,141, 21, 3, 88, 96,1 <16>
65,203,201, 54,240, 3, 76, 49,234
1050 DATA173, 24,208,141, 63, 3,173, 61, <207>
3,201, 1,240, 45,169, 0,133,254
1060 DATA133,252,169, 64,133,255,169, 48 <157>
,133,253,169, 1,141, 61, 3,162, 0
1070 DATA160, 0,177,254,145,252,200,240, <37>
3, 76,121,192,232,230,253,230,255
1080 DATA224, 16,208,235, 76,209,192,169 <92>
, 0,141, 61, 3,169, 0,133,250,133
1090 DATA254,169, 48,133,255,169,208,133 <199>
,251,165, 1,141, 60, 3, 41,251,133
1100 DATA 1,162, 0,160, 0,177,250,145,25 <165>
4,200,240, 3, 76,175,192,232,230
1110 DATA251,230,255,224, 16,208,235,173 <1>
, 60, 3,133, 1,173, 24,208, 41,241
1120 DATA 9, 12,141, 24,208,173, 63, 3,1 <253>
41, 24,208, 76, 49,234, 62
ENDE DES LISTINGS

```

## Character Switch DEMO

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 REM ***** <185>
20 REM * CHARAKTER SWITCH DEMO * <143>
30 REM * ----- * <238>
40 REM * (C) 1986 BY * <22>
50 REM * JOERG TUTTAS * <117>
60 REM * * <157>
70 REM ***** <245>
80 POKE 53281,0:POKE 53280,0:PRINT"WHIT <75>
E CLEAR"CHR$(14)
90 PRINT"DOWN SPACE)ITTE{SPACE}WARTEN; <6>
POKE{SPACE}JETZT{SPACE}-L{SPACE}O,|-|"
100 PRINT"DOWN SPACE)AB{SPACE}49152{SPA <44>
CE){SPACE}*000{SPACE}){SPACE}IN{SPACE}
DEN{SPACE}*PEICHER"
110 FOR T=49152 TO 49370:READ Q:S=S+Q <223>
120 POKE T,Q:NEXT <173>
130 IF S<>29039 THEN PRINT"FEHLER{SPACE} <70>
IN{SPACE}DATAS":STOP
140 : <198>
150 PRINT"CLEAR" <180>
160 PRINT"DOWN SPACE)*NSCHLIESSEND{SPAC <223>
E)LESE{SPACE}ICH{SPACE}EINEN{SPACE}*EICH
EN-"
170 PRINT"DOWN SPACE)SATZ{SPACE}EIN{SPA <242>
CE),DER{SPACE}NARTUERLICH
180 PRINT"DOWN SPACE)AUCH{SPACE}FREI{SP <188>
ACE)PROGRAMMIERBAR{SPACE}WAERE.

```

```

190 PRINT"DOWN SPACE)-ER{SPACE}*EICHENS <100>
ATZ{SPACE)LIEGT{SPACE}AB{SPACE}16384{SPA
CE)UND"
200 PRINT"DOWN SPACE)BELEGT{SPACE}512{S <168>
PACE}|YTES{SPACE}(ROSSCHRIFT7-LEIN-"
210 PRINT"DOWN SPACE)SCHRIFT) <240>
220 GOSUB 360 <49>
230 DATA120,165, 1,141, 60, 3, 41,251,13 <139>
3, 1,169, 0,133,250,169,208,133
240 DATA251,169, 0,133,252,169, 48,133,2 <1>
53,162, 0,160, 0,177,250,145,252
250 DATA200,240, 3, 76, 30,192,232,230,2 <153>
51,230,253,224, 16,208,235,173, 60
260 DATA 3,133, 1,173, 24,208, 41,241, 9 <10>
, 12,141, 24,208,169, 76,141, 20
270 DATA 3,169,192,141, 21, 3, 88, 96,16 <11>
5,203,201, 54,240, 3, 76, 49,234
280 DATA173, 24,208,141, 63, 3,173, 61, <202>
3,201, 1,240, 45,169, 0,133,254
290 DATA133,252,169, 64,133,255,169, 48, <152>
133,253,169, 1,141, 61, 3,162, 0
300 DATA160, 0,177,254,145,252,200,240, <32>
3, 76,121,192,232,230,253,230,255
310 DATA224, 16,208,235, 76,209,192,169, <87>
0,141, 61, 3,169, 0,133,250,133
320 DATA254,169, 48,133,255,169,208,133, <194>
251,165, 1,141, 60, 3, 41,251,133
330 DATA 1,162, 0,160, 0,177,250,145,254 <160>
,200,240, 3, 76,175,192,232,230
340 DATA251,230,255,224, 16,208,235,173, <252>
60, 3,133, 1,173, 24,208, 41,241
350 DATA 9, 12,141, 24,208,173, 63, 3,14 <248>
1, 24,208, 76, 49,234, 62
351 : <154>
352 REM DATAS DES ZEICHENSATZES <254>
353 : <156>
360 FOR T=0 TO 640:READ Q:POKE 16384+T,Q <136>
:NEXT <135>
365 PRINT"CLEAR":SYS 49152
366 PRINT"DOWN SPACE)NUN{SPACE}WIRD{SP <206>
ACE)BEI{SPACE}JEDEM{SPACE}DRUECKEN{SPACE
}DER{SPACE}TASTE"
367 PRINT"DOWN SPACE)^^{SPACE)DER{SPAC <164>
E)ZEICHENSATZ{SPACE)UMGESCHALTET."
368 PRINT"DOWN SPACE)SOMIT{SPACE)BIETET <77>
{SPACE)SICH{SPACE)DEM{SPACE)ANWENDER{SPA
CE)DIE"
369 PRINT"DOWN SPACE)MOEGlichkeit{SPACE <245>
)ZWEI{SPACE)ZEICHENSATZE{SPACE)IM"
370 PRINT"DOWN SPACE)IM{SPACE)SPEICHER{ <129>
SPACE)DES{SPACE)C{SPACE)64{SPACE)ZU{SPAC
E)HABEN{SPACE)UND{SPACE)"
371 PRINT"DOWN SPACE)DIESE{SPACE)NACH{S <24>
PACE)BELIEBEN{SPACE)UMZUSCHALTEN."
372 END <245>
379 DATA162,85,138,44,169,86,147,106,60, <6>
126,102,126,230,230,247,0,124,102
380 DATA118,124,246,230,252,0,60,110,224 <83>
,224,224,238,124,0,124,110,103,103
390 DATA231,238,252,0,127,99,104,120,104 <184>
,99,127,0,127,99,104,120,104,224
400 DATA240,0,60,110,224,238,230,230,124 <252>
,0,102,102,118,126,118,230,247,0
410 DATA126,24,24,24,56,60,126,0,62,12,1 <109>
2,14,14,110,60,0,102,110,120,112
420 DATA248,238,230,0,112,96,96,96,224,2 <179>
30,254,0,99,119,127,107,99,231,247
430 DATA0,102,118,126,126,110,230,247,0, <135>
60,102,231,231,231,102,60,0,124,118
440 DATA102,124,96,224,240,0,60,102,231, <173>
231,126,61,15,0,124,118,102,124,120
450 DATA238,230,0,62,100,96,60,6,102,252 <215>
,0,255,219,24,24,24,56,60,0,102,102
460 DATA102,102,230,246,252,0,102,102,10 <130>
2,102,246,252,120,0,99,99,99,107
470 DATA255,247,227,0,231,102,60,24,60,1 <74>
02,231,0,102,102,126,60,24,56,60

```

# programme

```

480 DATA0,127,102,12,24,48,102,254,0,62,
190,254,243,119,126,62,62,4,7,3,1 <54>
490 DATA0,0,0,0,166,149,74,176,139,93,50 <32>
,15,10,21,28,53,28,23,11,14,1,6,9
500 DATA23,29,48,102,83,0,0,0,0,0,0,0,0, <165>
0,0,24,108,207,52,157,103,0,0,192
510 DATA96,160,208,112,184,96,208,168,52 <83>
,156,84,172,216,84,136,168,80,224
520 DATA128,128,0,62,62,62,62,126,127,25 <40>
5,255,27,222,220,124,60,189,255,126
530 DATA0,0,128,192,96,96,48,48,192,96,5 <205>
1,30,28,28,28,0,0,0,2,55,255
540 DATA255,0,0,0,192,224,255,255,255,0, <8>
24,24,126,24,24,0,0,56,254,255,255
550 DATA255,127,255,112,0,0,0,0,24,63,25 <199>
5,255,255,253,127,191,199,0,0,0,255
560 DATA237,231,31,7,3,1,1,60,102,110,11 <84>
8,230,238,252,0,24,56,120,120,24
570 DATA60,60,0,62,102,102,12,48,102,254 <28>
,0,124,102,6,28,6,103,127,0,14,30
580 DATA54,102,255,6,15,0,126,96,124,30, <171>
14,102,124,0,60,118,96,124,102,230
590 DATA252,0,126,102,12,24,24,56,60,0,1 <82>
26,102,102,60,102,230,254,0,124,108
600 DATA108,60,12,108,124,0,255,255,255, <173>
255,248,240,192,192,255,191,127,236
610 DATA220,112,0,0,251,254,56,0,0,0,0,0 <200>
,255,255,254,238,192,0,0,0,0,126
620 DATA255,255,129,231,255,255,223,239, <10>
123,31,3,1,0,0,24,24,24,24,60
630 DATA126,102,0,0,192,248,127,248,192, <226>
0,0,0,3,31,254,31,3,0,0,0,28,62,255
640 DATA255,255,255,0,128,192,232,255,25 <20>
5,255,255,128,192,192,224,252,252
650 DATA254,255,0,1,15,31,63,255,255,255 <164>
,1,1,3,7,15,63,127,255,0,0,0,0
660 DATA0,46,255,255,255,231,193,241,121 <165>
,255,63,237,127,249,241,225,231,245
670 DATA243,28,63,127,255,127,63,14,3,25 <66>
5,255,255,252,232,0,0,0,24,36,90
680 DATA177,153,66,36,24,255,207,95,243, <95>
227,199,207,255,255,247,241,249,231
690 DATA241,255,255,255 <196>
700 : <248>
ENDE DES LISTINGS

```

```

200 DATA 62, 3,145,250, 32, 92,192,173, <80>
62, 3,164,252,145,250,200,177,250
210 DATA141, 61, 3, 32, 92,192,173, 62, <128>
3,164,252,200,145,250,164,252,200
220 DATA200,200,192, 64,144,196, 96,162, <193>
0,142, 62, 3,169, 7,141, 63, 3
230 DATA160, 0,173, 61, 3, 10,200,204, 6 <164>
3, 3,208,249, 41,128,201,128,240
240 DATA 16,206, 63, 3,232,224, 8,208,23 <34>
0,173, 61, 3, 41,128,208, 23, 96
250 DATA160, 0,169, 1, 24, 10,200,204, 6 <252>
3, 3,208,249, 24,109, 62, 3,141
260 DATA 62, 3, 76,120,192,238, 62, 3, 9 <137>
6,238
ENDE DES LISTINGS

```

## DEMO

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 REM ----- <125>
20 REM * SPRITE TURN DEMO * <223>
30 REM - (C) 1986 BY - <54>
40 REM * JOERG TUTTAS * <107>
50 REM ----- <165>
60 : <118>
70 REM START MIT SYS 49152,BLOCK <132>
80 : <138>
90 POKE 53281,0:POKE 53280,0:PRINT"(WHIT <85>
E CLEAR)"CHR$(14)
100 PRINT"(CLEAR DOWN SPACE)|ITTE(SPACE) <48>
WARTEN;LESE(SPACE)JETZT(SPACE)*PRITE(SPA
CE)|URN"
110 PRINT"(DOWN SPACE)AB(SPACE)49152(SPA <175>
CE)($-000)(SPACE)IN(SPACE)DEN(SPACE)*PEI
CHER(SPACE)UND"
120 PRINT"(DOWN SPACE)POKE(SPACE)ANSCHL. <72>
NOCH(SPACE)DEN(SPACE)-EMOSPRITE(SPACE)AB
(SPACE)B32"
130 PRINT"(DOWN SPACE)|(LOCK(SPACE)13)(S <72>
PACE)IN(SPACE)DEN(SPACE)*PEICHER"
140 FOR T=49152 TO 49314 <217>
150 READ AD:SU=SU+AD:POKE T,AD <251>
160 NEXT T <202>
170 IF SU<>20957 THEN PRINT"FEHLER(SPAC <39>
E)IN(SPACE)DATAS":STOP
180 PRINT"PROGRAMM(SPACE)OK!!" <187>
200 DATA 32,253,174, 32,158,183,138,160, <80>
0,132,250,132,251, 24, 42,133,250
210 DATA165,251, 42,133,251,165,250,200, <208>
192, 6,208,240,160, 0,132,252,177
220 DATA250,141, 61, 3, 32, 92,192,164,2 <174>
52,200,200,177,250,141, 61, 3,173
230 DATA 62, 3,145,250, 32, 92,192,173, <110>
62, 3,164,252,145,250,200,177,250
240 DATA141, 61, 3, 32, 92,192,173, 62, <158>
3,164,252,200,145,250,164,252,200
250 DATA200,200,192, 64,144,196, 96,162, <223>
0,142, 62, 3,169, 7,141, 63, 3
260 DATA160, 0,173, 61, 3, 10,200,204, 6 <195>
3, 3,208,249, 41,128,201,128,240
270 DATA 16,206, 63, 3,232,224, 8,208,23 <65>
0,173, 61, 3, 41,128,208, 23, 96
280 DATA160, 0,169, 1, 24, 10,200,204, 6 <27>
3, 3,208,249, 24,109, 62, 3,141
290 DATA 62, 3, 76,120,192,238, 62, 3, 9 <167>
6,238
300 FORJ=0TD62:READQ:POKE B32 +J,Q:POKE <196>
B96+J,Q:NEXT
310 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, <181>
0,128,31,248,128,18,12,128,34
320 DATA10,128,226,9,252,253,247,254,253 <246>
,119,254,121,247,62,118,238,220
330 DATA6,0,192,0,0,0,0,0,0,0,0,0,0,0,0, <47>
0,0,0,0,0,0

```

## Sprite Turn

C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 REM ----- <123>
20 REM * SPRITE TURN * <138>
30 REM - (C) 1986 BY - <117>
40 REM * JOERG TUTTAS * <7>
50 REM ----- <165>
60 REM * JOERG TUTTAS * <134>
70 : <128>
80 REM START MIT SYS 49152,BLOCK <142>
90 : <148>
100 FOR T=49152 TO 49314:READ A:S=S+A:PO <132>
KE T,A:NEXT
110 IF S<>20957 THEN PRINT"(DOWN2)??(SPA <67>
CE)FEHLER(SPACE)IN(SPACE)DATAS":STOP
120 PRINT"PROGRAMM(SPACE)OK(SPACE)!!":EN <55>
D <188>
130 : <11>
140 REM M-PROGRAMM <208>
150 :
170 DATA 32,253,174, 32,158,183,138,160, <50>
0,132,250,132,251, 24, 42,133,250
180 DATA165,251, 42,133,251,165,250,200, <178>
192, 6,208,240,160, 0,132,252,177
190 DATA250,141, 61, 3, 32, 92,192,164,2 <144>
52,200,200,177,250,141, 61, 3,173

```

```

340 PRINT "{RVSON SPACE} | ASTE {SPACE} RVSDU
E}";GOSUB 590 <133>
350 PRINT "{CLEAR DOWN SPACE} /ACHDEM {SPAC
E} DIESES {SPACE} NUN {SPACE} GESCHEHEN {SPACE
} IST {SPACE}" <157>
360 PRINT "{DOWN SPACE} KANN {SPACE} DER {SPA
CE} *PRITE {SPACE} POSITIONIERT {SPACE} U. {SP
ACE} EIN-" <195>
370 PRINT "{DOWN SPACE} GESCHATET {SPACE} WE
RDEN. {SPACE}" <189>
380 PRINT "{DOWN SPACE} --IESES {SPACE} IST {S
PACE} NUN {SPACE} AUCH {SPACE} GESCHEHEN {SPAC
E} UND {SPACE} MAN" <234>
390 PRINT "{DOWN SPACE} KANN {SPACE} *PRITE {
SPACE} | URN {SPACE} AUFRUFEN. -ER {SPACE} *PRI
TE" <18>
400 PRINT "{DOWN SPACE} LIEGT {SPACE} IM {SPA
CE} | LOCK {SPACE} 13 {SPACE} SOMIT {SPACE} LAUT
ET {SPACE} DER" <90>
410 PRINT "{DOWN SPACE} * | * {SPACE} *UFRUF {S
PACE} * | * {SPACE} 49152,13 {SPACE} (-). ." <221>
420 POKE 53248+21,1:POKE 53248,150:POKE
53249,200:POKE 2040,13 <114>
430 FOR T=1 TO 200:NEXT:SYS 49152,13 <185>
440 PRINT "{DOWN RVSON SPACE} | ASTE {SPACE
RVSDU}";GOSUB 590 <136>
450 PRINT "{CLEAR DOWN SPACE} -S {SPACE} KAN
N {SPACE} ALSO {SPACE} MIT {SPACE} | ILFE {SPACE
} VON {SPACE} *PRITE {SPACE} -" <73>
460 PRINT "{DOWN SPACE} | URN {SPACE} SCHNELL
{SPACE} UND {SPACE} EINFACH {SPACE} EIN {SPACE
} 2. ." <225>
470 PRINT "{DOWN SPACE} *PRITE {SPACE} ENTST
EHEN {SPACE} DER {SPACE} GENAU {SPACE} *EITENV
ER-" <181>
480 PRINT "{DOWN SPACE} KEHRT {SPACE} IST {SP
ACE} ALS {SPACE} DER {SPACE} *AUSGANGSSPRITE" <188>
490 GET A$:IF A$<>"" THEN 560 <127>
500 SYS 49152,13:FOR T=1 TO 255:POKE 532
48,T <151>
510 GET A$:IF A$<>"" THEN 560 <147>
520 NEXT <140>
530 SYS 49152,13:FOR T=255 TO 1 STEP-1:P
OKE 53248,T <123>
540 GET A$:IF A$<>"" THEN 560 <178>
550 NEXT:GOTO 490 <198>
560 PRINT "{CLEAR DOWN SPACE} .ND {SPACE} NU
N {SPACE} DAS {SPACE} -EMOPROGRAMM {SPACE} ZUR
{SPACE} -IN-" <154>
570 PRINT "{DOWN SPACE} SICHT....." <240>
580 FOR T=1 TO 3000:NEXT T:PRINT "{HOME D
OWN26}":LIST <83>
590 GET A$:IF A$="" THEN 590 <68>
600 RETURN <232>
ENDE DES LISTINGS

```

## Disk without a Directory

Vielleicht gibt es jetzt einen dauerhaften Schutz vor Raubkopieren. Das Programm "Disk without a Directory" formatiert Disketten so, daß das Directory nicht vorhanden ist. Aber dennoch

können Programme geladen oder abgespeichert werden. Die lieben Cracker-Freunde werden aber mit Sicherheit auch hiervon nicht passen, oder doch?

