

monatlich

Compute mit

COMMODORE & SCHNEIDER

1/87

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

3,80 DM
33 öS
3,80 sFr

Unabhängiges Magazin für Anwender von Commodore- und Schneider-Computern

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

VC 20:

**Torch
Bomberman**

C 16/116:

**The Champ
Pyramid**

C 64:

**Rylan
Ice Racing**

Schneider:

**Superbasic
Star Trek**



**Leser- + Meckerecke - Report - Kurse
Softwareservice - Tips & Tricks
★ Großer Kleinanzeigenmarkt ★**

EDITORIAL

Lieber Leser!

Hurry, hurry up! Vor Weihnachten herrscht für uns in der Redaktion wie jedes Jahr ungeheurer Zeitmangel. Nicht nur, daß der Dezember weniger Arbeitstage bietet (eigentlich ein Glück), sondern der Arbeitsaufwand, die Messen, viele neue Produkte usw. halten einen ständig in Atem. Dies gilt besonders für einen anderen Titel aus unserem Verlag, den "Aktuellen Software Markt", denn natürlich haben wir vor Weihnachten eine riesige Software-schwemme. "Compute mit" trifft das nicht so stark, da es hier vor allem darauf ankommt, Listings zu veröffentlichen, die ein beständiges oder sogar steigendes Niveau vorweisen können.

Wir sind mittlerweile an dem Punkt, daß wir stolz auf ein hohes Programmniveau verweisen können. Für einige von Ihnen, bringt das leider nicht nur Vorteile: In unseren Regalen stapeln sich bei Atari, C 16 und C64 massenhaft gute Programme, die auf den Abdruck warten. Programmautoren unter unseren Lesern müssen sich daher auf längere Wartezeiten bei der Bearbeitung ihrer Programme in unserem Verlag einstellen.



Uwe Knierim
Chefredakteur

Report

Software Review 5

Rubriken

Klartext für Commodore 3
 Softwareservice 32
 Leser- & Meckerecke 34
 Assemblerkurs 40
 Interruptprogrammierung des C 64.... 42
 Kleinanzeigen 61

IMPRESSUM

"Compute mit"
 ISSN 0179-6720
 erscheint monatlich im
 Tronic-Verlag, Am Stad 35
 3440 Eschwege
 Tel.: 05651/3 00 11



Redaktion:

Axel Credé (verantwortlich)
 Chefredakteur: Uwe Knierim
 Redakteure: Frank Beall, Gertfried Schmidt, Thomas Brandt, Martina Strack

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 Buchhandlung)
 sowie Österreich und Schweiz:
 Verlagsunion
 6200 Wiesbaden
 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, Erfassung in Datenverarbeitungsanlagen usw.) bedürfen der schriftlichen Genehmigung des Verlags. Alle veröffentlichte Software wurde von Mitarbeitern des Verlages oder von freien Mitarbeitern erstellt.

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

Software

Commodore

Torch (VC 20)	8
Bombberman (VC 20 + 3k)	9
Note Screen (C 16/116)	11
Champ (C 16/116)	14
Checksummer (C 16/116)	18
Checksummer (C 64)	20
Pyramid (C 16/116)	21
Turbo-Disc (C 64)	24
IRQ-Routine (C 64)	26
Rylan (C 64)	26
Ice Racing (C 64)	35

Schneider

Electric Gladiators	44
Bildschirmtoolbox	46
Suprabasic, Spritegenerator,	
Madgoon's Race	47
Star Trek	54
Checksummer	60

Bezugspreis:

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

Programmierabteilung:

Montag - Freitag von 14 - 16 Uhr
 Tel. (05651) 30013

Autoren, Manuskripte:

Der Verlag nimmt Manuskripte zur Veröffentlichung gerne entgegen.

Sollte keine andere Vereinbarung getroffen sein, so geben 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 Wendt

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

Das Sonderheft für alle C-16/116 und plus 4 User!

SA 1/87

65 55 sfr 6,50 DM 6,50

Compute mit

C-16

Heimcomputer

C-116/plus 4

Sonderheft 1/87

Diesmal:

Wettbewerb
plus 4 & Floppy
und viele andere Preise
zu gewinnen!

Sound & Grafik:
Alles über den
TED-CHIP!

TOPGAMES
in Maschinensprache

Jede Menge
Utilities!

Wichtige Systemroutinen
und
vieles mehr.

Turbo Tape

Die Software dazu:

Sonderheft 1/87:

CSOK 1/2 Kassette DM 35,-
CSOD 1/2 Diskette DM 35,-

Datamaker, Auto-Old, Double
Screen, 2xLaufschrift, Screen
Editor, Kundenrechnungen,
Riesenlabyrinth, Crazy Worms,
Convoy, Rich Freddy

Sonderheft 1/86: Heft ausverkauft*

CSOK 1 Kassette DM 30,-
CSOD 1 Diskette DM 30,-

Tips&Tricks, Windowmanager, Cowboy-Duell, Mover,
Fred-Man, Jump-Man, Poker, Turmspringern, TankWars,
Rocketman

Fotokopien der Anleitungen gegen DM 8,50

Die alten Ausgaben:

Sonderheft 2/86: Heft DM 6,50

CSOK 2 Kassette DM 30,-
CSOD 2 Diskette DM 30,-

Tetraprogramm, Haushaltstesten, Kay-Outter, Soundmaschine, Pac-
Mac, Jäger des verlorenen Schatzes, Oil of Texas, Mini, Jäger der
verlorenen Mitte, Stern, Stern-Berating u.a.