\*\*\*

### C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 POKE53280,12:POKE53281,12:PRINT "{BLAC
K}";CHR$(142);CHR$(8) <70>
20 PRINT "{CLEAR DOWN2 SPACES} DISK {SPACE}
WITHOUT {SPACE} A {SPACE} DIRECTORY {SPACE} V {
SPACE} 1.0" <202>
30 PRINT "{SPACES}" <172>
40 PRINT "{SPACE7} COPYRIGHT {SPACE} 1986 {SP
ACE} BY {SPACE} K. PRELLE" <74>

```

```

50 PRINT "{DOWN2 SPACE4} DIESES {SPACE} PRG.
{SPACE} FORMATIERT {SPACE} DIE" <29>
60 PRINT "{SPACE4} DISKETTE {SPACE} OHNE {SPA
CE} DIRECTORY" <27>
70 PRINT "{DOWN SPACE4} NAMENS {SPACE} & {SPA
CE} ID {SPACE} EINGABE {SPACE}?" <194>
80 GET A$:IFA$="" THENBO <83>
90 IFA$="J" THEN 120 <228>
100 IFA$="N" THEN N$="*****":I
$="00":GOTO 140 <129>
110 GOTOBO <247>
120 INPUT "{DOWN SPACE4} DISKETTENNAME";N$ <120>
130 INPUT "{DOWN SPACE4} ID";I$ <7>
140 OPEN2,8,15:PRINT "{DOWN SPACE4} DISK {S
PACE} WIRD {SPACE} FORMATIERT" <200>
150 PRINT#2,"NO: "+CHR$(20)+CHR$(20)+CHR$(
20)+CHR$(0)+CHR$(0)+CHR$(0)+N$+" "+I$ <37>
160 CLOSE2:PRINT "{DOWN SPACE4} OK. {SPACE}
NOCH {SPACE} EINE {SPACE} DISK {SPACE}?" <213>
170 GETB$:IFB$="" THEN170 <92>
180 IFB$="J" THEN20 <37>
190 IFB$="N" THENPRINT "{CLEAR}":END <53>
200 GOTO170 <24>
ENDE DES LISTINGS

```

## Directory-Retter

Das vorliegende Programme regeneriert Disketten, die versehentlich formatiert wurden. Dies geht natürlich nur, wenn die Diskette ohne ID formatiert wurde.

Vergewissern Sie sich, daß sich auf der Diskette nicht schon ein File mit der Bezeichnung "↑DR↑" ("↑" bedeutet hier "Pfeil nach oben"). Falls doch, können Sie

über Menüpunkt 3 einen anderen Filenamen mit Max. 4 Zeichen wählen. Danach wählen Sie aus dem Menü den Punkt 1.

Die Daten werden nun ausgelesen und als File abgespeichert. Mit Menüpunkt 2 bringen Sie die Directory nun in die endgültige Fassung.

\*\*\*

### C64 LISTING MIT CHECKSUMMEN (C V1.0)

```

10 REM ***** <183>
20 REM * <117>
30 REM * DIRECTORY-RETTNER * <27>
40 REM * * <137>
50 REM * 08. JANUAR 1985 * <103>
60 REM * * <157>
70 REM * GERD HOEHN * <67>
80 REM * * <177>
90 REM * KATH.-DIEZ-STR. 13 * <253>

```

```

100 REM * 5902 NETPHEN 1 * <178>
110 REM * TEL. 02738/1357 * <246>
120 REM * * <217>
130 REM * FUER C-64 * <201>
140 REM * * <237>
150 REM ***** <67>
160 : <218>
170 IFLA=1THENLA=0:GOTO700 <154>
180 POKE56,100:CLR:M=49341:GOSUB1020 <206>
190 N$="↑DR↑":N(1)=94:N(2)=68:N(3)=82:N(
4)=94 <148>

```

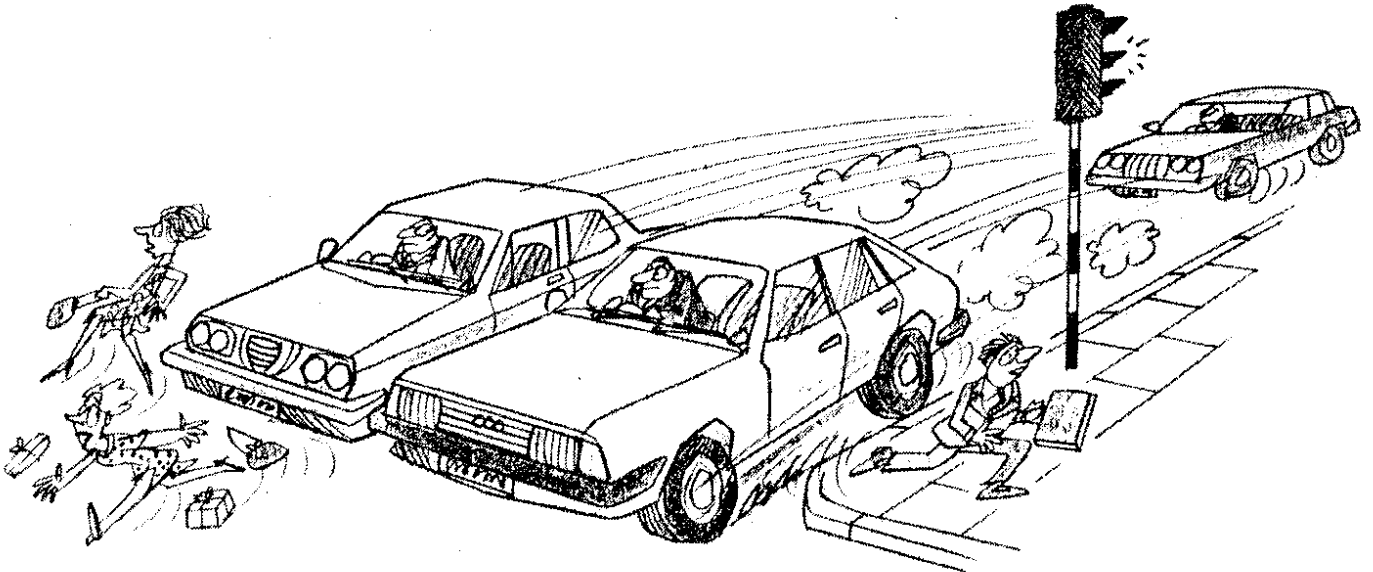
# programme

```

200 PRINT " (CLEAR) ":Y=3:GOSUBB60:SYSM,4,4
:PRINT"D (SPACE) I (SPACE) R (SPACE) E (SPACE) C
 (SPACE) T (SPACE) O (SPACE) R (SPACE) Y (SPACE) -
 (SPACE) R (SPACE) E (SPACE) T (SPACE) T (SPACE) E
 (SPACE) R" <242>
210 PRINT " (DOWN2 SPACE3) 1 (SPACE) = (SPACE)
VORHANDENE (SPACE) DIRECTORY (SPACE) SICHERN
" <201>
220 PRINT " (DOWN SPACE3) 2 (SPACE) = (SPACE) G
ELOESCHTE (SPACE) DIRECTORY (SPACE) ERNEUERN
" <6>
230 PRINT " (DOWN SPACE3) 3 (SPACE) = (SPACE) N
EUER (SPACE) NAME (SPACE) (MAX. (SPACE) 4 (SPAC
E) ZEICHEN) " <151>
240 Y=13:GOSUBB60 <100>
250 SYSM,14,3:PRINT"BITTE (SPACE) WAEHL EN (C
SPACE19)" <243>
260 GETA$ <148>
270 IFA$="1" THEN GOSUB980:GOTO340 <250>
280 IFA$="2" THEN GOSUB980:GOTO570 <88>
290 IFA$="3" THEN 920 <107>
300 SYSM,14,3:PRINT" (SPACE32)" <89>
310 GOTO250 <131>
320 : <123>
330 REM ===DIR.-BLOECKE LADEN===== <81>
340 SYSM,14,3:PRINT"DIRECTORY-BLOECKE (SP
ACE) LESEN (SPACE2)" <184>
350 GOSUB790 <197>
360 : <163>
370 REM ===DIR.-BL. AUF DISK SCHR.===== <5>
380 SYSM,14,3:PRINT"DIR.-BLOECKE (SPACE) A
UF (SPACE) DISK (SPACE) SCHREIBEN (SPACE)" <29>
390 POKE782,101+I:SYS49189 <122>
400 : <203>
410 REM ===PROG. - ^DR^ - SUCHEN===== <23>
420 SYSM,14,3:PRINT"FILE (SPACE) "N$ (SPAC
E) SUCHEN (SPACE17)" <170>
430 GOSUB790:I=25861 <6>
440 IFPEEK (I) <> N (1) ANDPEEK (I+1) <> N (2) THE
NI=I+32:GOTO440 <194>
450 IFPEEK (I+2) <> N (3) ANDPEEK (I+3) <> N (4) T
HENI=I+32:GOTO440 <84>
460 T=PEEK (I-2):S=PEEK (I-1):BL=PEEK (I+25
) <187>
470 : <17>
480 REM ===TR./SE./BL. AUF TR.18,1===== <6>
490 SYSM,14,3:PRINT"TR/SEC/BL (SPACE) AUF (
SPACE) TR (SPACE) 18,0 (SPACE) SCHREIBEN (SPAC
E)" <119>
500 PRINT#1,"U1 (SPACE) 2 (SPACE) 0"18,0:PRI
NT#1,"B-P (SPACE) 2 (SPACE) 180" <115>
510 PRINT#2,CHR*(T);CHR*(S);CHR*(BL); <6>
520 PRINT#1,"U2 (SPACE) 2 (SPACE) 0"18,0:CLO
SE2:CLOSE1 <190>
530 SYSM,14,3:PRINT"DIRECTORY (SPACE) IST (
SPACE) AUF (SPACE) DISK (SPACE) GESICHERT" <220>
540 END <158>
550 : <98>
560 REM ===TR./SE./BL. LADEN===== <101>
570 SYSM,14,3:PRINT"TR/SEC/BL (SPACE) VON (
SPACE) TR (SPACE) 18,0 (SPACE) LESEN (SPACE5)" <162>
580 GOSUB790 <172>
590 PRINT#1,"U1 (SPACE) 2 (SPACE) 0"18,0:PRI
NT#1,"B-P (SPACE) 2 (SPACE) 0" <223>
600 PRINT#2,CHR*(18);CHR*(1); <224>
610 PRINT#1,"U2 (SPACE) 2 (SPACE) 0"18,0 <129>
620 SYSM,14,3:PRINT"NAME (SPACE) A (SPACE) A
UF (SPACE) TR (SPACE) 18,1 (SPACE) SCHREIBEN (S
PACE4)" <10>
630 SYS49255:PRINT#1,"B-P (SPACE) 2 (SPACE)
0" <60>
640 FORI=0T0255:PRINT#2,CHR*(PEEK (25856+
I));:NEXT <93>
650 PRINT#1,"U2 (SPACE) 2 (SPACE) 0";18,1 <66>
660 : <208>
670 REM ===FILE A LADEN===== <71>
680 SYSM,14,3:PRINT"FILE (SPACE) A (SPACE) L
ESEN (SPACE20)" <134>
690 LA=1:LOAD"A",8,1 <197>
700 SYSM,14,3:PRINT"DIRECTORY (SPACE) AUF (
SPACE) DISK (SPACE) RETTEN (SPACE7)" <76>
710 X=1:GOSUB790 <25>
720 PRINT#1,"V" <82>
730 SYSM,14,3:PRINT"VALIDATE (SPACE) WIRD (
SPACE) DURCHGEFUEHRT (SPACE5)" <227>
740 CLOSE2:CLOSE1 <13>
750 SYSM,14,3:PRINT"DIRECTORY (SPACE) IST (
SPACE) NEU (SPACE) AUFGEBAUT (SPACE5)" <194>
760 END <122>
770 : <63>
780 REM ===LESEN/SCHREIBEN===== <66>
790 POKE250,0:POKE251,100:S=0 <224>
800 IFX=0THENPRINT#1,"U1 (SPACE) 2 (SPACE) 0
"18,S:PRINT#1,"B-P (SPACE) 2 (SPACE) 0":SYS4
9152 <168>
810 IFX=1THENPRINT#1,"B-P (SPACE) 2 (SPACE)
0":SYS49306:PRINT#1,"U2 (SPACE) 2 (SPACE) 0"
18,S <26>
820 S=PEEK (829):IFPEEK (828) <> 0THENI=I+1:
GOTO800 <18>
830 RETURN <207>
840 : <133>
850 REM ===BILDSCHIRM-MASKE===== <95>
860 SYSM,Y,0:PRINT" (SPACE2) _____
" <195>
870 PRINT" (SPACE2) | (SPACE33) | " <50>
880 PRINT" (SPACE2) _____
" <223>
890 RETURN <11>
900 : <193>
910 REM ===NEUER NAMEN===== <41>
920 SYSM,14,3:PRINT" (SPACE16)":SYSM,14,3
:INPUTN$:N$=LEFT*(N$,4) <239>
930 J=49200 <61>
940 FORI=1T04:N (I)=ASC (MID*(N$,I,1)):POK
EJ,N (I):J=J+5:NEXT <195>
950 GOTO250 <5>
960 : <253>
970 REM ===FILE EROEFFNEN===== <64>
980 OPEN1,8,15:OPEN2,8,2,"#" <243>
990 RETURN <111>
1000 : <37>
1010 REM ===MASCH.-ROUTINEN===== <169>
1020 FORJ=0T0204:READA:POKE49152+J,A:FF=
FF+A:NEXT <120>
1030 IFF<<>25537THENPRINT" (CLEAR) FEHLER (
SPACE) IN (SPACE) DATAS (SPACE) !!!":END <177>
1040 RETURN <162>
1050 DATA160,0,169,8,32,9,237,169,98,32,
199,237,32,19,238,145,250,200 <87>
1060 DATA208,248,177,250,141,60,3,160,1,
177,250,141,61,3,160,0,230,251 <230>
1070 DATA96,169,64,141,60,3,169,58,141,6
1,3,169,94,141,62,3,169,68,141,63 <36>
1080 DATA3,169,82,141,64,3,169,94,141,65
,3,169,6,133,183,169,60,133,187 <58>
1090 DATA169,3,133,188,169,0,133,168,162
,0,169,100,133,169,169,8,133 <115>
1100 DATA184,133,185,133,186,169,168,32,
216,255,96,169,0,160,0,153,0,101 <186>
1110 DATA200,208,250,169,130,141,2,101,1
73,180,100,141,3,101,173,181,100 <3>
1120 DATA141,4,101,169,65,141,5,101,169,
160,160,0,153,6,101,200,192,15 <215>
1130 DATA208,248,173,182,100,141,30,101,
96,160,0,169,8,32,12,237,169,98 <175>
1140 DATA32,185,237,177,250,32,221,237,2
00,208,248,177,250,141,60,3,160,1 <95>
1150 DATA177,250,141,61,3,230,251,96,32,
253,174,32,235,183,134,211,165 <47>
1160 DATA20,133,214,32,16,229,96 <126>
ENDE DES LISTINGS

```





# Panic-Car

## Schützt die Fußgänger!

Zu Beginn dauert es etwa fünf Sekunden, bis das Titelbild erscheint. Nach drücken der Feuer-taste erscheint die HALL OF FAME, wonach das Spielfeld aufgebaut wird. Das Programm wartet

nun auf einen Tastendruck, bevor es nun richtig losgeht.

Bei Panicar muß man einen Park durchfahren und dabei alle Felder passieren, die durch ein blinken

des Karo gekennzeichnet sind. Man muß immer auf die Straße bleiben und darf auch keinen der Fußgänger rammen, was zum Verlust eines Lebens führt. Von Level zu Level-es gibt drei-steigt die Zahl der Fußgänger, sowie ihre Geschwindigkeit. Nach jeder Runde bekommt man ein extra

Leben und einen Bonus. Hat man alle Leben verloren, bzw. die drei Runden geschafft, ist das Spiel zu Ende. Je nach Punktzahl kommt man in die TOP-TEN - ...oder auch nicht.

Wichtig: Panicar ist NUR mit Joystick zu spielen.

```

1000 ' -----
1010 ' PANICAR (c)
1986 by SUPRA-SO
FT
1020 ' -----

1030 '
1040 MEMORY &9FFF:CALL &BR4B:POKE &BD
EE,&C9:CALL &BC02:INK 15,13:INK 14,4,
8
1050 DEFINT a-z:DEF FNsee(x,y)=TEST(x
*32-16,408-y*16)
1060 OPENIN"topten2.fil":FOR I=1 TO 1
0:INPUT#9,j*(1):INPUT#9,k(1):NEXT:CL
SEIN
1070 DIM f(41,26),v(5),w(5),v1(5),w1(
5),ad(5),ri(5),n(7)
1080 INK 1,1:INK 13,1:MODE 0:PAPER 0:
CLS:PEN 13:LOCATE 1,25:PRINT"PANICAR"
1090 FOR I=0 TO 412 STEP 4:FOR m=2 TO
16 STEP 2:IF TEST(1,m)<>13 THEN 1110
1100 PLOT 2*1+100,2*m+32,1:PLOTR 4,0,
1:PLOTR 0,-2,1:PLOTR -4,0,1
1110 NEXT m,1:FOR I=7 TO 1 STEP-1:SO
ND 1,0,25,1,,,3:NEXT:INK 1,24
1120 FOR I=1 TO 21:LOCATE 20,25:PRINT
" ":NEXT:LOCATE 1,4:PRINT SPACE$(18):
1130 FOR I=1 TO 999:NEXT:INK 13,22:RE
STORE 1170:FOR ze=1 TO 8:READ pe,xe,y
e,ref$
1140 PEN pe:FOR te=1 TO LEN(ref$):LOCA
TE xe+te,ye:PRINT MID$(ref$,te,1);
1150 SOUND 1,50-2*te,3,6:FOR I=1 TO 8
0:NEXT I,te,ze:WHILE JOY(0)<>16:WEND
1160 BORDER 0:INK 0,0:FOR I=0 TO 400
STEP 2:MOVE 0,1:DRAWR 640,0,0:NEXT
1170 DATA 3,8,4,"$ by",3,5,6,"SUPRA-S
OFT"
1180 DATA 12,1,10,"You've to pass all
",12,2,12,"gleaming fields,"
1190 DATA 12,2,14,"but: Be careful-",
12,1,16,"you mustn't crash!"
1200 DATA 2,5,20,"Good luck!",2,0,22,
"Press FIRE for START"
1210 RANDOMIZE TIME:PAPER 5:CLS:INK 0
,1:GOSUB 2400:GOSUB 2630
1220 SYMBOL 255,60,126,255,255,126:SY
MBOL 254,14,8,8,8,8,8,8
1230 SYMBOL 253,0,0,126,66,126,254,19
8,132:SYMBOL 250,0,16,56,124,56,16
1240 RESTORE 1260
1250 FOR n=0 TO 48:READ I$:POKE &A000
+n,VAL("&"+I$):NEXT
1260 DATA 3a,a0,a0,fe,1,c2,9,a0,c9,2a
,a3,a0,cd,75,bb,3e,fa,cd,5a,bb,c9,21,
a,a0
1270 DATA 7e,3c,3c,77,3e,1,32,a0,a0,c
d,9,a0,c9,1,0,40,11,0,40,21,0,c0,ed,b
0,c9
1280 RESTORE 1300
1290 FOR n=0 TO 36:READ I$:POKE &A100
+n,VAL("&"+I$):NEXT
1300 DATA cd,78,bb,25,2d,cd,1a,bc,16,
B,1,30,a1,a,77,3,1e,3,23,a,77,3,1d,c2
,12
1310 DATA a1,c5,1,fd,7,9,c1,15,c2,d,a
1,c9
1320 RESTORE 1330:FOR n=0 TO 63:READ
I$:POKE &A130+n,I$:NEXT
1330 DATA 255,0,0,255,238,0,0,221,238
,16,32,221,170,16,32,85,170,16,32,85,
238
1340 DATA 16,32,221,238,0,0,221,255,0
,0,255
1350 DATA 255,252,253,255,255,252,253
,255,255,174,255,93,174,12,12,255,93,
12,93
1360 DATA 255,255,221,221,255,255,221
,221,255,250,221,216,255
1370 '
1380 ' S C R E E N
1390 '
1400 FOR I=0 TO 352 STEP 2:MOVE 0,1:D
RAW 640,1,15:NEXT:FOR I=1 TO 3:LOCATE
20,1
1410 PAPER 5:PRINT CHR$(17);:NEXT
1420 MOVE 0,360:DRAWR 640,0,3:PEN 3:L
OCATE 1,1:PRINT"LIVES TIME SCORE"
1430 ' F E L D
1440 PAPER 15:RESTORE 1560
1450 FOR m=4 TO 25:FOR I=1 TO 20:READ
z:ON z GOSUB 1470,1500,1510,1520,154
0
1460 NEXT I,m:PAPER 15:CALL &A025:GOS
UB 2120:GOSUB 2140:GOSUB 2150:GOTO 17
80

```

# programme

```
1470 f(1,m)=1:PAPER 9:LOCATE 1,m:PRINT CHR$(32);:IF RND<0.6 THEN RETURN
1480 LOCATE 1,m:PEN 5:PRINT CHR$(148)
CHR$(22)CHR$(1);
1490 LOCATE 1,m:PEN 12:PRINT CHR$(255)
CHR$(22)CHR$(0);:RETURN
1500 f(1,m)=2:RETURN
1510 PEN 6:LOCATE 1,m:PRINT CHR$(253)
:f(1,m)=1:RETURN
1520 PEN 8:LOCATE 1,m:PRINT CHR$(254)
:PLOT 1*32-8,(25-m)*16+12,4:PLOTR -4,
0
1530 f(1,m)=1:RETURN
1540 f(1,m)=5:RETURN
1550 ' F E L D - D A T A S
1560 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1570 DATA 1,4,5,0,1,4,0,1,4,0,0,1,4,0
,1,4,0,0,1,1
1580 DATA 1,1,0,0,1,3,0,1,1,0,0,1,1,0
,1,1,0,0,3,1
1590 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1600 DATA 0,0,2,0,0,0,2,0,0,2,0,0,2
,0,0,0,2,0,0
1610 DATA 0,0,5,2,0,0,0,0,0,5,2,0,0,0
,0,0,2,5,0,0
1620 DATA 3,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1630 DATA 1,4,0,0,1,4,0,1,4,0,0,1,4,0
,1,4,0,0,1,1
1640 DATA 1,1,0,0,1,1,0,1,1,0,0,3,1,0
,1,1,0,0,1,1
1650 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1660 DATA 0,0,2,0,0,0,2,0,0,0,2,0,0,2
,0,0,0,2,0,0
1670 DATA 0,0,0,2,0,0,2,0,0,2,0,0,0,0
,0,0,2,0,0,0
1680 DATA 1,1,0,0,1,1,0,3,1,0,0,1,1,0
,1,1,0,0,1,1
1690 DATA 1,4,0,0,1,4,0,1,4,0,0,1,4,0
,1,4,0,0,1,1
1700 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1710 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1720 DATA 0,0,5,2,0,0,0,0,0,2,0,0,2
,0,0,2,5,0,0
1730 DATA 0,0,2,0,0,0,2,0,0,2,0,0,0,0
,0,0,0,2,0,0
1740 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,3,1
1750 DATA 1,4,0,0,1,4,0,1,4,0,0,1,4,0
,1,4,0,0,1,1
1760 DATA 1,3,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,3,1
1770 DATA 1,1,0,0,1,1,0,1,1,0,0,1,1,0
,1,1,0,0,1,1
1780 ' F I G U R E N
1790 POKE &A10B,&30:LOCATE x,y:CALL &
A100
1800 POKE &A10B,&50:FOR q=1 TO arti:LO
CATE v(q),w(q):CALL &A100:NEXT
1810 POKE &A02A,&C0:POKE &A02D,&40:PO
KE &A00A,&A3:GOSUB 2460
1820 CALL &A009:WHILE INKEY<>"":WEND
:CALL &BB17
1830 EVERY uhr,0 GOSUB 2020:EVERY 9,1
GOSUB 1900:EVERY 10,2 GOSUB 2170
1840
1850 ' H A U P T P R O G R A M M
1860
1870 DI:IF u=1 THEN 2200 ELSE IF u1=1
THEN 2280
1880 IF u2=1 THEN 2690 ELSE EI:GOTO 1
870
1890 ' R O U T I N E 1
1900 x1=x:y1=y:IF u+u1+u2>0 THEN RETU
RN
1910 IF INKEY(74)>-1 THEN xr=-1:yr=0
ELSE IF INKEY(75)>-1 THEN xr=1:yr=0
1920 IF INKEY(72)>-1 THEN yr=-1:xr=0
ELSE IF INKEY(73)>-1 THEN yr=1:xr=0
1930 IF INKEY(79)>-1 THEN u2=1
1940 DI:x=x+xr:y=y+yr
```

```
1950 IF x>20 THEN x=1 ELSE IF x<1 THE
N x=20
1960 IF y<4 THEN y=25 ELSE IF y>24 TH
EN y=4
1970 EI:IF f(x,y)=1 THEN x=x1:y=y1:GO
SUB 2130
1980 IF FNsee(x,y)=14 THEN s=s+50:GOS
UB 2120:z=z+1:IF z=20 THEN DI:u=1:xr=
0:yr=0:GOTO 1990 ELSE CALL &A015
1990 LOCATE x1,y1:PRINT" ";:IF FNsee(
x,y)<14 THEN GOSUB 2130
2000 LOCATE x,y:POKE &A10B,&30:CALL &
A100:EI:RETURN
2010 ' R O U T I N E 2
2020 IF u+u1+u2>0 THEN RETURN ELSE DI
:GOSUB 2150
2030 EI:FOR q=1 TO art:v1(q)=v(q):w1(
q)=w(q)
2040 IF f(v(q),w(q))<>2 THEN 2060 EL
E IF RND>0.75 THEN 2060
2050 ri(q)=3-ri(q):IF RND>0.5 THEN ad
(q)=-ad(q)
2060 IF ri(q)=1 THEN v(q)=v(q)+ad(q):
GOTO 2070 ELSE w(q)=w(q)+ad(q)
2070 IF v(q)=21 THEN v(q)=1 ELSE IF v
(q)=0 THEN v(q)=20
2080 IF w(q)=3 THEN w(q)=25 ELSE IF w
(q)=25 THEN w(q)=4
2090 DI:LOCATE v1(q),w1(q):PRINT" ":L
OCATE v(q),w(q)
2100 POKE &A10B,&50:CALL &A100:CALL &
A009:EI:NEXT:RETURN
2110 ' A N Z E I G E
2120 PAPER 5:PEN 2:LOCATE 16,2:PRINT
USING"#####";s:PAPER 15:PEN 14:RETUR
N
2130 DI:SOUND 2,1555,30,6,,7:li=li-1
:POKE &A06A,0:IF li=0 THEN u1=1:GOSUB
2190
2140 PAPER 5:PEN 2:LOCATE 1,2:PRINT U
SING"#####";li:PAPER 15:PEN 14:RETUR
N
2150 PAPER 5:PEN 2:LOCATE 8,2:PRINT U
SING"#####";ti5:PAPER 15:PEN 14:RETU
RN
2160 ' S O U N D
2170 IF u+u1+u2>0 THEN RETURN ELSE SO
UND 1,(RND*5+1),15,2:ti=ti+1:RETURN
2180 ' W E C K E R R E S E T
2190 FOR l=0 TO 2:r=REMAIN(1):NEXT:RE
TURN
2200 ' N E U E R L E V E L
2210 GOSUB 2190:LOCATE x,y:POKE &A10B
,&30:CALL &A100
2220 IF s<1000 THEN s=s+50:GOSUB 2120
2230 FOR l=1 TO 7000:NEXT:SOUND 1,239
,40,5:SOUND 1,213,20,5:SOUND 1,190,50
,5
2240 SOUND 1,213,20,5:uhr=uhr-10:s=s+
50:art=RND*(128):li=li+1:art=art+1
2250 CALL &A025:GOSUB 2120:GOSUB 2140
:GOSUB 2150:GOSUB 2410:GOSUB 2460
2260 IF art=6 THEN s=s+900-5*t:GOTO 2
280 ELSE FOR l=1 TO 5000:NEXT:GOTO 17
80
2270 ' L I V E S = 0
2280 GOSUB 2190:FOR l=1 TO 7000:NEXT:
FOR l=0 TO 400 STEP 2:MOVE 0,1
2290 DRAW 640,1,5:NEXT:IF s>k(10) THE
N GOSUB 2500
2300 PAPER 5:PEN 2:CLS:FOR l=1 TO 300
0:NEXT:INK 2,0:INK 3,0:LOCATE 6,12
2310 PRINT"GAME OVER":PEN 3:LOCATE 6
,14:PRINT"RUN? (Y/N)":INK 2,20:INK 3,
6
2320 IF INKEY(43)>-1 THEN 2370 ELSE I
F INKEY(46)>-1 THEN 2330 ELSE 2320
2330 ' E N D E
2340 rr$="topten2.fil":!ERA,&rr$:DPEN
OUT"topten2.fil":FOR l=1 TO 10
2350 PRINT#9,j$(l):PRINT#9,k(1):NEXT:
CLOSEOUT:FOR l=1 TO 3000:NEXT:|BASIC
2360 ' R E S T A R T
2370 CLS:FOR l=1 TO 3000:NEXT:GOSUB 2
400
2380 CALL &A025:GOSUB 2120:GOSUB 2140
```

```
:GOSUB 2150:GOTO 1780
2390 ' I N I T I A L I S I E R U N G
2400 uhr=30:art=3:s=0:hs=0:ti=0:li=1
2410 x=3:y=5:yr=0:xr=0:sym=254:by=&A0
A0:sp=0:u=0:u1=0:u2=0:z=0
2420 RESTORE 2430:FOR l=1 TO 5:READ v
(1),w(1),ad(1),ri(1):NEXT
2430 DATA 3,9,-1,2,10,9,1,2,18,9,1,1,
3,20,1,2,18,20,-1,1
2440 RESTORE 2450:FOR l=1 TO 6:READ n
(1):NEXT:RETURN
2450 DATA 478,426,379,258,319,284,253
2460 xp=RND(TIME)*19+1:yp=RND(TIME)*1
6+4:IF f(xp,yp)<>0 THEN 2480
2470 POKE &A0A3+sp,yp:POKE &A0A4+sp,x
p:sp=sp+2
2480 IF sp<41 THEN 2460 ELSE PEN 14:P
OKE by,0:RETURN
2490 ' I N D E N T O P - T E N
2500 FOR l=1 TO 3000:NEXT:PAPER 5:CLS
:WHILE INKEY<>"":WEND
2510 INK 0,0:INK 3,0:INK 4,0:INK 12,0
:PEN 3:LOCATE 6,9:PRINT"EXCELLENT!"
2520 PRINT" -----":PRINT:PEN
4:PRINT" You're in the HALL"
2530 PEN 0:PRINT:PRINT" ENTER NAM
E"
2540 PEN 12:LOCATE 5,17:PRINT"-----
-----":INK 0,1:INK 3,6:INK 4,26:INK 1
2,18
2550 LOCATE 5,16:INPUT nam$
2560 IF LEN(nam$)>10 THEN LOCATE 4,16
:PRINT SPACE$(16);:GOTO 2540
2570 ' S O R T I E R E N
2580 FOR k1=1 TO 9:IF k(k1)>s THEN 26
10
2590 FOR k2=10 TO k1+1 STEP-1:k(k2)=k
(k2-1):j$(k2)=j$(k2-1):NEXT k2
2600 k(k1)=s:j$(k1)=nam$:GOTO 2620
2610 NEXT k1:k(10)=s:j$(10)=nam$
2620 ' H A L L O F F A M E
2630 PAPER 5:PEN 2:CLS:FOR l=1 TO 300
0:NEXT:INK 2,0:INK 3,0:INK 12,0
2640 LOCATE 3,1:PRINT"THE HALL OF FAM
E":PEN 12:FOR l=1 TO 10
2650 LOCATE 1,2+1*2:PRINT j$(l); " " :S
TRING$(14-LEN(j$(l)),"-");:LOCATE 16,
2+1*2
2660 PRINT USING"#####";k(1):NEXT:WH
ILE INKEY<>"":WEND:PEN 3:LOCATE 4,25
2670 PRINT"Press any key;":INK 2,20:I
NK 3,6:INK 12,18:CALL &BB1B:CLS:RETUR
N
2680 ' B R E A K
2690 FOR l=0 TO 3:r=REMAIN(1):NEXT:CA
LL &BCA7:FOR l=0 TO 400 STEP 2
2700 MOVE 0,1:DRAWR 640,0,5:NEXT:GOTO
2300
```



CHECK V1:PRUEFSUMMEN;

ZEILENNR.:	SUMMEN	1380-	1390:	4770	
1000-	1010:	69907	1400-	1410:	8188
1020-	1030:	59659	1420-	1430:	4341
1040-	1050:	108689	1440-	1450:	66101
1060-	1070:	66504	1460-	1470:	82964
1080-	1090:	56851	1480-	1490:	101767
1100-	1110:	32366	1500-	1510:	22958
1120-	1130:	94664	1520-	1530:	27889
1140-	1150:	66743	1540-	1550:	25336
1160-	1170:	39135	1560-	1570:	80158
1180-	1190:	120315	1580-	1590:	79934
1200-	1210:	44418	1600-	1610:	79981
1220-	1230:	48074	1620-	1630:	80129
1240-	1250:	19093	1640-	1650:	79882
1260-	1270:	51291	1660-	1670:	79686
1280-	1290:	18104	1680-	1690:	80157
1300-	1310:	43160	1700-	1710:	79832
1320-	1330:	61434	1720-	1730:	79876
1340-	1350:	92025	1740-	1750:	80201
1360-	1370:	5304	1760-	1770:	79920

# Highscoresteller

```

10 HIGHSCOREESTELLER
20 FOR l=1 TO 10:n$(l)="SUPRA-SOFT":s(l)
:=0:NEXT
30 OPENOUT"topen2.fil":FOR l=1 TO 10:P
RINT#9,n$(l):PRINT#9,s(l):NEXT:CLOSEOUT

CHECK V1:PRUEFSUMMEN:

ZEILENNR.: SUMMEN
10- 20: 69380
30- : 64217

GESAMTCHECKSUMME: 3844
    
```

1780- 1790: 27923	2120- 2130: 82473	2440- 2470: 84156
1800- 1810: 86198	2140- 2150: 116331	2480- 2490: 60187
1820- 1830: 61806	2160- 2170: 51720	2500- 2510: 111499
1840- 1850: 16732	2180- 2190: 51745	2520- 2530: 24356
1860- 1870: 49021	2200- 2210: 42238	2540- 2550: 67411
1880- 1890: 29427	2220- 2230: 50571	2560- 2570: 63121
1900- 1910: 79413	2240- 2250: 58507	2580- 2590: 102195
1920- 1930: 67022	2260- 2270: 43954	2600- 2610: 107509
1940- 1950: 116770	2280- 2290: 68648	2620- 2630: 63881
1960- 1970: 93312	2300- 2310: 116588	2640- 2650: 98977
1980- 1990: 72641	2320- 2330: 65141	2660- 2670: 4150
2000- 2010: 40454	2340- 2350: 75629	2680- 2690: 45224
2020- 2030: 64939	2360- 2370: 26361	2700- : 22130
2040- 2050: 70963	2380- 2390: 46365	
2060- 2070: 55042	2400- 2410: 83937	
2080- 2090: 44262	2420- 2430: 73205	
2100- 2110: 33224	2440- 2450: 67157	

GESAMTCHECKSUMME: 41313



# Scurfy Joe ist unterwegs!

## Bomben Stimmung

Wahrlich eine Bombenstimmung bei Scurfy Joe. Scurfy Joe, ein entflohener Häftling, wird von Polizisten gejagt. Doch der Böse Bube hat auf einer Brücke Bomben gelegt und schießt wild um sich. Selber ist man auf der Seite der Rechtsvertreter. Sollte die augenblickliche Lage zu brenzlich werden können sich die Beamten an einem langen Baukran in Sicherheit bringen. Aber aufgepaßt: der Haken ist tückisch. Falsches steuern der Spielfigur führt zur Explosion oder zum Absturz. Die X-Position nach der letzten Bombe führt in das zweite Bild. Hier hat Scurfy Joe einen Bombenanschlag vorbereitet. Nun gilt es den Auslöser vor Scurfy Joe zu erreichen (Time). Der Weg führt über die Dächer der Hochhäuser. Die Verbindungsstellen zwischen den Häusern müssen zeitlich genau angesprungen und verlassen werden. Ziel ist die X-Position genau vor dem Auslöser.



```

10 *****
20 *
*
30 * SCURFY JOE
*
40 *
*
50 * (c) 1986 by Klaus Grese
*
60 *
*
70 *****
80 BORDER 1

90 GOSUB 3320:MODE 0:TAG
100 SYMBOL AFTER 32
110 RESTORE 3400:FDR w=-350 TO 150 ST
EP 6
120 GOSUB 3360
130 MOVE w,250
140 PRINT" SCURFY JOE";
150 NEXT
160 FOR w=7 TO 0 STEP -1:SOUND 1,256,
20,w,0,1:NEXT
170 FOR w=600 TO 500 STEP -2:MOVE w,2
50
180 PRINT CHR$(242);" ";NEXT
190 FOR w=149 TO 120 STEP -3:MOVE w,2

50
200 PRINT " SCURFY JOE";CHR$(242);" "
:NEXT
210 FOR w=475 TO 610 STEP 2:MOVE w,25
0
220 PRINT" ";CHR$(243);:NEXT
230 ORIGIN 100,195
240 DRAWR 420,0:DRAWR 0,100:DRAWR -42
0,0:DRAWR 0,-100
250 TAGOFF
260 FOR a=1 TO 25
270 INK 3,a
280 LOCATE 2,21:PEN 3:PRINT"(c) by Kl
aus Grese"
    
```

# programme

```
290 FOR j=1 TO 200:NEXT j,0
300 FOR a=1 TO 75
310 ee=RND*26:BORDER ee:FOR ff=1 TO 3
: NEXT ff,a
320 ENT 1,110,-30,1
330 n=7
340 FOR a=10 TO 1000 STEP 10
350 IF a>860 THEN 370
360 GOTO 380
370 n=n-0.5
380 SOUND 1,a,5,n,0,1:SOUND 2,a+5,5,n
,0,1:SOUND 4,a-5,5,n,0,1
390 NEXT a
400 CLEAR
410 MODE 1
420 BORDER 9
430 lev=1
440 li=3
450 sc=0
460 z=250
470 SYMBOL AFTER 225
480 SYMBOL 226,0,0,0,0,0,128,154,255
490 SYMBOL 227,0,0,0,0,0,6,71,233
500 SYMBOL 228,7,1,1,1,1,33,251
510 SYMBOL 229,8,28,22,59,55,55,23,22
520 SYMBOL 230,22,12,0,8,8,16,16,255
530 SYMBOL 231,129,129,131,133,137,14
5,161,193
540 SYMBOL 232,231,231,66,66,66,66,66
,66
550 SYMBOL 233,255,0,0,0,0,0,255
560 SYMBOL 234,255,24,30,1,1,129,129,
126
570 SYMBOL 235,129,66,36,24,24,36,66,
129
580 SYMBOL 236,0,254,231,255,220,127,
103,255
590 SYMBOL 237,0,248,254,198,254,190,
242,254
600 SYMBOL 238,126,153,153,255,102,66
,153,255
610 SYMBOL 239,4,8,24,60,126,255,255,
126
620 SYMBOL 240,255,171,213,171,213,17
1,213,255
630 SYMBOL 241,255,195,165,153,153,16
5,195,255
640 SYMBOL 242,255,131,135,143,147,16
3,195,255
650 SYMBOL 243,254,130,130,130,130,13
0,254,254
660 SYMBOL 244,255,129,129,129,129,12
9,129,255
670 SYMBOL 245,126,24,24,255,255,255,
255,0
680 SYMBOL 246,85,85,85,0,0,0,0,0
690 SYMBOL 247,170,42,42,10,10,2,2,0
700 SYMBOL 248,42,42,42,10,10,2,2,0
710 SYMBOL 249,255,231,255,0,0,0,0,0
720 SYMBOL 250,84,84,84,80,80,64,64,0
730 SYMBOL 251,254,254,254,254,254,25
4,254,0
740 SYMBOL 252,0,32,96,255,96,32,0,0
750 SYMBOL 253,36,36,36,36,36,231,231
,0
760 SYMBOL 254,126,90,126,231,189,195
,255,126
770 RESTORE
780 READ g
790 IF g>=0 THEN h=h+CHR$(g):GOTO 7
80
800 DATA 254,8,10,253,-1
810 READ i
820 IF i>=0 THEN j=j+CHR$(i):GOTO 8
10
830 DATA 128,8,10,128,-1
840 READ p
850 IF p>=0 THEN q=q+CHR$(p):GOTO 8
40
860 DATA 232,8,10,254,8,10,253,-1
870 READ r
880 IF r>=0 THEN s=s+CHR$(r):GOTO 8
70
890 DATA 128,8,10,128,8,10,128,-1
900 aa=CHR$(251)+CHR$(251)+CHR$(251)
910 bb=CHR$(250)+CHR$(249)+CHR$(248)
```

```
920 cc=CHR$(247)+CHR$(143)
930 dd=CHR$(245)+CHR$(249)+CHR$(246)
940 ee=CHR$(251)+CHR$(243)+CHR$(251)
950 ff=CHR$(242)+CHR$(242)+CHR$(242)
960 gg=CHR$(240)+CHR$(244)+CHR$(240)
970 hh=CHR$(236)+CHR$(237)
980 kh=CHR$(226)+CHR$(227)+CHR$(228)
990 IF lev=1 THEN 2070
1000 INK 1,11:INK 2,16:INK 3,25
1010 b=1:c=19:d=34:e=36:f=24:x=3:y=4:
CLS
1020 LOCATE 1,1:PEN 1:PRINT"***OFFICE
R **SCORE **TIME ***"
1030 LOCATE 11,1:PEN 3:PRINT li
1040 LOCATE 21,1:PEN 3:PRINT zz
1050 FOR a=6 TO 25
1060 LOCATE 2,a:PEN 3:PRINT ee$:TAB(3
7) ff$:NEXT
1070 FOR a=15 TO 25
1080 LOCATE 8,a:PRINT aa$:NEXT
1090 FOR a=23 TO 25
1100 LOCATE 14,a:PRINT CHR$(241):TAB(
35) STRING$(2,CHR$(244)):NEXT
1110 LOCATE 34,22:PRINT STRING$(3,CHR
$(143))
1120 FOR a=6 TO 25
1130 LOCATE 17,a:PRINT gg$:NEXT
1140 LOCATE 35,21:PEN 2:PRINT CHR$(24
5)
1150 LOCATE 39,5:PRINT CHR$(239)
1160 LOCATE 37,4:PRINT CHR$(238):LOCA
TE 37,5:PRINT CHR$(253)
1170 z=z-1:LOCATE 33,1:PEN 3:PRINT z
1180 IF z<26 AND z>0 THEN SOUND 1,18,
5,7:SOUND 2,19,5,7
1190 IF z=0 THEN 3190
1200 CALL @BBO3
1210 b=b+1
1220 IF b<20 AND x=6 THEN 1740
1230 IF b<40 AND x=12 THEN 1740
1240 IF b=20 THEN 1260
1250 GOTO 1280
1260 LOCATE 5,15:PEN 1:PRINT bb$
1270 IF x=6 AND y=13 THEN y=13
1280 IF b>28 THEN 1300
1290 GOTO 1320
1300 LOCATE 5,15:PRINT " "
1310 IF x=6 THEN 1740
1320 IF b=40 THEN 1340
1330 GOTO 1360
1340 LOCATE 11,23:PEN 1:PRINT bb$
1350 IF x=12 AND y=21 THEN y=21
1360 IF b>48 THEN 1380
1370 GOTO 1400
1380 LOCATE 11,23:PRINT " "
1390 IF x=12 THEN 1740
1400 IF b=50 THEN b=1
1410 c=c+1
1420 IF c>33 THEN 1460
1430 LOCATE c,22:PEN 2:PRINT CHR$(249
)
1440 IF c=33 THEN d=34
1450 GOTO 1490
1460 d=d-1
1470 LOCATE d,22:PRINT " "
1480 IF d=20 THEN c=19
1490 e=e-1
1500 LOCATE e,5:PEN 2:PRINT CHR$(252)
," "
1510 IF e<2 THEN 1530
1520 GOTO 1540
1530 LOCATE 1,5:PRINT " *e=36
1540 f=f-1
1550 LOCATE 15,6:PRINT " "
1560 IF f<6 THEN 1600
1570 LOCATE 15,f:PRINT cc$
1580 LOCATE 15,f+1:PRINT " "
1590 GOTO 1620
1600 LOCATE 15,f:PRINT " "
1610 IF f<3 THEN f=24
1620 LOCATE x,y:PEN 2:PRINT h$
1630 j=JOY(0):PEN 2
1640 IF j AND 8 OR j AND 4 OR j AND 1
6 THEN 1650 ELSE 1730
1650 LOCATE x,y:PRINT j$
1660 IF x=e AND y=4 THEN 3950
```

```
1670 IF j AND 8 THEN x=x+1
1680 IF j AND 4 THEN x=x-1
1690 IF x=e AND y=4 THEN 3950
1700 IF j AND 16 THEN 1710 ELSE 1730
1710 IF x=e AND y=4 THEN 3950 ELSE 17
20
1720 y=y-1:LOCATE x,y:PRINT h$:LOCATE
x,y:PRINT j$:x=x+2:LOCATE x,y:PRINT
h$:LOCATE x,y:PRINT j$:y=y+1:SOUND 1,
800,5,7:zz=zz+50:LOCATE 21,1:PEN 3:PR
INT zz:PEN 2
1730 LOCATE x,y:PRINT h$
1740 IF e=x AND y=4 THEN 3950
1750 IF y=20 AND x>20 THEN 1760 ELSE
1810
1760 IF c<34 THEN 1770 ELSE 1780
1770 IF x>c THEN 1790 ELSE 1800
1780 IF x>d THEN 1790 ELSE 1800
1790 IF x=34 THEN 2010 ELSE 1940
1800 GOTO 1930
1810 IF f<22 AND y=21 AND x=16 THEN 1
940
1820 IF x=16 AND f-3>y THEN 1940
1830 IF x=16 AND f<24 THEN y=f-2:LOCA
TE x,y:PRINT h$:LOCATE x,f:PRINT CHR$
(143)
1840 IF x=16 AND y>3 THEN 1930
1850 IF x=1 OR x=37 THEN 1940
1860 IF x=2 OR x=3 OR x=4 OR x=17 OR
x=18 OR x=19 THEN 1870 ELSE 1880
1870 IF y=4 THEN 1930 ELSE 1880
1880 IF x=8 OR x=9 OR x=10 THEN 1890
ELSE 1900
1890 IF y=13 THEN 1930 ELSE 1900
1900 IF x=14 OR x=34 OR x=35 OR x=36
THEN 1910 ELSE 1920
1910 IF y=21 THEN 1930 ELSE 1920
1920 IF x=6 OR x=12 OR x=21 THEN y=y+
1:LOCATE x,y-1:PRINT " ":LOCATE x,y:PR
INT h$:ELSE 1940
1930 GOTO 1170
1940 FOR a=y TO 23
1950 LOCATE x,a:PRINT " ":LOCATE x,a+1
:PRINT h$
1960 NEXT
1970 FOR a=30 TO 600 STEP 30
1980 SOUND 1,a,3,7:SOUND 2,a+3,4,7:NE
XT
1990 li=li-1
2000 GOTO 2950
2010 FOR a=1 TO 3
2020 FOR w=1 TO 10
2030 CALL @BD20
2040 NEXT w,a
2050 LOCATE 37,4:PRINT " ":LOCATE 37,5
:PRINT " ":LOCATE 37,6:PRINT " ":LO
CATE 37,7:PEN 3:PRINT kh$
2060 GOTO 3270
2070 CLS
2080 INK 1,24:INK 2,20:INK 3,16
2090 x=3:y=14:k=40:l=30:xx=5:yy=9:m=2
4:n=10:kk=38:l=25:mm=23:nn=14
2100 ' x=3:y=14:k=40:l=30:xx=5:yy=9:m
=20:n=10:kk=38:l=25:mm=15:nn=5
2110 LOCATE 1,1:PEN 2:PRINT"***OFFICE
R **SCORE **TIME ***"
2120 LOCATE 11,1:PEN 1:PRINT li
2130 LOCATE 21,1:PEN 1:PRINT zz:PEN 2
2140 FOR a=1 TO 37 STEP 2
2150 LOCATE a,16:PRINT hh$
2160 LOCATE a+1,17:PRINT hh$:NEXT:LOC
ATE 39,16:PRINT hh$
2170 LOCATE 1,17:PRINT CHR$(237):LOCA
TE 40,17:PRINT CHR$(236)
2180 FOR a=18 TO 25
2190 LOCATE 3,a:PRINT hh$:TAB(15) hh$
:TAB(25) hh$:TAB(37) hh$:NEXT
2200 xm=0:ym=0:FOR yd=0 TO 100
2210 MOVE 180,ym:DRAW xm,ym:yd:NEXT
2220 xm=460:ym=0:FOR yd=0 TO 100
2230 MOVE 640,ym+yd:DRAW 460,ym:NEXT
2240 LOCATE 5,6:PEN 3:PRINT STRING$(2
4,CHR$(95))
2250 LOCATE 1,7:PRINT STRING$(4,CHR$(
244)):TAB(5) STRING$(24,CHR$(235)):TA
B(29) CHR$(211)
```

```

2260 LOCATE 4,8:PRINT CHR$(209);TAB(5
) STRING$(24,CHR$(233));TAB(29) CHR$(
211)
2270 LOCATE 1,15:PRINT CHR$(239)
2280 FOR a=5 TO 29 STEP 2
2290 LOCATE a,15:PRINT CHR$(239):NEXT
2300 z=z-1:LOCATE 33,1:PEN 1:PRINT z
2310 IF z=0 THEN 3190
2320 CALL &BBO3
2330 k=k-1:l=l-1:m=m-1:n=n-1:kk=kk-1:
l1=l1-1:mm=mm-1:nn=nn-1
2340 IF k=0 THEN k=39:LOCATE 1,14:PRI
NT" "
2350 IF l=0 THEN l=39:LOCATE 1,14:PRI
NT" "
2360 IF m=0 THEN m=39:LOCATE 1,14:PRI
NT" "
2370 IF n=0 THEN n=39:LOCATE 1,14:PRI
NT" "
2380 LOCATE k,14:PEN 2:PRINT CHR$(252
);" "
2390 LOCATE l,14:PRINT CHR$(252);" "
2400 LOCATE m,14:PRINT CHR$(252);" "
2410 LOCATE n,14:PRINT CHR$(252);" "
2420 IF kk=0 THEN kk=39:LOCATE 1,13:P
RINT" "
2430 IF l1=0 THEN l1=39:LOCATE 1,13:P
RINT" "
2440 IF mm=0 THEN mm=39:LOCATE 1,13:P
RINT" "
2450 IF nn=0 THEN nn=39:LOCATE 1,13:P
RINT" "
2460 LOCATE kk,13:PRINT CHR$(252);" "
2470 LOCATE l1,13:PRINT CHR$(252);" "
2480 LOCATE mm,13:PRINT CHR$(252);" "
2490 LOCATE nn,13:PRINT CHR$(252);" "
2500 LOCATE xx,yy:PRINT" "
2510 IF xx>27 THEN ax=-1
2520 IF xx=5 THEN ax=1
2530 xx=xx+ax
2540 LOCATE xx,yy:PRINT CHR$(234)
2550 LOCATE x,y:PEN 1:PRINT h$
2560 j=JOY(0)
2570 IF y=10 THEN 2640
2580 IF j AND 4 OR j AND 8 THEN 2590
ELSE 2600
2590 IF x=k OR x=1 OR x=m OR x=n THEN
3950 ELSE 2670
2600 IF j AND 16 THEN 2610 ELSE 2640
2610 IF x=k OR x=1 OR x=m OR x=n OR x
=kk OR x=11 OR x=mm OR x=nn THEN 3950
ELSE 2620
2620 LOCATE x,y:PRINT j$:y=y-2:LOCATE
x,y:PRINT h$:LOCATE x,y:PRINT j$:x=x
+2:LOCATE x,y:PRINT h$:LOCATE x,y:PRI
NT j$:y=y+2:LOCATE x,y:PRINT h$:SOUND
1,800,5,7:GOSUB 2860
2630 GOTO 2790
2640 IF j AND 2 THEN 2710
2650 IF j AND 1 THEN 2660 ELSE 2780
2660 IF x=k OR x=1 OR x=m OR x=n OR x
=kk OR x=11 OR x=mm OR x=nn THEN 3950
ELSE 2750
2670 LOCATE x,y:PRINT j$
2680 IF j AND 8 THEN x=x+1
2690 IF j AND 4 THEN x=x-1
2700 GOTO 2780
2710 IF x=kk OR x=11 OR x=mm OR x=nn
THEN 3950 ELSE 2720
2720 LOCATE x,y-1:PRINT s$
2730 IF j AND 2 THEN y=y+4
2740 GOTO 2780
2750 LOCATE x,y-1:PRINT s$
2760 IF j AND 1 THEN y=y-4
2770 LOCATE x,y-1:PRINT q$:GOTO 2790
2780 LOCATE x,y:PRINT h$
2790 IF x<2 THEN LOCATE x,y:PRINT s$:
GOTO 2910
2800 IF y>14 OR y<10 THEN 3760
2810 IF x=5 OR x=7 OR x=9 OR x=11 OR
x=13 OR x=15 OR x=17 OR x=19 OR x=21
OR x=23 OR x=25 OR x=27 OR x=29 THEN
LOCATE x,y:PRINT s$:GOTO 2910
2820 IF y=14 THEN 2830 ELSE 2840
2830 IF x=k OR x=1 OR x=m OR x=n THEN
3950

```

```

2840 IF y=13 THEN 2860 ELSE 2870
2850 GOTO 2870
2860 zz=zz+50:LOCATE 21,1:PRINT zz:IF
x=kk OR x=11 OR x=mm OR x=nn THEN 39
50 ELSE RETURN
2870 IF y=10 AND x=xx THEN 3760
2880 IF y=10 AND x<5 OR y=10 AND x>28
THEN 3760
2890 IF y=14 AND x>29 THEN lev=2:CLS:
LOCATE 17,12:PRINT"READY?":FOR sk=1
TO 500:NEXT:GOSUB 3440:GOTO 1000
2900 GOTO 2300
2910 FOR ab=1 TO 3
2920 FOR ac=1 TO 10
2930 CALL &BD20
2940 NEXT ac,ab:GOSUB 3670:FOR a=1 TO
2000:NEXT:GOTO 1970
2950 CLS
2960 FOR a=6 TO 14
2970 LOCATE 20,a:PRINT"+":NEXT
2980 FOR a=17 TO 23
2990 LOCATE a,8:PRINT"+":NEXT
3000 IF i=0 THEN 3040 ELSE 3010
3010 LOCATE 7,19:PRINT"THE OFFICER IS
OUT OF ORDER"
3020 FOR a=1 TO 1000:NEXT
3030 GOTO 3100
3040 LOCATE 9,19:PRINT"THE SPECIAL TE
AM IS DEAD"
3050 LOCATE 14,21:PRINT"YOUR SCORE:";
zz
3060 ENV 5,1,15,1,60,-1,60
3070 SOUND 1,1000,700,0,5:SOUND 2,110
0,700,0,5
3080 LOCATE 9,23:PRINT"PRESS SPACE FO
R NEW TEAM"
3090 FOR a=1 TO 500:NEXT
3100 FOR a=6 TO 13:FOR i=1 TO 15:NEXT
i
3110 LOCATE 17,a:PRINT" ";CHR$(231)
;" "
3120 LOCATE 17,a+1:PRINT"+++++":NEX
T
3130 LOCATE 17,14:PRINT" ";STRING$(3
,CHR$(143));" "
3140 LOCATE 20,5:PRINT CHR$(230)
3150 LOCATE 20,4:PRINT CHR$(229)
3160 FOR a=1 TO 1000:NEXT
3170 IF i=0 THEN 3650
3180 GOTO 990
3190 LOCATE 16,3:PRINT"TIME OVER!"
3200 ef=5
3210 FOR ed=1 TO 100
3220 IF ed>65 THEN ef=ef-0.1
3230 ea=RND*1000:eb=RND*1000:ec=RND*1
000
3240 SOUND 1,ea,5,ef:SOUND 2,eb,5,ef:
SOUND 4,ec,5,ef:NEXT
3250 LOCATE 9,5:PRINT"PRESS SPACE FOR
NEW GAME"
3260 GOTO 3650
3270 LOCATE 10,3:PRINT"THE SCURFY JOE
IS DEAD"
3280 LOCATE 9,5:PRINT"PRESS SPACE FOR
NEW GAME"
3290 FOR a=8 TO 0 STEP -1:SOUND 1,100
0,10,a,0,0,30:SOUND 2,1000,10,a,0,0,3
0:SOUND 4,700,10,a,0,0,30:NEXT
3300 FOR a=2 TO 0 STEP -1:zz=zz+50:SO
UND 1,60,1,7:LOCATE 21,1:PRINT zz:LOC
ATE 33,1:PRINT a:NEXT
3310 GOTO 3550
3320 ENT 1,0,0,3,1,1,3,1,2,3,1,3,3
3330 ENV 1,0,-1,3,1,-2,3,1,-1,3,1,-2,
3
3340 ENV 2,0,-3,3,1,-4,3,1,-3,3,1,-4,
3
3350 RETURN
3360 READ a:IF a=-1 THEN RESTORE 3400
:GOTO 3360
3370 SOUND 1,a,20,0,1,1
3380 SOUND 2,a+2,20,0,1,1
3390 RETURN
3400 DATA 200,200,265,265,200,200,265
,265
3410 DATA 178,178,235,235,178,178,235
,235

```

```

,235
3420 DATA 210,210,265,265,235,235,210
,210
3430 DATA 200,200,235,235,265,265,235
,210,-1
3440 RESTORE 3450
3450 ENV 1,0,-1,5,0,-4,5
3460 ENT 1,1,-1,20
3470 READ a:IF a=-1 THEN RETURN
3480 READ b
3490 SOUND 1,a,15,0,1,1 : SOUND 2,b,1
5,0,1,1
3500 GOTO 3470
3510 DATA 200,160,4,4,200,160,200,160
3520 DATA 160,135,4,4,160,135,160,135
3530 DATA 180,150,4,4,180,150,180,150
3540 DATA 200,160,-1
3550 RESTORE 3550:ENT 1,1,-1,3,1,0,3,
1,1,3,1,0,3,1,-2,3
3560 ENV 1,1,-1,3,1,-2,3,1,-3,3,1,-1,
3
3570 READ a:IF a=-1 THEN RESTORE 3570
:GOTO 3570:ELSE READ b
3580 a$=INKEY$:IF a$=" " THEN RUN 410
3590 IF a=0 THEN SOUND 1,0,25,8:GOTO
3570
3600 SOUND 1,b,25,4,1:SOUND 2,a,25,5,
1,1
3610 GOTO 3570
3620 DATA 400,99,320,0,263,99,320,99,
400,99,320,99,263,90,320,99,400,78,32
0,99,263,0,320,0,400,0,320,0,263,0,32
0,0
3630 DATA 400,99,320,0,263,99,320,99,
400,99,320,99,263,90,320,78,520,90,42
0,0,355,0,420,0,520,0,420,0,355,0,420
,0
3640 DATA 520,105,420,0,355,105,420,1
05,520,105,420,105,355,100,420,105,52
0,90,420,105,355,0,420,0,520,0,420,0,
355,59,420,0,520,67,420,0,355,73,420,
0,520,89,420,0,355,85,420,0,400,79,32
0,0,263,0,320,0,400,0,320,0,263,0,320
,0,-1
3650 IF i=0 THEN GOTO 3790
3660 a$=INKEY$:IF a$=" " THEN RUN 410
ELSE 3660
3670 FOR a=8 TO 0 STEP -1:SOUND 1,100
0,10,a,0,0,30:SOUND 2,1000,10,a,0,0,3
0:SOUND 4,700,10,a,0,0,30:NEXT:RETURN
3680 CLS
3690 ENV 1,1,-1,3,1,-2,3,1,-3,3,1,-4,
3
3700 ENT 1,1,-1,3,1,-2,3,1,-3,3,1,-2,
3
3710 READ a:IF a=-1 THEN FOR a=4 TO 0
STEP -1:SOUND 2,210,20,a,0,1:SOUND 2
,200,20,a,0,1:SOUND 4,190,20,a,0,1:NE
XT:END
3720 IF a=0 THEN SOUND 1,0,20,0:SOUND
2,0,20,0:SOUND 4,0,20,0:GOTO 3710
3730 SOUND 1,a,25,0,1,1:SOUND 2,a+10,
25,5,1,1:SOUND 4,a-10,25,0,1,1:GOTO 3
710
3740 DATA 200,0,225,200,180,180,200
3750 DATA 225,200,200,180,225,200,-1
3760 FOR a=y TO 24
3770 LOCATE x,a-1:PRINT" ":LOCATE x,a
:PRINT h$
3780 NEXT:GOTO 1970
3790 ENV 1,1,1,1,1,2,1,1,3,1,1,4,1,1,
5,1
3800 RESTORE 3840
3810 READ a:IF a=-1 THEN RESTORE 3840
:READ a
3820 a$=INKEY$:IF a$=" " THEN 3930
3830 SOUND 1,a-2,17,0,1:SOUND 2,6,5,6
,0,0,10:SOUND 2,0,12,0:GOTO 3810
3840 DATA 500,340,250,340,200,340,250
,340
3850 DATA 500,340,250,340,200,340,250
,340
3860 DATA 570,380,280,380,225,380,280
,380
3870 DATA 600,400,300,400,250,400,300
,400

```



# programme

3880 DATA 570,380,280,380,225,380,280,380  
 3890 DATA 600,400,300,400,250,400,300  
 3900 DATA 400,500,340,250,340,200,340

3910 DATA 250,340,670,450,270,450,225  
 3920 DATA 450,270,450,-1  
 3930 FOR b=7 TO 0 STEP -0.5: SOUND 1,a  
 ,b,b:NEXT

3940 RUN 400  
 3950 SOUND 1,25,10,7: SOUND 2,69,10,6:  
 SOUND 4,34,10,6: IF lev=1 THEN 3760 EL  
 SE 1940

CHECK V1: PRUEFSUMMEN:

ZEILENNR.: SUMMEN

10- 20: 30254  
 30- 40: 16797  
 50- 60: 36275  
 70- 80: 28406  
 90- 100: 5658  
 110- 120: 27946  
 130- 140: 11844  
 150- 160: 42642  
 170- 180: 49701  
 190- 200: 67071  
 210- 220: 43838  
 230- 240: 35608  
 250- 260: 6618  
 270- 280: 41764  
 290- 300: 30369  
 310- 320: 15674  
 330- 340: 16608  
 350- 360: 9999  
 370- 380: 54759  
 390- 400: 1699  
 410- 420: 417  
 430- 440: 7382  
 450- 460: 7248  
 470- 480: 20854  
 490- 500: 27540  
 510- 520: 30358  
 530- 540: 53219  
 550- 560: 35401  
 570- 580: 53203  
 590- 600: 71741  
 610- 620: 65790  
 630- 640: 72420  
 650- 660: 68134  
 670- 680: 42902  
 690- 700: 22416  
 710- 720: 33778  
 730- 740: 53798  
 750- 760: 57904

770- 780: 1819  
 790- 800: 51388  
 810- 820: 46787  
 830- 840: 9089  
 850- 860: 63654  
 870- 880: 49502  
 890- 900: 57983  
 910- 920: 60621  
 930- 940: 81553  
 950- 960: 80957  
 970- 980: 61143  
 990- 1000: 17051  
 1010- 1020: 97443  
 1030- 1040: 22156  
 1050- 1060: 46461  
 1070- 1080: 19297  
 1090- 1100: 71265  
 1110- 1120: 24644  
 1130- 1140: 29463  
 1150- 1160: 46678  
 1170- 1180: 73902  
 1190- 1200: 9059  
 1210- 1220: 30694  
 1230- 1240: 34731  
 1250- 1260: 10624  
 1270- 1280: 39863  
 1290- 1300: 3335  
 1310- 1320: 15932  
 1330- 1340: 11868  
 1350- 1360: 43258  
 1370- 1380: 4117  
 1390- 1400: 23403  
 1410- 1420: 16137  
 1430- 1440: 34503  
 1450- 1460: 7980  
 1470- 1480: 20158  
 1490- 1500: 27835  
 1510- 1520: 9715  
 1530- 1540: 17830  
 1550- 1560: 9946  
 1570- 1580: 19091

1590- 1600: 6317  
 1610- 1620: 29994  
 1630- 1640: 9811  
 1650- 1660: 39204  
 1670- 1680: 54021  
 1690- 1700: 46568  
 1710- 1720: 40818  
 1730- 1740: 39265  
 1750- 1760: 51285  
 1770- 1780: 35365  
 1790- 1800: 17433  
 1810- 1820: 84962  
 1830- 1840: 83833  
 1850- 1860: 62791  
 1870- 1880: 67444  
 1890- 1900: 45364  
 1910- 1920: 75391  
 1930- 1940: 10728  
 1950- 1960: 47333  
 1970- 1980: 39403  
 1990- 2000: 10754  
 2010- 2020: 12166  
 2030- 2040: 5744  
 2050- 2060: 17832  
 2070- 2080: 7617  
 2090- 2100: 49243  
 2110- 2120: 64683  
 2130- 2140: 25664  
 2150- 2160: 57662  
 2170- 2180: 44061  
 2190- 2200: 71603  
 2210- 2220: 88725  
 2230- 2240: 60677  
 2250- 2260: 62874  
 2270- 2280: 19620  
 2290- 2300: 44574  
 2310- 2320: 9059  
 2330- 2340: 44221  
 2350- 2360: 54509  
 2370- 2380: 49592  
 2390- 2400: 32600

2410- 2420: 48727  
 2430- 2440: 64974  
 2450- 2460: 50732  
 2470- 2480: 36391  
 2490- 2500: 30072  
 2510- 2520: 41774  
 2530- 2540: 37619  
 2550- 2560: 20560  
 2570- 2580: 38466  
 2590- 2600: 16820  
 2610- 2620: 49093  
 2630- 2640: 10634  
 2650- 2660: 67141  
 2670- 2680: 30818  
 2690- 2700: 28182  
 2710- 2720: 44488  
 2730- 2740: 28250  
 2750- 2760: 42855  
 2770- 2780: 37566  
 2790- 2800: 65094  
 2810- 2820: 42404  
 2830- 2840: 62882  
 2850- 2860: 16667  
 2870- 2880: 48545  
 2890- 2900: 39289  
 2910- 2920: 14701  
 2930- 2940: 66314  
 2950- 2960: 6360  
 2970- 2980: 15848  
 2990- 3000: 26391  
 3010- 3020: 54083  
 3030- 3040: 34924  
 3050- 3060: 37820  
 3070- 3080: 59128  
 3090- 3100: 51028  
 3110- 3120: 40374  
 3130- 3140: 36292  
 3150- 3160: 20704  
 3170- 3180: 9779  
 3190- 3200: 15586  
 3210- 3220: 65700

3230- 3240: 52613  
 3250- 3260: 34142  
 3270- 3280: 64250  
 3290- 3300: 118530  
 3310- 3320: 12402  
 3330- 3340: 59571  
 3350- 3360: 29649  
 3370- 3380: 14221  
 3390- 3400: 27881  
 3410- 3420: 55680  
 3430- 3440: 33358  
 3450- 3460: 12495  
 3470- 3480: 23035  
 3490- 3500: 27395  
 3510- 3520: 42372  
 3530- 3540: 24982  
 3550- 3560: 67269  
 3570- 3580: 88986  
 3590- 3600: 46396  
 3610- 3620: 15890  
 3630- 3640: 61312  
 3650- 3660: 48615  
 3670- 3680: 15295  
 3690- 3700: 59575  
 3710- 3720: 69132  
 3730- 3740: 57644  
 3750- 3760: 29384  
 3770- 3780: 46402  
 3790- 3800: 16968  
 3810- 3820: 56127  
 3830- 3840: 86829  
 3850- 3860: 55207  
 3870- 3880: 54904  
 3890- 3900: 42048  
 3910- 3920: 28087  
 3930- 3940: 50866  
 3950- : 37157

GESAMTCHECKSUMME: 7170

# DRAWGENERATOR

Diese nützliche Zeichenhilfe



bindet das Ergebnis in Ihr Basicprogramm ein!

Wer kennt das nicht: Man will eine Überschrift DRAWen oder ein Bild auf dem Bildschirm entwerfen; aber dafür gibt es nur zwei Möglichkeiten:

1. Man setzt sich stundenlang an den Computer und entwickelt das Programm durch ausprobieren.
2. Man benutzt ein Zeichenprogramm, bei dem man das Bild al-

lerdings nicht als Listing ausdrucken kann.

Beide Arten haben also ihre Nachteile. Mit dem DRAWGENERATOR kann solch ein Bild menügesteuert werden gezeichnet und dann in Basic Zeilen abgespeichert werden!

Zeichenmenü

1. Linie: 1. Punkt setzen; 2. Punkt setzen; Farbe eingeben

2. Balken horiz.: 1. Punkt setzen; 2. Punkt setzen; Farbe eingeben; Dicke eingeben (Dicke=Länge)
3. Balken vert.: wie 2., jedoch bestimmt die Dicke hier die Höhe
4. Vollkreis: Mittelpunkt setzen; Punkt auf Kreislinie setzen; Farbe eingeben

5. Kreislinie: wie 4.
  6. Text: Anfangspunkt setzen; Farbe eingeben; Text eingeben
- Hauptmenü
1. Zurück zum Zeichnen
  2. Bild aufbauen (zum Überprüfen)
  3. Bild SAVEN
  4. Letzten Schritt löschen
- Die Steuerung erfolgt mit den

CURSER-Tasten, wobei die Schrittweite durch SHIFT erhöht werden kann. Der 1. Punkt kann mit DEL gelöscht werden, solange der 2. noch nicht gesetzt ist.

Achtung:

Bei Cassettbetrieb muß in Zeile 1190 hinter OPENOUT ein Ausrufezeichen!

```

10 *****
20 **** Draw-Generator V-1.1 ****
30 *** Geschrieben: Sept. 85 ***
40 **** (c) by P.W. Soft ****
50 ***** Peter Woelflick *****
60 *****
70 DEFINT a-z
80 KEY 1,"MODE 2:LIST"
90 DIM a(200),b(200),bc(200),ab(200),
f(200),h(200),d(200),st(200),text$(20
0)
100 MODE 1
110 BORDER 13
120 LOCATE 15,7:INPUT"Mode=";m
130 IF m<0 OR m>2 THEN 100
140 CLS
150 PEN #2,3
160 WINDOW #0,1,40,1,20
170 WINDOW#2,1,40,21,25
180 PAPER #2,2:CLS#2
190 *****
200 *** Haupt-Schleife ***
210 *****
220 z=z+1
230 CLS #2:GOSUB 1410:INPUT #2,"";mo:
IF mo<0 THEN 1340
240 ON mo GOTO 260,320,410,500,580,64
0
250 *** Linie ***
260 GOSUB 710
270 ORIGIN a,ab:DRAW b-a,bc-ab,f
280 a(z)=a:ab(z)=ab:b(z)=b:bc(z)=bc:f
(z)=f
290 h(z)=mo
300 GOTO 220
310 *** Balken nach oben ***
320 a=0:ab=0:GOSUB 710:INPUT #2,"Dick
e=";d
330 IF d<0 THEN st=-1 ELSE st=1
340 IF m<2 THEN st=st*2:text$(z)="STE
P"+STR$(st)
350 IF m=2 AND st<0 THEN text$(z)="ST
EP"+STR$(st)
360 FOR t=1 TO d STEP st
370 ORIGIN a,ab+t:DRAW b-a,bc-ab,f:NE
XT
380 a(z)=a:ab(z)=ab:b(z)=b:bc(z)=bc:f
(z)=f:d(z)=d:st(z)=st
390 GOTO 220
400 *** Balken nach rechts ***
410 a=0:ab=0:GOSUB 710:INPUT #2,"Dick
e=";d
420 IF d<0 THEN st=-1 ELSE st=1
430 IF m<2 THEN st=st*2:text$(z)="STE
P"+STR$(st)
440 IF m=2 AND st<0 THEN text$(z)="ST
EP"+STR$(st)
450 FOR t=1 TO d STEP st
460 ORIGIN a+t,ab:DRAW b-a,bc-ab,f:NE
XT
470 a(z)=a:ab(z)=ab:b(z)=b:bc(z)=bc:f
(z)=f:d(z)=d:st(z)=st
480 GOTO 220
490 *** Voll-Kreis ***
500 a=0:ab=0:GOSUB 710:ra=SQR((a-b)^2
+(bc-ab)^2):ORIGIN a,ab:DEG
510 IF ra<=60 AND m<2 THEN st=2:text$
(z)="STEP 2" ELSE st=1
520 FOR j=0 TO 180 STEP st
530 PLOT ra*SIN(j),ra*COS(j),f
540 DRAW 2*ra*COS(90+j),0,f
550 NEXT
560 a(z)=a:ab(z)=ab:b(z)=b:bc(z)=bc:f
(z)=f:st(z)=st:GOTO 220

```

```

570 *** Kreis-Linie ***
580 a=0:ab=0:GOSUB 710:ra=SQR((a-b)^2
+(bc-ab)^2):ORIGIN a,ab:DEG
590 IF ra<=60 AND m<2 THEN st=2:text$
(z)="STEP 2" ELSE st=1
600 FOR j=1 TO 180 STEP st:PLOT ra*S
IN(j),ra*COS(j),f
610 PLOT 2*ra*COS(90+j),0,f
620 NEXT:a(z)=a:ab(z)=ab:b(z)=b:bc(z)
=bc:f(z)=f:st(z)=st:GOTO 220
630 *** Text ***
640 flag=1:fa=2:GOSUB 720
650 ORIGIN 0,0:PLOT -10,-10,f:CLS #2:
INPUT #2,"Text=";text$:MOVE b,bc:TAG:
PRINT text$:TAGOFF
660 b(z)=b:bc(z)=bc:f(z)=f:text$(z)=t
ext$
670 GOTO 220
680 *****
690 *** Steuerung ***
700 *****
710 fa=2:flag=0
720 CALL &BB03
730 x=320:y=200:CLS #2
740 *** Zeichenart-Anzeige ***
750 LOCATE #2,30,3:IF mo=2 THEN PRINT
#2,"Dicke="CHR$(1);
" ELSE IF mo=3 T
HEN PRINT #2,"Dicke="CHR$(1); " ELSE
IF mo=4 THEN PRINT #2,CHR$(231) ELSE
IF mo=5 THEN PRINT #2,CHR$(230) ELSE
IF mo=6 THEN PRINT #2,"Text"
760 LOCATE #2,30,2:PRINT #2,"Art=";mo
770 t=1+(TIME/100)
780 IF INKEY(0)=0 THEN y=y+2:wa=1
790 IF INKEY(2)=0 THEN y=y-2:wa=1
800 IF INKEY(8)=0 THEN x=x-2:wa=1
810 IF INKEY(1)=0 THEN x=x+2:wa=1
820 IF INKEY(0)=32 THEN y=y+8:wa=1
830 IF INKEY(2)=32 THEN y=y-8:wa=1
840 IF INKEY(8)=32 THEN x=x-8:wa=1
850 IF INKEY(1)=32 THEN x=x+8:wa=1
860 IF INKEY(79)<>-1 THEN flag=0:LOCA
TE #2,1,1
870 IF INKEY(18)<>-1 THEN IF t<TIME/1
00 THEN IF flag=1 THEN flag=0:b=x:bc=
y:PRINT #2,"
2.Punkt:"b";"bc:ORIGIN 0,0:t1=TEST(x,
y):PLOT x,y,3-t1:INPUT #2,"Farbe=";f:
RETURN ELSE a=x:ab=y:flag=1:PRINT #2,
"1.Punkt:"a";"ab:ORIGIN 0,0:t1=TEST(x
,y):PLOT x,y,3-t1
880 ORIGIN 0,0
890 a1=TEST(x,y):a2=TEST(x+6,y):a3=TE
ST(x+8,y)
900 a4=TEST(x-6,y):a5=TEST(x-8,y):a6=
TEST(x,y+6):a7=TEST(x,y+8):a8=TEST(x,
y-6):a9=TEST(x,y-8)
910 LOCATE #2,30,1:PRINT #2,x,y
920 IF fa=1 THEN IF fa<2 THEN fa=fa+
1 ELSE IF fa>3 THEN fa=fa-1 ELSE fa=f
a+1
930 IF a1<>2 THEN fa=2
940 *** Fadenkreuz plotten ***
950 ORIGIN x,y:PLOT 6,0,fa:PLOT 0,0,f
a
960 PLOT 8,0,fa:PLOT -6,0:PLOT -8,0:P
LOT 0,6
970 PLOT 0,8:PLOT 0,-6:PLOT 0,-8
980 IF wa=1 THEN wa=0:GOTO 1000 ELSE
WHILE INKEY#="" :WEND
990 *** Hintergrund plotten ***
1000 ORIGIN x,y:PLOT 6,0,a2:PLOT 0,0,
a1
1010 PLOT 8,0,a3:PLOT -6,0,a4:PLOT -8
,0,a5:PLOT 0,6,a6
1020 PLOT 0,8,a7:PLOT 0,-6,a8:PLOT 0,
-8,a9
1030 GOTO 780
1040 *** Bild aufbauen ***
1050 zx=z:CLS:CLS #2:FOR z=1 TO zx-1:
LOCATE #2,27,5:PRINT #2,"Schritt:"z;
1060 IF h(z)=1 THEN ORIGIN a(z),ab(z)
:DRAW b(z)-a(z),bc(z)-ab(z),f(z)
1070 IF h(z)=2 THEN FOR a=1 TO d(z) S
TEP st(z):ORIGIN a(z),ab(z)+a:DRAW b(
z)-a(z),bc(z)-ab(z),f(z)NEXT a

```

```

1080 IF h(z)=3 THEN FOR a=1 TO d(z) S
TEP st(z):ORIGIN a(z)+a,ab(z):DRAW b(
z)-a(z),bc(z)-ab(z),f(z):NEXT a
1090 IF h(z)=4 THEN DEG:ra=SQR((a(z)-
b(z))^2+(bc(z)-ab(z))^2):FOR j=1 TO 1
80 STEP st(z):ORIGIN a(z),ab(z):PLOT
ra*SIN(j),ra*COS(j),f(z):DRAW 2*ra*C
OS(90+j),0,f(z):NEXT j
1100 IF h(z)=5 THEN DEG:ra=SQR((a(z)-
b(z))^2+(bc(z)-ab(z))^2):FOR j=1 TO 1
80 STEP st(z):ORIGIN a(z),ab(z):PLOT
ra*SIN(j),ra*COS(j),f(z):PLOT 2*ra*C
OS(90+j),0,f(z):NEXT j
1110 IF h(z)=6 THEN ORIGIN 0,0:PLOT -
10,-10,f(z):MOVE b(z),bc(z):TAG:PRINT
text$(z);TAGOFF
1120 NEXT z
1130 z=zx:GOTO 230
1140 *****
1150 *** Abspeichern ***
1160 *****
1170 zx=z
1180 z=0
1190 OPENOUT"drawbild.bas"
1200 PRINT #9,"10 mode"m
1210 FOR z=1 TO zx
1220 IF h(z)=1 THEN PRINT #9,z*10+10;
"ORIGIN"+STR$(a(z))+","+STR$(ab(z))+
";draw "+STR$(b(z)-a(z))+","+STR$(bc(z)
)-ab(z))+","+STR$(f(z))
1230 IF h(z)=2 THEN PRINT #9,z*10+10;
"for a=1 to"+STR$(d(z));text$(z)+";or
igin"+STR$(a(z))+","+STR$(ab(z))+";a:
draw"+STR$(b(z)-a(z))+","+STR$(bc(z)-
ab(z))+","+STR$(f(z));:next a"
1240 IF h(z)=3 THEN PRINT #9,z*10+10;
"for a=1 to"+STR$(d(z));text$(z)+";or
igin"+STR$(a(z))+"+a,"+STR$(ab(z))+";
draw"+STR$(b(z)-a(z))+","+STR$(bc(z)-
ab(z))+","+STR$(f(z));:next a"
1250 IF h(z)<>4 THEN 1270 ELSE PRINT
#9,z*10+10;"deg:ra=sqr(("+STR$(a(z))+
"+STR$(b(z))+")^2+("+STR$(bc(z))+"+
"+STR$(ab(z))+")^2):FOR j=1 TO 180"+t
ext$(z);:ORIGIN "+STR$(a(z))+","+STR$(
ab(z));
1260 PRINT#9,";PLOT ra*SIN(j),ra*COS(
j),"+STR$(f(z))+";DRAW 2*ra*COS(90+j
),0,"+STR$(f(z))+";:NEXT j"
1270 IF h(z)<>5 THEN 1290 ELSE PRINT
#9,z*10+10;"deg:ra=sqr(("+STR$(a(z))+
"+STR$(b(z))+")^2+("+STR$(bc(z))+"+
"+STR$(ab(z))+")^2):FOR j=1 TO 180"+t
ext$(z);:ORIGIN "+STR$(a(z))+","+STR$(
ab(z));
1280 PRINT#9,";PLOT ra*SIN(j),ra*COS(
j),"+STR$(f(z))+";PLOT 2*ra*COS(90+j
),0,"+STR$(f(z))+";:NEXT j"
1290 IF h(z)=6 THEN PRINT #9,z*10+10;
"ORIGIN 0,0:PLOT -10,-10,"+STR$(f(z))
+";:MOVE"+STR$(b(z))+","+STR$(bc(z))+
";TAG:PRINT"CHR$(34)text$(z)CHR$(34)
";TAGOFF"
1300 NEXT z
1310 CLOSEDOUT
1320 z=zx:GOTO 230
1330 *** Hilfs-Menue ***
1340 WINDOW SWAP 2,0:CLS:PRINT"
Hilfsmenue"
1350 PRINT"1 --> Zurueck zum Zeichnen"
1360 PRINT"2 --> Bild aufbauen"
1370 PRINT"3 --> Saven";PRINT"4 --> L
etzten Schritt loeschen"
1380 WINDOW SWAP 0,2
1390 in=VAL(INKEY#):IF in=0 THEN 1390
ELSE IF in=4 THEN z=z-1:GOTO 230 ELS
E ON in GOTO 230,1050,1170
1400 *** Haupt-Menue ***
1410 WINDOW SWAP 2,0:PRINT"
Zeichen-Arten"
1420 PRINT" 1 --> Linie 4 -->
ausgef. Kreis";
1430 PRINT" 2 --> Balken vert. 5 -->
Kreislinie"
1440 PRINT" 3 --> Balken horiz. 6 -->
Text"

```

# programme

1450 PRINT" ENTER --> zum Hilf  
smenuue"  
1460 LOCATE 25,1:WINDOW SWAP 0,2:RETU  
RN



CHECK V1:PRUEFSUMMEN:

ZEILENNR.: SUMMEN

10- 20: 55976  
30- 40: 58732  
50- 60: 55922  
70- 80: 10381  
90- 100: 20334  
110- 120: 13981  
130- 140: 21135  
150- 160: 3971

170- 180: 5681  
190- 200: 42913  
210- 220: 24438  
230- 240: 81603  
250- 260: 8227  
270- 280: 115647  
290- 300: 13215  
310- 320: 76250  
330- 340: 44312  
350- 360: 71444  
370- 380: 59719

\*\*\*\*\*

390- 400: 30721  
410- 420: 92722  
430- 440: 52980  
450- 460: 46325  
470- 480: 34302  
490- 500: 44969  
510- 520: 40395  
530- 540: 84683  
550- 560: 2844  
570- 580: 46553  
590- 600: 27359  
610- 620: 47747  
630- 640: 28689  
650- 660: 7675  
670- 680: 8630  
690- 700: 22069  
710- 720: 15872  
730- 740: 48687  
750- 760: 75727

770- 780: 55490  
790- 800: 86787  
810- 820: 88684  
830- 840: 90889  
850- 860: 81059  
870- 880: 51096  
890- 900: 36901  
910- 920: 74698  
930- 940: 49689  
950- 960: 67018  
970- 980: 15233  
990- 1000: 69214  
1010- 1020: 52243  
1030- 1040: 20280  
1050- 1060: 26861  
1070- 1080: 3726  
1090- 1100: 113234  
1110- 1120: 14394  
1130- 1140: 21612

1150- 1160: 27087  
1170- 1180: 8040  
1190- 1200: 22037  
1210- 1220: 72327  
1230- 1240: 74754  
1250- 1260: 53427  
1270- 1280: 52770  
1290- 1300: 11495  
1310- 1320: 11788  
1330- 1340: 56589  
1350- 1360: 58512  
1370- 1380: 35694  
1390- 1400: 79761  
1410- 1420: 100426  
1430- 1440: 101517  
1450- 1460: 56575

GESAMTCHECKSUMME: 42423

## Demo

```
10 MODE 1
20 FOR a=1 TO 100 STEP 2:ORIGIN 200+
a, 296:DRAW 128, 56, 2:NEXT a
30 FOR a=1 TO-100 STEP-2:ORIGIN 300+
a, 296:DRAW -168, 56, 2:NEXT a
40 FOR a=1 TO 200 STEP 2:ORIGIN 136+
a, 316:DRAW 0, 36, 2:NEXT a
50 FOR a=1 TO-60 STEP-2:ORIGIN 200,
296+a:DRAW -168, 56, 3:NEXT a
60 FOR a=1 TO-168 STEP-2:ORIGIN 200+
a, 236:DRAW 0, 56, 3:NEXT a
70 FOR a=1 TO 60 STEP 2:ORIGIN 200,
232+a:DRAW -168,-56, 3:NEXT a
80 FOR a=1 TO-60 STEP-2:ORIGIN 300,
296+a:DRAW 128, 56, 3:NEXT a
90 FOR a=1 TO 130 STEP 2:ORIGIN 300+
a, 232:DRAW 0, 64, 3:NEXT a
110 FOR a=1 TO 60 STEP 2:ORIGIN 300,
232+a:DRAW 128,-58, 3:NEXT a
120 FOR a=1 TO 100 STEP 2:ORIGIN 200
+a, 232:DRAW -168,-54, 1:NEXT a
130 FOR a=1 TO-100 STEP-2:ORIGIN 300
+a, 232:DRAW 128,-56, 1:NEXT a
140 FOR a=1 TO 216 STEP 2:ORIGIN 134
+a, 176:DRAW 2, 36, 1:NEXT a
```

```
150 FOR a=1 TO 22 STEP 2:ORIGIN 256,
264+a:DRAW 8, 8, 3:NEXT a
160 FOR a=1 TO 22 STEP 2:ORIGIN 254,
264+a:DRAW -8, 6, 2:NEXT a
170 FOR a=1 TO-18 STEP-2:ORIGIN 256,
262+a:DRAW 8,-8, 2:NEXT a
180 FOR a=1 TO-18 STEP-2:ORIGIN 254,
262+a:DRAW -8,-8, 3:NEXT a
190 FOR a=1 TO 24 STEP 2:ORIGIN 256+
a, 262:DRAW 8,-8, 1:NEXT a
200 FOR a=1 TO 24 STEP 2:ORIGIN 256+
a, 262:DRAW 8, 8, 2:NEXT a
210 FOR a=1 TO-24 STEP-2:ORIGIN 256+
a, 262:DRAW -12, 8, 1:NEXT a
220 FOR a=1 TO-24 STEP-2:ORIGIN 254+
a, 262:DRAW -8,-8, 2:NEXT a
230 ORIGIN 0,0:PLOT -10,-10, 2:MOVE 4
0, 112:TAG:PRINT"P.W. Soft presents:"
;:TAGOFF
240 ORIGIN 0,0:PLOT -10,-10, 3:MOVE 2
72, 88:TAG:PRINT"DRAW-GENERATOR";:TAG
OFF
250 FOR a=1 TO-12 STEP-2:ORIGIN 16+a
, 368:DRAW 0,-368, 1:NEXT a
260 FOR a=1 TO 12 STEP 2:ORIGIN 24,
360+a:DRAW -24, 0, 3:NEXT a
270 ORIGIN 32, 352:DRAW -16, 8, 2
280 ORIGIN 32, 176:DRAW -16,-8, 2
```

CHECK V1:PRUEFSUMMEN:

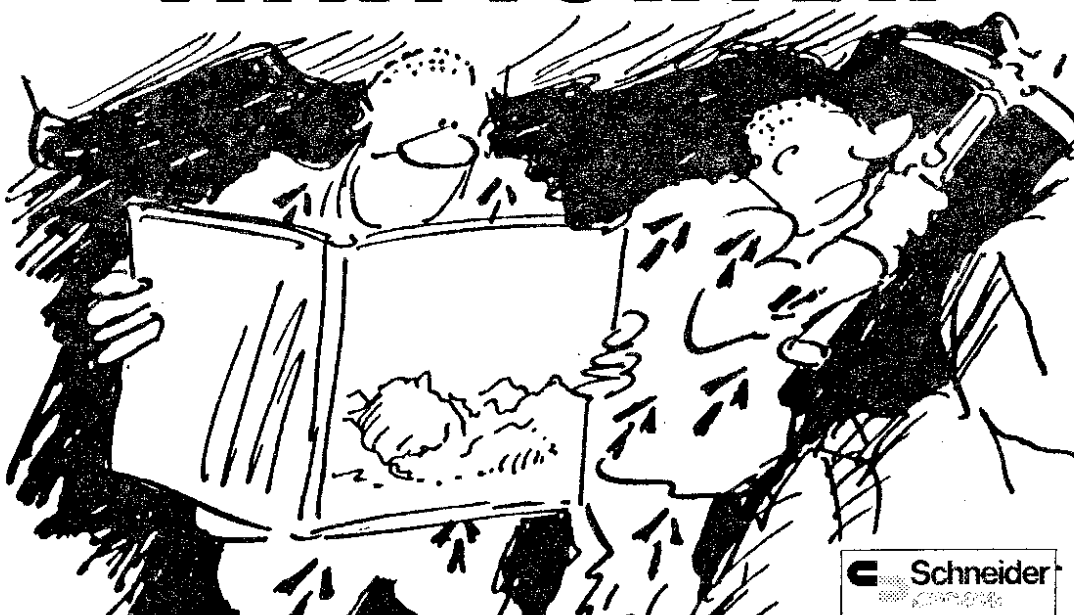
ZEILENNR.: SUMMEN

10- 20: 10360  
30- 40: 29380  
50- 60: 45883  
70- 80: 51500  
90- 110: 31767  
120- 130: 65492  
140- 150: 70620  
160- 170: 26422  
180- 190: 34250  
200- 210: 74807  
220- 230: 78712  
240- 250: 43584  
260- 270: 20118  
280- : 14182

GESAMTCHECKSUMME: 58607



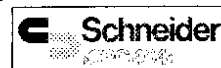
# AIRFIGHTER



Ein Grubenunglück hat 36 Bergarbeiter verschüttet. Die einzige Möglichkeit, die Leute vor dem Erstickten zu retten, liegt in der Befreiung mit Hilfe eines wendigen Hubschraubers, den "AIR-FIGHTER".

Meine Aufgabe ist es jetzt, mich an den Joystick zu klammern und unter Tage zu begeben. Der gefühlvoll zu steuernde Helikopter verbraucht allerdings sehr viel Treibstoff. Doch einmaliges Auftanken pro Level ist möglich. Vorsicht: Nur nicht vergessen den Helikopter zu steuern, sonst sinkt er nach einer gewissen Zeit ab. Also los, retten was das Zeug hält. Um am Ende belohnt zu werden, müssen alle Männer gerettet werden.

Wenn ich zu Beginn einfach ein wenig herumballere, kann ich auch ein Level überspringen. Bonuspunkte und Siegerzeremonie gehen aber flöten.



```

100 *****
110 *
120 * AIR FIGHTER *
130 *
140 * (c) 1986 by MKP *
150 *
160 * USW Product *
170 *
180 * MUSIK by *
190 *
200 * O. SCHMIDT *
210 *
220 *****
230
240 *****
250 Gosubroutinen
260 *****
270 CLS:MODE 1
280 GOSUB 350
290 GOSUB 420
300 GOTO 2940
310 GOSUB 3290
320 GOSUB 1700
330 GOSUB 930
340 GOTO 930
350 *****
360 Variablen
370 *****
380 DIM manx(10),many(10),bild$(31)
390 sc=0:lev=1:hsc=1000:fue=637
400 x=65:y=50:x1=0:y1=0:xstep=0:ystep=0:
hub$(1)="ab":hub$(2)="kj":ou=0
410 RETURN
420 *****
430 Zeichensatz aendern
440 *****
450 SYMBOL AFTER 32
460 SYMBOL 99,0,126,102,96,96,102,126
470 SYMBOL 101,0,126,102,112,112,102,126
480 SYMBOL 102,0,126,102,112,112,96,96
490 SYMBOL 104,0,102,102,126,126,102,102
500 SYMBOL 105,0,126,24,24,24,24,126
510 SYMBOL 108,0,96,96,96,96,102,126
520 SYMBOL 111,0,126,102,102,102,102,126
530 SYMBOL 114,0,126,102,102,126,108,102
540 SYMBOL 115,0,126,96,96,126,6,126
550 SYMBOL 103,0,126,102,96,110,102,126
560 SYMBOL 117,0,102,102,102,102,60,24
570 SYMBOL 118,0,102,102,102,102,60,24
580 SYMBOL 120,255,255,195,195,195,195,255,255
590 SYMBOL 121,93,107,54,28,8,28,20,54
600 SYMBOL 123,0,31,63,113,117,113,63,31
:SYMBOL 125,0,248,252,94,94,78,252,248
610 SYMBOL 97,127,1,7,8,8,7,2,6:SYMBOL 9
8,255,192,240,136,136,240,32,48
620 SYMBOL 91,64,2,21,7,20,41,2,36
630 RETURN
640 *****
650 Alles richtig gemacht
660 *****
670 IF sc<6750 THEN PEN 1:MODE 1:TAGOFF:
PRINT CHR$(23)CHR$(0):LOCATE 15,12:PRIN
T"OH,SO NOT !!!":PEN 0:END
680 PEN 1:MODE 1:TAGOFF:PRINT CHR$(23)CH
R$(0):
690 DATA Y,D,U," ",H,A,V,E," ",B,R,O,K,E
,N," ",F,I,R,S,T," ",A,R,E,A," ",!,:!
700 RESTORE 690:LOCATE 6,10:FOR re=1 TO
30:READ BILD$:PRINT bild$:SOUND 1,70,1,
7:FOR tim=1 TO 70:NEXT:
710 FOR posi=400 TO 150 STEP-2:TAG:CALL
&BD19:MOVE posi,200:PRINT"ab ":SOUND 2,
500,8,15,0,0,1:SOUND 1,0,10:NEXT
720 LOCATE 13,13:PRINT"CONGRATULATIONS a
b":TAGOFF
730 RESTORE 750:FOR soun=1 TO 11:READ me
in,laenge
740 SOUND 4,melo,laenge,15,1,1
750 DATA 478,50,0,3,478,20,0,2,478,20,0,
10,358,50,0,10,426,20,0,15,319,75
760 NEXT
770 TAGOFF:LOCATE 16,17:PRINT"SCORE":sc:
PEN 2:LOCATE 19,21:PRINT"DEL":PEN 1
780 IF INKEY(79)=0 THEN 790 ELSE 780
790 CLS:PEN 2:POKE &B1C8,0:LOCATE 6,5:PR
INT"AIR FIGHTER":POKE &B1C8,1:PEN 1
800 LOCATE 17,9:PRINT nam$+"",
810 LOCATE 3,13:PRINT"TRAGE DICH IN DIE
"HALL OF BEST EIN"
820 LOCATE 10,17:INPUT "YOUR NAME : ",ha
llnam$
830 IF sc<7000 THEN LOCATE 16,21:PRINT"P
LATZ - 8" ELSE IF sc<7500 THEN LOCATE 16
,21:PRINT"PLATZ - 6" ELSE IF sc<8000 THE
N LOCATE 16,21:PRINT"PLATZ - 4" ELSE IF

```

```

sc<8500 THEN LOCATE 16,21:PRINT"PLATZ -
3"
840 IF sc>=8500 AND sc<9000 THEN LOCATE
16,21:PRINT"PLATZ - 2"
850 IF sc>=9000 THEN LOCATE 16,21:PEN 2
:PRINT "PLATZ - ":PEN 3
860 RESTORE 880:FOR soun=1 TO 12:READ me
in,laenge
870 SOUND 4,melo,laenge,15,1,1
880 DATA 478,50,0,3,478,20,0,2,478,20,0,
10,358,50,0,10,426,20,0,15,319,75
890 NEXT:LOCATE 24,21:PRINT"1":PEN 1
900 IF INKEY(79)=0 THEN 910 ELSE 900
910 CLS:LOCATE 10,12:PRINT"ANOTHER GAME
(J/N) ?"
920 IF INKEY(45)=0 THEN RUN ELSE IF INKE
Y(46)=0 THEN CLS:SYMBOL AFTER 256:END EL
SE 920
930 *****
940 Bewegung Hubi
950 *****
960 POKE &B338,0:PRINT CHR$(23):CHR$(1):
TAG
970 z=PEEK(&B4F4):GOSUB 1180:IF fue=144
THEN z=0
980 IF xstep=0 AND ystep=0 THEN t=t+1:IF
t>20 THEN t=0:ystep=-1
990 IF z AND 1 THEN sou=sou-1:GOSUB 1320
:ystep=ystep+1
1000 IF z AND 2 THEN sou=sou-1:GOSUB 132
0:ystep=ystep-1
1010 IF z AND 4 THEN sou=sou-1:GOSUB 132
0:xstep=xstep-2
1020 IF z AND 8 THEN sou=sou-1:GOSUB 132
0:xstep=xstep+2
1030 IF z AND 16 THEN SOUND 129,0,20,15,
2,0,20:MOVE x+17,y-14:DRAW x,y-50:POKE &
B338,240:MOVE x+17,y-14:DRAW x,y-50:IF y
-50<=1 THEN lev=lev+1:IF lev>9 THEN 640
ELSE:TAGOFF:PRINT CHR$(23)CHR$(0):GOTO
1700
1040 x=x+xstep:y=y+ystep:IF sou<2 THEN s
ou=2
1050 SOUND 2,500,10,15,0,0,1:SOUND 1,0,s
ou:sou=sou+1:IF sou>10 THEN sou=10
1060 IF y>350 OR y<32 THEN ystep=0
1070 IF x>608 THEN x=608:xstep=0
1080 IF y<20 THEN y=20:ystep=0
1090 IF x<0 THEN x=0:xstep=0
1100 res=res+1:IF res=15 THEN res=0:GOSU
B 1320
1110 MOVE x1,y1:PRINT hub$(1):
1120 POKE &B338,240:MOVE x,y:PRINT hub$(
1):
1130 IF TEST(x+10,y-14)=0 OR TEST(x+22,y
-14)=0 THEN 1560
1140 IF TEST(x+10,y-14)=3 OR TEST(x+22,y
-14)=3 THEN GOSUB 1370
1150 IF TEST(x+2,y)=0 OR TEST(x+30,y)=0
THEN 1560
1160 IF TEST(x+10,y-14)=2 OR TEST(x+22,y
-14)=2 THEN GOSUB 1500
1170 x1=x:y1=y:GOTO 970
1180 *****
1190 Reste Loeschen
1200 *****
1210 TAGOFF:PRINT CHR$(23):CHR$(0):
1220 IF m=1 AND x<525 THEN LOCATE manx(5
),many(5):PRINT "":m=0:ou=ou+1
1230 IF m=2 AND x<525 THEN LOCATE manx(4
),many(4):PRINT "":m=0:ou=ou+1
1240 IF m=3 AND x<525 THEN LOCATE manx(3
),many(3):PRINT "":m=0:ou=ou+1
1250 IF m=4 AND x<525 THEN LOCATE manx(2
),many(2):PRINT "":m=0:ou=ou+1
1260 IF m=5 AND x>75 THEN LOCATE manx(1
),many(1):PRINT "":m=0:ou=ou+1
1270 IF m=6 AND x>75 THEN LOCATE manx(6
),many(6):PRINT "":m=0:ou=ou+1
1280 IF o=1 AND x>75 THEN LOCATE fasn,f
asy:PRINT "":o=0
1290 IF ou>=6 AND x<220 AND y>345 THEN 1
ev=lev+1:If lev>9 THEN 640 ELSE sc=sc+15
0:fue=637:GOTO 1700
1300 IF tank=1 AND x>75 THEN LOCATE fas
x,fasy:PRINT "":
1310 PRINT CHR$(23):CHR$(1):TAG:RETURN
1320 *****
1330 Fuelverbrauch
1340 *****
1350 IF fue>135 THEN MOVE fue,2:DRAW 0,
10,2:fue=fue-2:ELSE GOTO 1630
1360 POKE &B338,240:RETURN
1370 *****
1380 Maenner aufnehmen
1390 *****

```

```

1400 IF m=0 THEN sc=sc+100:s=1
1410 IF x>500 AND y<150 THEN m=1
1420 IF x>500 AND y<200 AND y>105 THEN m
=2
1430 IF x>500 AND y<250 AND y>200 THEN m
=3
1440 IF x>500 AND y>250 AND y<350 THEN m
=4
1450 IF x<100 AND y>250 AND y<350 THEN m
=5
1460 IF x<100 AND y<200 AND y>130 THEN m
=6
1470 LOCATE #1,10,2:PRINT #1,sc
1480 IF s>0 THEN FOR t=15 TO 8 STEP-0.5:
SOUND 1,379,10,t:NEXT:s=0:RETURN
1490 RETURN
1500 *****
1510 Oilfass beruehrt
1520 *****
1530 IF x<140 AND fue<600 AND tank=0 THE
N TAGOFF:PRINT CHR$(23)CHR$(0):fue=637:
pe=1:fu=1:GOSUB 1760:TAG
1540 IF fu=1 AND tank=0 THEN FOR H=15 TO
8 STEP-0.5:SOUND 1,478,10,H:SOUND 2,458
,10,H:NEXT:o=1:tank=1
1550 RETURN
1560 *****
1570 Kollision
1580 *****
1590 OUT &BC00,8:OUT &BD00,1:FOR i=15 TO
5 STEP-0.2:SOUND 1,0,5,i,2,0,10:MOVE x,
y:PRINT"EE":NEXT:OUT &BC00,8:OUT &BD00,
2:m=0:PEN 13
1600 TAGOFF:PRINT CHR$(23):CHR$(0):fue=
fue-30:FOR a=637 TO fue STEP-1:PLOT a,2:
DRAW a,12,2:SOUND 1,a,1.5,15:NEXT:IF fue
<135 THEN GOTO 1630
1610 LOCATE 1,1:GOSUB 1770
1620 GOSUB 400:GOTO 930
1630 *****
1640 Kein Fuel Mehr
1650 *****
1660 TAG:OUT &BC00,8:OUT &BD00,1:FOR i=1
5 TO 5 STEP-0.1:SOUND 1,0,5,i,2,0,10:MOV
E x,y:PRINT"EE":NEXT:OUT &BC00,8:OUT &B
D00,2:PEN 13
1670 IF sc>hsc THEN hsc=sc:LOCATE #1,34,
2:PRINT #1,hsc
1680 TAGOFF:LOCATE 12,13:PRINT" G A M E
O V E R "
1690 IF JOY(0)=16 THEN 2810 ELSE 1690
1700 *****
1710 Bildschirmaufbau
1720 *****
1730 PRINT CHR$(23)CHR$(0):TAGOFF:INK 1
,26:INK 2,11:INK 3,6:INK 0,0:BOARD 1:PA
PER 0:pe=1:tank=0:fu=0:PEN 1:MODE 1
1740 WINDOW#1,1,40,1,3:PAPER #1,2:CLS #1
:PLOT 0,352,13:DRAW 639,352:DRAW 639,399
:DRAW 0,399:DRAW 0,352:WINDOW #2,1,40,25
,25:PAPER #2,2:CLS #2:PLDT 0,0,13:DRAW 6
39,0:DRAW 639,15:DRAW 0,15:DRAW 0,0
1750 LOCATE #1,3,2:PEN #1,1:PRINT #1,"sc
ore ":sc:LOCATE #1,16,2:PRINT #1,"level
":lev:LOCATE #1,28,2:PRINT #1,"high ":
hsc
1760 LOCATE #2,3,1:PRINT #2,"fuel ":FOR
a=fue TO 135 STEP-2:PLOT a,2,pe:DRAW a,
12:SOUND 1,a,1.5,15:NEXT a:PLOT 0,0,13:D
RAW 640,0:PLOT 0,15,13:DRAW 640,15:IF fu
=1 THEN RETURN ELSE FOR H=15 TO 8 STEP-
0.5:SOUND 1,478,10,H:NEXT h
1770 ON lev GOSUB 1810,2060,2310,2560,34
70,3720,3970,4220,4470
1780 PEN 2:FOR i=1 TO 6:LOCATE manx(i),m
any(i):PRINT "y":NEXT:PEN 3:FOR i=1 TO 3
:LOCATE fasn,fasy:PRINT "":NEXT
1790 IF lev>1 THEN GOSUB 400:GOTO 930
1800 RETURN
1810 PRINT:PRINT:PRINT:PRINT "xxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxx"
1820 PRINT "xxxxx xxxxxxxx
xx"
1830 PRINT "xx xxxxx
xx"
1840 PRINT "xx xx xxxxx
xxxxxxxxxxxx"
1850 PRINT "xxxxxxx xxxxxxxx
xxxxxxxxxxxx"
1860 PRINT "xxxxxxxx xxxxxxxxxxxx
xxxxxxxxxxxx"
1870 PRINT "xxxxxxx xx xxxx
xx"
1880 PRINT "xx xx xxxx
xx"
1890 PRINT "xx xx xxxx xxx

```

```

xxxxxxx";
1900 PRINT "xxxxxxx"      xxx   xxx
xxxxxxx";
1910 PRINT "xxxxxxx"      xxx   xxx
xxxxxxx";
1920 PRINT "xxxxxxx"      xxxxxx
    xx";
1930 PRINT "xx"           xxxxxx
    xx";
1940 PRINT "xx"           xxx
xxxxxxx";
1950 PRINT "xxxxxxx"      xxx
xxxxxxx";
1960 PRINT "xxxxxxxxxxxxxxxxxxxxxxx" xxx
xxxxxxx";
1970 PRINT "xxxxxxxxxxxxxxxxxxxxxxx"
    xx";
1980 PRINT "xx"
    xx";
1990 PRINT "xx"           xxx
xxxxxxx";
2000 PRINT "xx"           xxxxxxxxxxxxxxxxxxxx
xxxxxxx";
2010 PRINT "xx"           xxxxxxxxxxxxxxxxxxxx
xxxxxxx";
2020 RESTORE 2030:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
2030 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
2040 fasx=4:fasy=12
2050 RETURN
2060 PRINT:PRINT:PRINT "xxxxxxx"
xxxxxxx";
2070 PRINT "xxxxxxx"      xxxxx
x   xx";
2080 PRINT "xx"           xxxxx
x   xx";
2090 PRINT "xx"           xxxxxxxxxxx xxxxx
x   xxxxxx";
2100 PRINT "xxxxxxx"      xxxxxxxxxxx
xxxxxxx";
2110 PRINT "xxxxxxx"
xxxxxxx";
2120 PRINT "xxxxxxx"      xxxxx
x   xx";
2130 PRINT "xx"           xxxxxxxxxxx xxxxx
x   xx";
2140 PRINT "xx"           xxxxxxxxxxx
xxxxxxx";
2150 PRINT "xxxxxxx"      xxxxx
xxxxxxx";
2160 PRINT "xxxxxxx"      xxxxx
x   xxxxxx";
2170 PRINT "xxxxxxx"      xxxxxxxxxxx xxxxx
x   xx";
2180 PRINT "xx"           xxxxxxxxxxx xxxxx
x   xx";
2190 PRINT "xx"           xx           xxxxx
xxxxxxx";
2200 PRINT "xxxxxxx"      xx           xxxxx
xxxxxxx";
2210 PRINT "xxxxxxx"      xxxxxxxxxxx xxxxx
xxxxxxx";
2220 PRINT "xxxxxxx"      xxxxxxxxxxx xxxxx
x   xx";
2230 PRINT "xx"           xxxxxxxxxxx xxxxx
x   xx";
2240 PRINT "xx"           xxxxxxxxxxx
xxxxxxx";
2250 PRINT "xx"           xxxxxxxxxxx
xxxxxxx";
2260 PRINT "xx"           xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2270 RESTORE 2280:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
2280 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
2290 fasx=4:fasy=12
2300 RETURN
2310 PRINT:PRINT:PRINT "xxxxxxxxxxx"
xxxxxxx";
2320 PRINT "xxxxxxxxxxx"      xx
x   xx";
2330 PRINT "xx"           x           xx
x   xx";
2340 PRINT "xx"           xx xxxxxxxxxxx xx
x   xxxxxx";
2350 PRINT "xxxxxxx"      xx           xx
x   xxxxxx";
2360 PRINT "xxxxxxx"      xx           xx
x   xxxxxx";
2370 PRINT "xxxxxxx"      xxx xxxxxxxxxxx xx
x   xx";
2380 PRINT "xx"           xxx           xx
x   xx";
2390 PRINT "xx"           xxx           xx
x   xx";
2400 PRINT "xxxxxxx"      xxx xxxxxxxx xx
x   xxxxxx";
2410 PRINT "xxxxxxx"      xxx           xx
x   xxxxxx";
2420 PRINT "xxxxxxx"      xxx           xx
x   xx";
2430 PRINT "xx"           xxx xxxxxx xx
x   xx";
2440 PRINT "xx"           xxx           xx
x   xxxxxx";
2450 PRINT "xxxxxxx"      xxx           xx
x   xxxxxx";
2460 PRINT "xxxxxxx"      xxx           xx
x   xxxxxx";
2470 PRINT "xxxxxxx"      xxx           xx
x   xx";
2480 PRINT "xxxxxxx"      xxx           xx
x   xx";
2490 PRINT "xx"           xxxxxxxx xxxxxxxx
x   xxxxxx";
2500 PRINT "xx"           xxxxxxxx
xxxxxxx";
2510 PRINT "xx"           xxxxxx";
2520 RESTORE 2530:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
2530 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
2540 fasx=4:fasy=12
2550 RETURN
2560 PRINT:PRINT:PRINT "xxxxxxx"
xxxxxxx";
2570 PRINT "xxxxxxx"      xxxxx
    xx";
2580 PRINT "xx"
    xx";
2590 PRINT "xx"           xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2600 PRINT "xxxxxxx"      xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2610 PRINT "xxxxxxx"      xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2620 PRINT "xxxxxxx"
    xx";
2630 PRINT "xx"
    xx";
2640 PRINT "xx"           xxxxx xxxxxxxxxx
xxxxxxx";
2650 PRINT "xxxxxxx"      xxxxx xxxxxxxxxx
xxxxxxx";
2660 PRINT "xxxxxxx"      xxxxx xxxxxxxxxx
xxxxxxx";
2670 PRINT "xxxxxxx"      xxxxx
    xx";
2680 PRINT "xx"           xxxxx
    xx";
2690 PRINT "xx"           xxxxx xxxxxxxxxx
xxxxxxx";
2700 PRINT "xxxxxxx"      xxxxx xxxxxxxxxx
xxxxxxx";
2710 PRINT "xxxxxxx"      xxxxx xxxxxxxxxx
xxxxxxx";
2720 PRINT "xxxxxxx"      xxxxx
    xx";
2730 PRINT "xx"           xxxxx
    xx";
2740 PRINT "xx"           xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2750 PRINT "xx"           xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2760 PRINT "xx"           xxxxxxxxxxxxxxxxxxx
xxxxxxx";
2770 RESTORE 2780:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
2780 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
2790 fasx=4:fasy=12
2800 RETURN
2810 *****
2820 "Belohnung Ja o. Nein
2830 *****
2840 MODE 1:LOCATE 12,4:PEN 2:PRINT"A I
R F I G H T E R":LOCATE 12,8:PEN 1:PRINT
"YOUR POINTS .":sc:LOCATE 12,11:PRINT"
HE HIGHSCORE":hsc
2850 IF sc>1000 THEN LOCATE 2,16:PEN 3:P
RINT nam$:PEN 1:LOCATE 10,16:PRINT"IT CO
ULD BE BETTER NEXT TIME !"
2860 IF SC=hsc OR SC>2500 THEN LOCATE 3,
16:PEN 3:PRINT nam$:PEN 1:LOCATE 11,16:P
RINT"YOU'RE VERY GOOD THIS TIME !"
2870 IF SC<1000 THEN LOCATE 3,16:PEN 3:P
RINT nam$:PEN 1:LOCATE 11,16:PRINT"YOU'R
E VERY BAD THIS TIME !!"
2880 LOCATE 7,20:PEN 2:PRINT"WILL YOU TR
Y IT AGAIN <J/N> ?"
2890 a$=INKEY$:IF a$="" THEN 2890
2900 IF a$="j" OR a$="J" THEN RUN
2910 IF a$="n" OR a$="N" THEN INK 3,26,0
:SPPEED INK 30,30:PEN 3:LOCATE 18,23:PRIN
T"GOOD BEY":IF INKEY$<>"N" THEN PEN 0:EN
D
2920 IF a$<>"j" THEN 2890
2930 GOTO 2900
2940 *****
2950 "Titelbild
2960 *****
2970 CLS:MODE 1
2980 PEN 2:POKE &B1CB,0:LOCATE 6,4:PRINT
"AIREFIGHTER":POKE &B1CB,1
2990 PEN 1:LOCATE 16,8:PRINT"PRESENT BY"
3000 LOCATE 19,23:PEN 2:PRINT"DEL":PEN 1
3010 LOCATE 11,12:PRINT"x x x x xx
xx"
3020 LOCATE 11,13:PRINT"xx xx x x
x"
3030 LOCATE 11,14:PRINT"x x x x xx
xx"
3040 LOCATE 11,15:PRINT"x x xx x x"
3050 LOCATE 11,16:PRINT"x x x x x"
3060 MK=595:POKE &B33B,0:PRINT CHR$(23)C
HR$(1):TAG
3070 EVERY 3 GOSUB 3090
3080 GOTO 3110
3090 MK=MK-1:MOVE MK,96:PRINT"MKP":POKE
&B33B,240:IF MK=5 THEN MOVE mk,96:PRINT
"MKP":POKE &B33B,0:MK=595
3100 RETURN
3110 IF INKEY(79)=0 THEN MOVE 289,46:PRI
NT"DEL":PEN 1:GOTO 3120 ELSE 3110
3120 ENV 1,=11,7000:ENV 2,1,15,1,15,-1,5
:ENT -1,3,0.75,6,3,-0.75,6:ENT -2,2,4,
2,-2,4,1,1,7,1,-1,7
3130 BMUS=4:XMUS=478:RESTORE 3200
3140 READ AMUS:IF AMUS=0 THEN RESTORE 32
10:READ AMUS:IF BMUS=4 THEN BMUS=2 ELSE
BMUS=4
3150 IF AMUS<>1 THEN SOUND 130,AMUS*2,90
,7,1,1:GOUND 132,AMUS*BMUS,160,7,1,2
3160 SOUND 1,XMUS,24,10,2,2:IF XMUS=478
THEN XMUS=379 ELSE XMUS=478
3170 IF INKEY(79)=0 THEN A=REMAIN(0):GOT
O 310
3180 WHILE SB(1)>127:WEND
3190 GOTO 3140
3200 DATA 1,1,1,1,119,1,1,80,1,1,1,1,1
,119,1,1,80,1,1,1,1,1
3210 DATA 119,89,80,71,80,1,95,1
3220 DATA 1,119,89,80,71,80,1
3230 DATA 1,1,119,89,80,71,80,1,95,1
3240 DATA 1,95,89,95,119,119,1
3250 DATA 1,1,60,63,71,80,63,1,80,71,1,8
9,80
3260 DATA 1,60,63,71,80,63,1,80,71,1,89,
80
3270 DATA 1,95,89,95,119,119
3280 DATA 1,1,1,1,1,1,1,0
3290 *****
3300 "Instructionen
3310 *****
3320 PRINT CHR$(23)CHR$(0):TAGOFF:MODE
1:LOCATE 11,2:PEN 2:PRINT"A I R F I G H
T E R"
3330 PEN 1:LOCATE 14,6:INPUT "NAME: ",na
m$
3340 IF LEN(nam$)>7 OR LEN(nam$)<1 THEN
3290
3350 LOCATE 14,6:PRINT"OKEY":LOCATE 6,1
0:PRINT"INSTRUCTIONS, JA ODER NEIN ?"
3360 PEN 3:a$=INKEY$:IF a$="" THEN 3360
3370 IF a$="J" OR a$="j" THEN 3400
3380 IF a$="N" OR a$="n" THEN RETURN
3390 IF a$<>"J" THEN 3360
3400 LOCATE 14,13:PEN 3:PRINT"INSTRUCTIO
NEN":LOCATE 20,10:PRINT"JA"
3410 PEN 1:LOCATE 3,16:PRINT"DU MUSST VE
RSUCHEN, DEN Airfighter ":PRINT" DURC
H DIE HEHLEN ZU STEUERN UND ":PRINT"
DIE MAENNER ZU RETTEN,ABER PASS ":PRIN
T" AUF DIE WAENDE AUF ! FALLS DEIN ":
PRINT" fuel AUS GEHT,KANNST DU DAS oil
":
3420 PRINT" FASS HOLEN.UM DAS level ZU
UEBER-":PRINT" SPRINGEN MUSST DU AM AN
FANG ETWAS":PRINT" HERUMBALLERN,BEKOMM
ST ABER KEINE ":PRINT" PUNKTE DAZU. V

```



```

IEL GLUECK !!!
3430 IF INKEY<>" THEN RETURN ELSE 3430
3440 *****
3450 Zusatzliche Levels
3460 *****
3470 PRINT:PRINT:PRINT:PRINT "XXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
3480 PRINT "xx xxx
xx";
3490 PRINT "xx xxx
xx";
3500 PRINT "xx xxx XXXXXXXX
XXXXXXXXXXXXXXXXX";
3510 PRINT "XXXXXXXXX xxx XXXXXXXX
XXXXXXXXXXXXXXXXX";
3520 PRINT "XXXXXXXXX xxx XXXXXXXX
XXXXXXXXXXXXXXXXX";
3530 PRINT "XXXXXXXXX
xx";
3540 PRINT "xx
xx";
3550 PRINT "xx xxx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3560 PRINT "XXXXXXXXX xxx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3570 PRINT "XXXXXXXXX xxx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3580 PRINT "XXXXXXXXX xxx
xx";
3590 PRINT "xx xxx
xx";
3600 PRINT "xx XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3610 PRINT "XXXXXXXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3620 PRINT "XXXXXXXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3630 PRINT "XXXXXX
xx";
3640 PRINT "XXXXX
xx";
3650 PRINT "XXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3660 PRINT "XXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3670 PRINT "xx XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3680 RESTORE 3690:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
3690 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
3700 fasx=4:fasy=12
3710 RETURN
3720 PRINT:PRINT:PRINT:PRINT "XXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
3730 PRINT "XXXXX
xx";
3740 PRINT "xx
xx";
3750 PRINT "xx
XXXXXXXXX";
3760 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
3770 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX XXXXX";
3780 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX XX";
3790 PRINT "xx
xx";
3800 PRINT "xx
XXXXXXXXX";
3810 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
3820 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX XXXXX";
3830 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX XX";
3840 PRINT "xx
xx";
3850 PRINT "xx
XXXXXXXXX";
3860 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
3870 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XX XXXXX";
3880 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXX XX";
3890 PRINT "XXXX
xx";
3900 PRINT "XXXX
XXXXXXXXX";
3910 PRINT "XXXX XXXXXXXX
XXXXXXXXXXXXXXXXX";
3920 PRINT "XXXX XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
3930 RESTORE 3940:FOR i=1 TO 6:READ manx

```

```

(i),many(i):NEXT
3940 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
3950 fasx=4:fasy=12
3960 RETURN
3970 PRINT:PRINT:PRINT:PRINT "XXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
3980 PRINT "xx
xx";
3990 PRINT "xx
xx";
4000 PRINT "xx XXXXXXXX
XXXXXXXXXXXXXXXXX";
4010 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX
XXXXXXXXXXXXXXXXX";
4020 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4030 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xx";
4040 PRINT "xx
xx";
4050 PRINT "xx XXXXX
XXXXXXXXXXXXXXXXX";
4060 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4070 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4080 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xx";
4090 PRINT "xx
xx";
4100 PRINT "xx XXX
XXXXXXXXXXXXXXXXX";
4110 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4120 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X
XXXXXXXXXXXXXXXXX";
4130 PRINT "XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
xx";
4140 PRINT "XXXX
xx";
4150 PRINT "XXXX X
XXXXXXXXXXXXXXXXX";
4160 PRINT "xx XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4170 PRINT "xx XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4180 RESTORE 4190:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
4190 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
4200 fasx=4:fasy=12
4210 RETURN
4220 PRINT:PRINT:PRINT:PRINT "XXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
4230 PRINT "XXXXX
XXXXX xx";
4240 PRINT "xx XXXXXXXXXXXX
XXX XX";
4250 PRINT "xx XXXXXXXX
XX XXXXX";
4260 PRINT "XXXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
4270 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX
XXXXXXXXX";
4280 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXX
xx";
4290 PRINT "xx
xx";
4300 PRINT "xx
XXXXXXXXX";
4310 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXX";
4320 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXXXXXXX";
4330 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXX
xx";
4340 PRINT "xx XXXXXXXX
xx";
4350 PRINT "xx XXXXXXXX
X XXXXXXXX";
4360 PRINT "XXXXXXXXXXXXXXXXXXXXX
XX XXXXXXXX";
4370 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXX XXXXX";
4380 PRINT "XXXXXXXXXXXXXXXXXXXXX
XXXX XXX";
4390 PRINT "xx XXXXXXXXXXXX
XXXXX";
4400 PRINT "xx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4410 PRINT "xx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4420 PRINT "xx XXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4430 RESTORE 4440:FOR i=1 TO 6:READ manx

```

```

(i),many(i):NEXT
4440 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
4450 fasx=4:fasy=12
4460 RETURN
4470 PRINT:PRINT:PRINT:PRINT "XXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX";
4480 PRINT "xx X X X
X xx";
4490 PRINT "xx X X X
X xx";
4500 PRINT "xx X X X
X XXXXXXXX";
4510 PRINT "XXXXX X X X
X XXXXXXXX";
4520 PRINT "XXXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
4530 PRINT "XXXXX X X X
X xx";
4540 PRINT "xx X X X
X xx";
4550 PRINT "xx X X X
X XXXXXXXX";
4560 PRINT "XXXXX X X X
X XXXXXXXX";
4570 PRINT "XXXXX X XXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
4580 PRINT "XXXXX X X X
X xx";
4590 PRINT "xx X X X
X xx";
4600 PRINT "xx X X X
X XXXXXXXX";
4610 PRINT "XXXXX X X X
X XXXXXXXX";
4620 PRINT "XXXXX XXXXXXXXXXXXXXXXXXXX
X XXXXXXXX";
4630 PRINT "XXXX
xx";
4640 PRINT "XXXX
xx";
4650 PRINT "xx X X
XXXXXXXXXXXXXXXXX";
4660 PRINT "xx X XXXXXXXX
XXXXXXXXXXXXXXXXX";
4670 PRINT "xx XXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXX";
4680 RESTORE 4690:FOR i=1 TO 6:READ manx
(i),many(i):NEXT
4690 DATA 4,7,36,6,36,11,36,16,36,21,4,1
7
4700 fasx=4:fasy=12
4710 RETURN

```



CHECK V1:PRUEFSUMMEN:

760-	770:	63044
780-	790:	17192
800-	810:	74319
820-	830:	51291
840-	850:	52896
860-	870:	33653
880-	890:	50926
900-	910:	46185
920-	930:	18295
940-	950:	18254
960-	970:	25574
980-	990:	35831
1000-	1010:	9087
1020-	1030:	53403
1040-	1050:	44568
1060-	1070:	96791
1080-	1090:	78997
1100-	1110:	33655
1120-	1130:	83737
1140-	1150:	76314
1160-	1170:	95004
1180-	1190:	21527
1200-	1210:	15910
1220-	1230:	6915
1240-	1250:	6639
1260-	1270:	8851
1280-	1290:	59041
1300-	1310:	38969
1320-	1330:	19944
1340-	1350:	58720
1360-	1370:	14724
1380-	1390:	29664
1400-	1410:	87248
1420-	1430:	15831
1440-	1450:	11207
1460-	1470:	14596

# kleinanzeigen

1480-1490: 58363	2020-2030: 53877	2560-2570: 30255	3100-3110: 52005	3640-3650: 63952	4180-4190: 53346
1500-1510: 27142	2040-2050: 19733	2580-2590: 56632	3120-3130: 38765	3660-3670: 93216	4200-4210: 19733
1520-1530: 57430	2060-2070: 48255	2600-2610: 91416	3140-3150: 36069	3680-3690: 53384	4220-4230: 21719
1540-1550: 59967	2080-2090: 48712	2620-2630: 36248	3160-3170: 55024	3700-3710: 19733	4240-4250: 126248
1560-1570: 11519	2100-2110: 45712	2640-2650: 71496	3180-3190: 8196	3720-3730: 30015	4260-4270: 25896
1580-1590: 43889	2120-2130: 95648	2660-2670: 68872	3200-3210: 22112	3740-3750: 54848	4280-4290: 72968
1600-1610: 24874	2140-2150: 106448	2680-2690: 59272	3220-3230: 31444	3760-3770: 60816	4300-4310: 51472
1620-1630: 10438	2160-2170: 123848	2700-2710: 76296	3240-3250: 45542	3780-3790: 35512	4320-4330: 54232
1640-1650: 19551	2180-2190: 68992	2720-2730: 56648	3260-3270: 40065	3800-3810: 69832	4340-4350: 98288
1660-1670: 24275	2200-2210: 53016	2740-2750: 81816	3280-3290: 12968	3820-3830: 49896	4360-4370: 30816
1680-1690: 36160	2220-2230: 119048	2760-2770: 67235	3300-3310: 20323	3840-3850: 54848	4380-4390: 20496
1700-1710: 27806	2240-2250: 125048	2780-2790: 47025	3320-3330: 25696	3860-3870: 60816	4400-4410: 63696
1720-1730: 31505	2260-2270: 73273	2800-2810: 11926	3340-3350: 74427	3880-3890: 37072	4420-4430: 60982
1740-1750: 93502	2280-2290: 47025	2820-2830: 33469	3360-3370: 53658	3900-3910: 62632	4440-4450: 47025
1760-1770: 115943	2300-2310: 13892	2840-2850: 115818	3380-3390: 37630	3920-3930: 73020	4460-4470: 11612
1780-1790: 46271	2320-2330: 64448	2860-2870: 104385	3400-3410: 116675	3940-3950: 47025	4480-4490: 51608
1800-1810: 9932	2340-2350: 88192	2880-2890: 73412	3420-3430: 64968	3960-3970: 11972	4500-4510: 111488
1820-1830: 61928	2360-2370: 115328	2900-2910: 64754	3440-3450: 35406	3980-3990: 31448	4520-4530: 54352
1840-1850: 40296	2380-2390: 91568	2920-2930: 9947	3460-3470: 22530	4000-4010: 57936	4540-4550: 80288
1860-1870: 68152	2400-2410: 65272	2940-2950: 11242	3480-3490: 42968	4020-4030: 86992	4560-4570: 86872
1880-1890: 49552	2420-2430: 78608	2960-2970: 4143	3500-3510: 52656	4040-4050: 27112	4580-4590: 54128
1900-1910: 40776	2440-2450: 118568	2980-2990: 74654	3520-3530: 51232	4060-4070: 78456	4600-4610: 111488
1920-1930: 83048	2460-2470: 94808	3000-3010: 38241	3540-3550: 49792	4080-4090: 68848	4620-4630: 42352
1940-1950: 14376	2480-2490: 56632	3020-3030: 47005	3560-3570: 80376	4100-4110: 43296	4640-4650: 80288
1960-1970: 88912	2500-2510: 73808	3040-3050: 31435	3580-3590: 49088	4120-4130: 97312	4660-4670: 67416
1980-1990: 20512	2520-2530: 53839	3060-3070: 43044	3600-3610: 87936	4140-4150: 81248	4680-4690: 53308
2000-2010: 93816	2540-2550: 19733	3080-3090: 26591	3620-3630: 65032	4160-4170: 66275	4700-4710: 19733

GESAMTCHECKSUMME: 11153

## \* Kleinanzeigen \* Kleinanzeigen \* Kleinanzeigen \*

### Biete Hardware

- + C16/116/+4 Hardware/Zubehör +
- + 16 K RAM Erweiterung 79.- +
- + Drucker Star Gemini 10598.- +
- + Monitor Phillips 8500548.- +
- + Plus 4239.- +
- + Floppy 1551379.- +
- + Joystickadapter 10.- +
- + Abdeckhauben ab 7.90 +
- + Datenset C15 10St +
- + 16.90 +
- + Comp.Service PF1304 7913 +
- + Senden +

**Sensationell!!!** Die Super 64K-Erweiterung für C16/116. Jetzt nur noch 85 DM! (incl.Sof+Tps)- einfacher Einbau ohne Löten!  
- Superindustrialqualität!  
- geringer Stromverbrauch  
- umschaltbar auf 16 K (2 Programme gleichzeitig im Speicher)  
Info: Hartmut Schäfer, Schutzbergstr. 16, 6600 Saarbrücken

**Warum Experimente?**  
Plus 4 Userport am C16. Nur 4 IC erforderlich. Wie?  
Gegen 20,- DM Vorkasse Manuskript anfordern (nur Schein oder Verrechnungsscheck)  
HWE Schmidt  
Trakenerstr. 19  
3004 Isernhagen

Verkaufe Original dk'tronics Light Pen incl. Software für nur 50,-.  
Jörn B. Gruhlike, Gartenstr. 6, 2347 Süderbrarup, Tel.: 04641/8396, bei Fam.Schmechel

Verkaufe C-116 + Datensette für 200,- DM. Bitte melden bei: Arno von Germerten, Triftweg 5, 3549 Volkmarsen, 05693/7612

**\*\* C16 / 116 RAM-Erweiterung !!!**  
C16/116 60671 Bytes free für DM 97,80  
- Alle Ports frei - keine Einschränkungen - incl. Umbau\*\*  
Computer ohne Zubehör mit 100,-DM (2,20 f. Post) an:  
Elektronik - Technik  
Tannenweg 9, D 2351 Trappenkamp

**\*\* C 16/116 RAM-Erweiterung!! \*\***  
C16/116 60671 Bytes free für DM 97,80- alle Ports frei - keine Einschränkungen - incl. Umbau \*\* Computer ohne Zubehör mit 100,-DM (2,20 DM für die Post) an:  
Elektronik-Technik  
Tannenweg 9, D 2351 Trappenkamp

**Commodore \*C-16/116\* C-16/116\*\***  
Speichererweiterung 64 KByte Bausatz mit allen benötigten Teilen \* Schalter für 16KByte - 64KByte umzuschalten und ausführlicher Bauanleitung für nur 49,50 DM (incl. MwSt. Porto u. Verpackung) per Nachnahme. Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel.: 02247/8111 ab 18 Uhr

**Schneider CPC 664 m. Grünm. + Floppy** für 550,-DM, sowie Commodore VC 20 mit verschiedenen Modulen für 60,-DM, Tel. 06423/2995

**Berlin, Berlin**  
W.A.W.-Elektronik, Tegelerstr. 2, 1000 Berlin 28, Tel.: 030/4043331, autorisierter Commodore-Vertragshändler bietet:  
Für C16/116 64K-Erweiterung m. Einbau 69,-DM, C-64 neues Design incl. GEOS 538,-, 128D 1698,-DM, Commodore Amiga 3695,-DM und Monitore, Floppys, Drucker, Zubehör

----- C-16 und C-116 -----  
**64KB Speichererweiterung** mit Schalter 16/64K. **Bausatz DM 49**  
**Umbauservice (Garantie!) DM 89**  
**UNEX 16 (4 ROM-Socket + RS 232 + 6522 VIA Port) nur DM 195**  
**1EP-16 Eprom-Brenner nur DM 228-**  
**4fach-Epromkarte 64 KB DM 69**  
**\*\* Info anfordern bei S&J HAAS\*\***  
Vohenstraußerstr.90, 8480 Weiden

**Für C16/116/plus 4 kostenlos!**  
Infos - Tips - Angebote - Umbauten  
\*\*\*\*\* vom Computerfachmann \*\*\*\*\*  
Erweiterungen + Reparaturen...  
Noch Fragen? Freilmschlag an:  
Elektronik-Technik-Peters  
Tannenweg 9, DE2351 Trappenkamp

VC-20 mit 5 Spielmodulen und 5 Kassetten für VB 150,-DM zu verkaufen, Heiko Göcke, Mackenstr.29, 4432 Gronau

Verkaufe: C64 + Floppy = 800,-DM, Drucker Seik.100-VC = 250,- (+Papier), Speedos + = 85,-, Datensette + Programme = 80,-, 5 Bücher = 100,-, 100 Disks = 350,- alles = 1500,- DM Tel.: 08232/5824 (Seppi verlangen)

Commodore: Alles für C16 C 128. EPROMS, Platinen, Zubehör, anfrage telefonisch oder gegen 2,-DM in Marken (werden angerechnet) bei:

- \*\*Fritz Zander
- \*\*Hopfenstr. 6
- \*\*5600 Wuppertal 1
- \*\*Tel.: 020/428311
- \*\*Auch Schneider und Atari\*\*

C-16 64K-Erweiterung incl. Einbau normal 65,-DM  
mit Umschaltung 16/64K 75,-DM  
Hanspeter Zölle, Roseggerstr.17, 7100 Heilbronn, 07131/80694

Turbofloppy Modul V 2.4  
Befehle wie Backup, Cat f. Directory, Header f. Formatierung, usw., Preis 150,-DM, Tel.: 08157/8942 ab 19:00 Uhr

Achtung C-16, 116, Plus 4-Be3sitzer-Biete Stromausfallsicherung für ihren Computer. Kein Programmverlust mehr bei Stromausfall. Nur DM 29,-!! Bausatz mit allen Einzelteilen nur DM 22,50!! SEHR einfach anzuschließen. Kein Eingriff ins Gerät notwendig.  
Harald Pliet, Erikastr. 2, 4534 Recke.

Suche Drucker (für C16): MPS 801, MPS 803, GP 250X, suche Datensette u. Floppy ab 15:00 Uhr Tel: 05807/200, Th. Kietzke

Suche Drucker Commodore, tech. o.K. MPS 801, 802, 803. Zahle b. DM 250 Harald Hähne, In der Freiheit 44, 5650 Solingen

**Farbbänder Seikosa 08761/4245**  
GP50-100-500-550 nur DM 19.50  
GP700 schw/verf. DM 28.-/44.-  
SP800/1000 DM 22.- MS15 DM 9.90  
=====

**Restposten Drucker (m.Garantie)**  
Farb-Dr. GP700 Centronics 599.-  
Klein-Dr. GP50 Centronics 199.-  
Disketten 2D/2S 10St. 29.90  
...Sumrex 2D/2S 10St. 39.90

Commodore C16 - Original 0 mit div. Zubehör und Spiele zu verkaufen - Preis nach Vereinbarung. Tel.: 09141/5510

Alles rund um Commodore, Atari und Schneider. Z.B. Kernal-o. Epromplatinen, Eprombrennservice und Hardware. Tauschbörse: Gebrauchte Soft- und Hardware zu niedrigen Preisen. Anfragen unter Tel.: 020/428311 oder schriftlich an Zander, Hopfenstr. 6, 5600 Wuppertal 1 (Rückporto).

### Biete Software

- \* original Elite plus V1.0 \*\*\* \*
- \* fantastische neue Program-
- \* miersprache für strukturiertes \*
- \* Programmieren auf Ihrem plus \*
- \* 4 / C16 / C116 (64KB) \*
- \* \*
- \* 29,- mit ausf. Handbuch, incl. \*
- \* Porto & Verp. Info geg. Rp. \*
- \* bei TEDsoft / Bohlingerstr.2 \*
- \* \*\*\*\*\* 7700 Singen \*\*\*\*\* \*



# kleinanzeigen

\*\*\*\* Software für \*\*\*\*  
 Atari 800XL, 130XE \* Atari 260ST, 520ST, 520ST+ \* Commodore VC-20, C64, C128, C16, C116, plus4 \* Commodore Amiga \* MSX-Computer \* Schneider, CPC-464,664,6128 \* kostenlos Liste anfordern (Bitte Computer angeben) \* Frank Schmitz, Kurzer Weg 1,5206 Neunkirchen 2  
 Tel.: 02247/8111 ab 18 Uhr

**C-64! NUR 1DM JE PROGRAMM!**  
**C-64!**  
 Clubsoftware (Anwender-, Spiel-, Lern-, Denk- und Musikprogramme gegen Kopiergebühr von nur 1 DM abzugeben. Z.B. Vokabeltrainer, Astrologieprgr. mit Bildschirm- und Druckerausgabe, Synthesizer, Biorhythmus, Actionspiele und Hilfsroutinen. Kein Schund! Info (Rückp) TEXASOFT, HERTHASTR.26, 8 MÜNCHEN 19

**Supergelegenheit auch einzeln**  
 5 Originalspiele für nur 55 DM  
 Oliver Windherst  
 Auf der Dohnhorst 11  
 2862 Worpswede 1

\*\*\*\*\*  
 \* **Spiel- und Anwenderprogramme** \*  
 \* für C 16/116 zu kleinen Preisen. Info gegen Rückporto bei \*  
 \* Matthias Poppen, Erlenweg \*  
 \* 11, 2955 Bunde \*  
 \*\*\*\*\*

**Verk. 19 Spiele** Cass. für C-16 wegen Systemwechsel Tel.: 08404/1252. Mo-Fr 17-19 Uhr, Neu: 350 DM, VB: 250 DM. Fan.4, Berks(123), Rob.to Resc., Petch, Worldcup, 3D TIME, Trek u.a.

**10 Top-Games** für den C-16 auf einer Kassette zu nur 20 DM. Dieses Gamespack besitzt: Turmspringen, Mimi, Poker, Ropulette, Dame, Super Mind usw. Tel.: 0504/3244

**\*\* SCHNEIDER CPC 464 \*\***  
 Verkaufe und tausche Spiele und Anwenderprogramme. Kostenlos Info bei: Ansgar Hütten, Schwartenpohl 11, 4455 Wietmarschen 2

+ + **C16/116/+4 Software** + +  
 + Bomb Jack.....Cass.24.90 +  
 + Frank Bruno's Boxing "24.90 +  
 + Yie are Kung Fu, Cass.29.- +  
 + Rambo, Cass.29.- +  
 + Winter Olympiade, Cass.22.90 +  
 + Superhits (6Pgr.) Cass.27.90 +  
 + King Size (50Pgr.)Cass.24.90 +  
 + 24Std Bestellservice +  
 + 07307/6230 +  
 + Comp.Service PF1304 7913 +  
 + Senden +

+ + **C16/116/+4** + +  
 + **Anwenderprogramme** + +  
 + Tubo Tape, Cass. 15.90 +  
 + Graphic Designer, Cass.15.90 +  
 + Paint Box, Cass.19.90 +  
 + Music Master, Cass.22.90 +  
 + Micro Date!, Cass.22.90 +  
 + Home Office, Cass.33.90 +  
 + Jetzt Neu: DISKETTENSOFT-WARE +  
 + Preise Stand 08/86 zzgl Ver-sand +  
 + Comp.Service PF1304 7913 +  
 + Senden +

**Achtung C 16/116/plus4 User!**  
 Originalprg. ab 5,-! Keine Kopien! Liste gegen Porto  
**Riesen Spezialmappe nur 6,-**  
 30 Prge.: Spiel + Anwend. nur 25,-  
**Playbit-Software \*\*\***  
 Trimbser Weg 26,  
 5440 Mayen 14

Top Spiele für C-16 Spielpaket:  
 Jump-Man, Pferderennen, Turmspringen, Black-Jack, Cowboy-Duell, Pac-Mac, auf einer Kassette zu verkaufen, Preis DM 10 + 2 DM für Porto und Verpackung. Tausche auch gern Spiele aus!  
 Jürgen Arns, Jugendheimstr. 1  
 5550 Bernkastel-Kuels

**PC 128 u. VC 64 Originalprogramme:**  
 Turbo Texter 128 (40/80 Z.) DM 24,80  
 Monarchie (Spiel) DM 15,-  
 VC 64 Programme (D=Disk, T=Tape):  
 Das Riff (D) DM 19,80, Astray in Jungle (D/T) DM 10,-, Kaufman (D) DM 10,-, Schneewittchen (D/T) DM 10,-, Schatzsuche (D/T) DM 10,-, Rapunzel (D/T) DM 10,-, Liste -80 in Briefmarken bei: Oldenburg, Postfach 1132, 2190 Cuxhaven 12

**C64, C128, VC 20, C 16/116**  
 Spiele, Termin, Lager, Fibu, EK-VK, Lot-to, Adress, Tabellen, Foto-, Film-, Video-, Musikarchiv, Digitalo, Denktraining, YSI, Kartei, Biblo, KfZ, Handbücher, ab 9.90 DM,  
 Katalog gegen 80 Pf-Marke  
 Computerservice T. Hofstede  
 An der Windmühle 8, 5010 Bergheim

\*\*\*\*\*  
 \* **\*\*\* Sprite Animation Set \*\*\*** \*  
 \* Fantastisch animierte Multi- \*  
 \* Color-Sprites erstellen, wie in \*  
 \* Topgames. Oder machen Sie \*  
 \* einen Trickfilm. \*  
 \* Menüsteuerung, Scrolling, \*  
 \* Spiegeln, Animationsroutine, \*  
 \* Demosprites, u.a. Disk für C \*  
 \* 64 nur 30 DM! \*  
 \* Schein an: Holger Geldermann \*  
 \* Schuhmannstr. 4, 4320 Hat- \*  
 \* tingen \*  
 \* \*\*\*\*\* \*  
 \* \*\*\*\*\* \*

Anspruchsvolle Lernsoftware für Schule u. Beruf für CPC Schneider. Liste bei: Dr. Kolb, Bergstr.34, 6900 Heidelberg, Tel.: 06221/474711, Programme auch nach Wunsch bestellbar!

Das beste Biorhythmusprogramm plus Partnerschaftsanalyse. Bei Druckerbetr. ca. 8 Din A4 Seiten pro Monat + Grafik sehr ausführliche Hinweise und Informationen für jeden Tag. Ein TOP-Softwarepaket, daß bereits erfolgreich auf kleinen Messen und Märkten Anwendung fand. Progr. (70K) Disk DM 40,- Kass. DM 30,- Ralph Hilterscheid, Ludwig-Rosenberg-Ring 47, 2050 HH-80, Tel. 040/7211974

CPC\* CPC\* CPC\*! Topgames?!  
 \*CPC\* CPC\* CPC  
 \* Wow! \*\$ Tausche Software \$\* Wow! \*  
 C Schickt Eure Listen an: Rolf C  
 P Eppinger Junior Fraunhoferweg 4 P  
 C 7440 Nürtingen Tel: 07022/8763 C  
 \* LOGISCHERWEISE: 3'Disc & Tape ! \*  
 CPC\* CPC\* CPC\*?? Hey you??  
 \*CPC\* CPC\* CPC

\* CPC Software zu Toppreisen! \*  
 \* tel. best. bei M+K 0821/91115 \*  
 \* S. Fox Strip.Poker 29,90DM \*  
 \* Bomb Jack 27,90DM \*  
 \* Kung Fu Master 30,90DM \*  
 \* Gun-Fright 33,90DM / Shogun \*  
 \* 29,90 / Biggles 33,90 / \*  
 \* Ghost'N'Goblins 29,90 / Green \*  
 \* Beret 29,90 / Tomahawk 29,90 \*  
 \* / Equino X 29,90 /The Comet \*  
 \* 24,90 \*

Verkaufe Original-Software für den C 64 auf Kassette z.B.: Beach Head 2/ Winter Games / Titanic / Fight Night für :DM 15,-. Rambo / War Play für DM 10,-. The Last V8 / Five-a-Side für DM 7,- und eine Menge Mastertronic-Spiele für 5,- DM. Ich tausche alle Spiele auch gegen : Herz von Afrika /Disk o. Kass., Frank Grosstück, Vor der Loos 6, 5427 Bad Ems, 02603/124321

**C-16, C-116 Superspiele!\*\*\*\*\***  
 Info gegen Rückporto bei Hannes Kaltenbach, Prielmayerstr. 16,7990 Friedrichshafen 1

Mc Stoney in the Caves. Kassette für C-16/116/+4 DM 10 + 3 DM Rückporto. Das Action- und Strategiespiel. Fantastische Grafik und anspruchsvoller Sound, 21 Hires-Multicolor-Screens. Versuchen Sie Diamanten zu sammeln und zu überleben. Dieses Spiel wird Sie fesseln. Bei C.REinke, Otto-Dix-Str.3, 1000 Berlin 21

Top Tape 64: Verkaufe (tausche) gute Originalsoft!!!  
 SEARCHING : Suche Floppy 1541 für Commodore 64 (möglichst billig)  
 M.Süssli, Flughafenstr. 16, CH-8302 Kloten, Tel. 0118/36255

\*\*\* CPC 464 \*\*\* Tausche Software aller Art in Basic & MC-Code  
 Suche auch Tauschpartner für meinen Grünmonitor gegen Farbe (Wertausgleich in DM oder Soft/Hardware)  
 C.Arn, Hauptstr. 17, 6464 Altenhaslau

\*\*\*\*\*  
 \* C16/116 Zeichen- und Grafik- \*  
 \* programme 16 K Version, und \*  
 \* CAD-System (Basis-System) \*  
 \* 64 K Version, für technische \*  
 \* Zeichnungen. Info gegen Por- \*  
 \* to: Dipl.Ing.M.Rätzel, Ulven- \*  
 \* bergstr.6, D-6100 Darmstadt \*  
 \* \*\*\*\*\* \*

Programme aus Eigenentwicklung für alle Schneider-CPC-Computer erhalten sie preisgünstig von Friedrich Neuper, Postfach 72, 8473 Pfreimd. Fordern sie einfach die kostenlose Preisliste an. Eine kurze Postkarte genügt.

VC-20 GAMES billig zu verkaufen  
 GV-32 KB; Spiele- Anwenderprogramme/AB 2 DM >>Maschinencode<<  
 >Assem.  
 Tel: 0234/412498 od. 0234/491127

Thomas-Soft-Tsamadou 4 Athina Greece  
 C-64,-128  
 GEOS!!! Auf Disk nur 30 DM, Megadisk v.4 nur 20 DM, 128 Modus-Theilast V8 10 DM, Wir haben 4000 Prg. für sie. Free INFO. I come every week in Germany.-Also Scenary Disk (west cost) Flightsimulator FF -Jetsimulator und mehr-Good Prices-

\*\*\*\*Hey! Ihr\* Freaks\*\*\*\* C-64  
 Hättet ihr nicht Lust meine Spiele zu tauschen oder kaufen! Christian Doeres, Friedrichstr. 5 b, 5550 Bernkastel-Kuels. \*C64\*

Plus 4 C 16-64K, Fußball Bundesliga Verwaltung. Tabelle wird nach Punkte und Tor diff. sort.Mit Anzeige für Gesamte-Gew.verl. unentsch. Spiele sowie Punkte+ Torestand + Ergebnisse je Verein und Spieltag - Rätselprogramm mit über 2000 Begriffen. Suchvorgang dauert nur wenige Sek. Nur Disk Dietmar Winhold, Rhönstr. 24, 6490 Schlüchtern

Public Domain Software für C 64 verkauft unser Club für DM 10,- pro Disk - Spiele, Comal, Utilities, Info: 80PF-Marke an M.Hub, Kennenburger Str. 32, 7300 Esslingen

\*\*\* Jupitersoft GdbR presents\*\*\*  
 C ..C64/128 Bomb Jack 29.90  
 C ..... Space Invasion 29.90  
 D ..... Critical Mass/Combat Lynx (Doppeldiskette) 49.90  
 C .SVI 318/328 Bruce Li 6.90  
 C .Vt200 /Laser Lemunia 14.50  
 C .ZX81...Chess 16k 19.90  
 Computertyp angeben! Info bei s.o., Höttinger Str. 34b, 8836 Ellinger

\*CPC\* CPC\* CPC\* TOPGAMES!  
 \*CPC\* CPC\*

Tausche CPC-Soft! Schickt Eure Listen an: Rolf jun. Eppinger, Fraunhoferweg 4, 7440 Würtlingen, West Germany, Tel.: 07022/8763

Tausche auf Disk oder Tape! Legt bitte Rückporto bei! CPC 464/ 664/ 6128/ Joyce - Soft\*\*\*\*\*

\* Computer Hard- und Software \* \*  
 \* C-64, C-16, Atari, Schneider, MSX \*  
 \* Alter Ego D 69,00 \*  
 \* Herz von Afrika D 56,00 \*  
 \* Hanse D 48,00 \*  
 \* Golf Construction \*  
 \* Set D 49,90 \*  
 \* Modem, Disketten und vieles \*  
 \* mehr. Gesamtkatalog anfor- \*  
 \* dern. H&S Werner Wohlfahrt- \*  
 \* staetter Postfach 301033, \*  
 \* 4 Düsseldorf \*

\*\*\*\*\*  
 \* **NEBENVERDIENST** \*  
 \* **für CPC-Anwender** \*  
 \* \*\*\*\*\* \*

An alle C16/116 User: Bei uns gibt es selbstgeschriebene Programme schon ab 1 DM! Infos gibt's gegen Freiumschlag bei:

Torsten Gilsbach, Nerastenberger Str. 1, 5788 Winterberg o. Stefan Kramer, Hauptstr. 30, 5788 Winterberg. Und wer's ganz eilig hat, ruft an: 02981/6848 P.S. Clubmitglieder gesucht!!!!

Achtung! 3 brandneue C16/116 Programme (Programmpaket 1)  
 1. Sherlock's Problem  
 ... (Strategiespiel)  
 2. HORSE (Aktionspiel)  
 3. Chath the Mouse  
 Preis: 8.90 DM + 1.10 Porto  
 Lieferung nur gegen Vorkasse!  
 MULISOFT, Hubert Mühlbacher, Plattenberg 3, 8221 Waging

Achtung C 64 User! POKE-LISTE (322) nur 10,-/ NEBENVERDIENST mit dem COMPUTER Tips & Tricks nur 20,-/ Lot-to Tips/ System usw INFO 80PF Adressen von 64 Usern je Stk 2,- od. INFO C-R/ Software Hochheimstr. 13, 5162 Niederzier 2

Verk. C-16+Datsette o. Netzteil DM 140,- sowie Software z. Preis v. DM 130,-. Markus Voss, Breslauer Str. 15, 5910 Kreuztal 1, Tel: 02732/12907

Spiele u. Anwendungsprogramme für den C-16 auf Kassette zu verkaufen. INFOS bei: Dietmar Schwigon, An den Gehren 34 b, 4800 Bielefeld 1

Suche Musikprogramme jeder Art für den Schneider CPC 464 /664. Jörn B. Gruhlke, Gartenstr. 6 2347 Süderbrarup, Tel.: 04641/8396

Bin Einsteiger und suche Kontakt/ Erfahrungsaustausch für plus 4. Wer kann mir helfen C-Programme auf dem C128 zum Laufen zu bringen?  
Lothar Wenzel  
Finkenweg 7a  
6231 Sulzbach

**C-16-Einsteiger.** Sind Einsteiger, suche Kontakt und Erfahrungsaustausch. Peter Thiemann und Frank Lohmann, Am Peterskamp 16, 4703 Bönen

## Suche

VC-20 \* VC-20 \* VC-20 \* VC-20 \* VC-20  
Suche Spitzensoftware für VC-20. Vor allem Datenveraltungen Statistikprogramme, Adventures, (Text+Grafik). Wenn's geht in deutsch. Actionspiele. Die ersten 10 Einsendungen bekommen 3 Flugsimulatoren - habe nur Datas. Michael Speidel, Hauptstr. 97, 7521 Liepoldsheim  
-Zahle sehr gut!

## Kontakte

**HEY BOY'S !!!** Wer sucht einen Tauschkollegen aus der Schweiz? Habe sehr viele Games (abgetippte + Originale). Z.B. Commando....  
Abs.: Marc Trachsel, CH-2543 Lengnau!

**C-16 -Anfänger-**  
Suche entsprechende Kontakte zu C-16 Profis. Wer hilft mir beim Einstieg.  
Th. Maluschka, Bonifaciusstr.17a  
4300 Essen 13

\*\*\*\*\*Hallo C16- Freunde\*\*\*\*\*  
Wer möchte mit mir Programme tauschen(Tape). Habe ca. 120!!  
Schickt Eure Listen an Karin Henk, Breslauerstr.4, 5180 Eschweiler.  
P.S. Verkaufe auch Netzstecker für 25,- DM und Antennenkabel 15,-DM(neu). Nur einmal vorhanden.  
\*\*\*\*\*Bis bald!\*\*\*\*\*

\*\*\* C-16&116&plus 4-Club \*\*\*  
Wer möchte einsteigen?  
Info gratis bei:  
Torben Sofa  
Asperheide 4a  
4902 Bad Salzuffen 5

VC-20 Suche noch Mitglieder für VC-20 Spitzenclub zum Softwaretausch Wie Flugzeug-, Hubschrauber-, Space Shuttle Simulationen, Meinungsaustausch, Tips und Tricks usw.. Adventures und viele Actionspiele. Also ran an die Stifte und schreibt an Michael Speidel  
Hauptstr.97  
7521 Liedolsheim - es lohnt sich!!

# Das nächste Heft erscheint am 20. 10. 86

C-16 \* CPC \* C-64 \* MSX \* ATARI \* VC-20 \* C-128 \* SOFTWARE \* JOYSTICKS \* C-16

\*  
F  
A  
R  
B  
B  
Ä  
N  
D  
E  
R  
\*  
C  
-  
1  
6  
\*  
C  
P  
C  
\*  
C-64 \* MSX \* C-128 \* ATARI \* VC-20 \* DISKETTEN \* JOYSTICKS \* FARBBÄNDER \* SOFT

## ICH GLAUBE, ICH WERDE BLOND! SOFTWARE EN MASSE BEI LINDENSCHMIDT!!!

### C-64/C-128

### CPC

	CASS.	DISK	CASS.	DISK (3")
YIE AR-KUNG FU (TOP-GRAFIK)	DM 37,95	DM 47,95	DM 37,95	—
STARSHIP ANDROMEDA	DM 37,95	DM 47,95	—	—
TAU CETI (ROBOTERREVOLTE)	—	—	DM 33,95	DM 47,95
HANSE (WIRTSCHAFTSSIM.)	DM 37,95	DM 58,95	DM 44,75	DM 67,95
BOMB JACK (SUPERARCADE)	DM 37,95	DM 57,95	DM 37,95	DM 57,95
INTERNATIONAL KARATE	DM 23,75	DM 38,95	—	—

### C-16 SUPERSPIELE AUF CASSETTE

STREET OLYMPICS	DM 9,95	GOLF	DM 9,95
FAVOURITE FOUR	DM 26,95	AUTOBAHN	DM 9,95
FRANK BRUNO'S BOXING	DM 29,95	SUPER HITS	DM 27,95

EINFACH BESTELLEN! LIEFERUNG ERFOLGT FREI HAUS! KEINE MINDESTBESTELLMENGEN. KEINE ZUSÄTZLICHEN PORTOKOSTEN.

KEINE NACHNAHMEGEBÜHR! FORDERT UNSERE KOSTENLOSEN INFOS AN (COMPUTERTYP ANGEBEN).

UMFANGREICHEN KATALOG GEGEN DM 2,- SCHUTZGEBÜHR.

VERSANDHANDEL R. LINDENSCHMIDT, SCHULSTR. 14, POSTFACH 1328, 4972 LÖHNE 2, TELEFON 05732-72849

9.00 BIS 11.30 UHR UND 12.30 BIS 17.00 UHR

C-64 \* MSX \* C-128 \* ATARI \* VC-20 \* DISKETTEN \* JOYSTICKS \* FARBBÄNDER \* SOFT



# Das erste und einzige SOFTWARE-MAGAZIN

Die aktuelle Ausgabe gibt's jetzt überall

**AS Maktueler Software Markt**  
Nr. 7 Sept./Okt. 1986 öS 50 sfr 6.- DM 6.-

100 Seiten Programme im Test  
Der Software-Markt  
auf einen Blick

**Tolle Wettbewerbe!**  
Jede Menge Spiele zu gewinnen, u. a. Martech's Samantha Fox Strip Poker!  
Original Autogramm-Poster von Sam Fox zu gewinnen!  
Ergebnisse der großen ASM-Leserumfrage

Aktion-Games + Adventures + Anwender-, Sound- und Lernprogramme

**HOTLINE** Hitparade  
Die Software-  
immer beibehalten die ASM Secret-  
Service-Seite  
NEU! Feedback - die  
Leserseite

Die umfangreiche Information des Software-Marktes

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