Sonderheft 3/86: Heft DM 6,50

CSOK 3 Kassette DM 30,-
CSOD 3 Diskette DM 30,-

Genesis-Assembler, Musikcompiler, Witch, Super-
Break-Out, Indiana Joe, MasterMind, ROM (Risiko),
Jump, Spionagelehrling, Monsterhunter, Tips&Tricks

* Bitte beachten Sie: Die Anleitungen zu den Programmen stehen in den Heften. Auch diese können nachbestellt werden. Preis: DM 6,50
Tronic Verlag, Postfach 870, 3440 Eschwege, Telefon (05651) 30011

Genesis-Super-Assembler

Der große Erfolg unseres Assemblers für den C-16 hat uns bewogen, dieses nützliche und für jeden C-16-Programmierer unbedingt notwendige Utility ab sofort auch separat zu vertreiben. Der komplette Assembler wird inclusive Demoprogramm und ausführlicher Anleitung entweder auf Diskette oder Tape geliefert. Tatsächlich ist GENESIS momentan der leistungsstärkste Assembler für den C-16/116/plus 4 auf dem gesamten Softwaremarkt. Ein ganze Reihe von Features erleichtern dabei das Pro-

grammieren in Maschinensprache ungemein. Das besondere Bonbon für die Benutzer von nicht erweiterten Systemen C-16/116 ist die Möglichkeit, Quelltexte, die in Bearbeitung sind, auszulagern (auf Diskette oder Tape), um so wertvollen Speicherplatz zu sparen.

Genesis-Super-Assembler kostet 15,- DM sowohl für Kasette wie auch Diskette und ist beim Tronic-Verlag, PF 870, 3440 Eschwege (Tel.:05651/30011) zu beziehen.

Justieren und Reinigen

Programm: Macic Disk Kit

System: C64

Preis: 69 DM Diskette

Hersteller: Robtek Limited, Unit 4, Isleworth Business Complex St Johns Road, Isleworth, Middlesex, TW7 6NL

Vertrieb: Golden Games

Wer die Commodore 1541 kennt, kennt aus die Probleme die nach einiger Zeit auftauchen können. Je nachdem wie oft man das Laufwerk nutzt, verstellt sich die Magnetkopf Ausrichtung des Laufwerkes mehr oder weniger. Dies Toleranzen sind oft so gering das nur selten ein Ladefehler entsteht, allerdings können einige Kopierschutzmethoden ärgerlich auf diese dejustierung reagieren. **MAGIC DISK KIT** ist nun ein Utility welches diesem Problem entgegen treten soll. Dieses Programm soll Softwaremäßig kontrollieren in wie weit der Magnetkopf verstellt oder die Geschwindigkeit des Laufwerkes vom Standard abweicht. Dieses Programm wird zusammen mit einer **"REINIGUNGSDISKETTE"** und einer ausführlichen deutschen Anleitung geliefert. Nach dem laden des Programmes erscheint zuerst eine wunderschöne Grafik des Floppylaufwerkes danach erscheint das Hauptmenü des Programmes:

F1 Gerätenummer
F3 Geschwindigkeits Test
F5 Justage Test
F7 Reinigungs Programm
F8 ENDE

Um das Programm in seinen Messergebnissen zu prüfen, ließen wir 4 verschiedene Floppylaufwerke durch das Programm untersuchen. Das Ergebnis war sehr überraschend: Nur ein einziges Laufwerk wurde im Geschwindigkeitstest mit dem Wort "GUT" bezeichnet. Alle 3 anderen Laufwerke schnitten mehr oder weniger schlecht ab. Interessant ist auch, daß die grafisch ansprechende Messkurve nicht mit den Messergebnissen übereinstimmt, gerade bei dem Laufwerk welches als gut bezeichnet wurde, zeigt die Messkurve das schlechteste Ergebnis. Offensichtlich ist hier das Programm noch nicht so ganz ausgereift. Überrascht waren wir auch deshalb, weil wir bisher mit keinem der vier Laufwerke Probleme hatten, aber vielleicht liegen wir immer noch unter der Toleranzgrenze. Übrigens der Messvorgang beruht offensichtlich darauf, das die Diskette mit einem speziellen Messfile beschrieben wird, was während der Messung in bestimmten Zeitabschnitten untersucht wird. Aus diesem Grund darf die Diskette auch nach dem Laden nicht aus dem Laufwerk entfernt werden. Dies gilt genauso für den Justage-Test. Auch dieser Test wird durch eine ansprechende Grafik gestaltet. Allerdings hat man hier auf den ersten Blick etwas Schwierigkeiten das Messergebnis zu interpretieren. Ein weiterer Blick in das Handbuch sagte uns jedoch bald, das unser Laufwerk völlig

Dateiverwaltung auf dem C16

Programm: Datei C16

System: C16

Preis: 29,90

Hersteller: CSJ Computersoft Jonnigk, An der Tiefenriede 27, 3000 Hannover 1

Ob es sinnvoll ist oder nicht, jedenfalls existieren auch Dateiprogramme auf dem C16. Wir waren gespannt, was ein solches Programm auf einem Rechner leistet, welcher nur eine maximale Speicherkapazität von ca 12000 Bytes besitzt.

Das Programm "DATEI C16" von CSJ arbeitet noch dazu nur auf Kasette, was eine relative (auch RANDOM genannt) Dateiverwaltung ausschließt. Da nach dem Laden des Programmes lediglich knappe 6 KBytes Ram zur Verfügung stehen ist die Anzahl der Datensätze stark eingeschränkt. Um auf die Datensätze zugreifen zu können, müssen diese ständig im Speicher gehalten werden. Um eine größere Anzahl von Datensätzen zu verarbeiten, bleibt nur die Möglichkeit mehrere Dateien auf Kasette abzulegen, was aber erhöhten Arbeitsaufwand erfordert.

Die Bedienung sowie die Funktionen des Programmes sind nur auf das Grundsätzliche eines Dateiprogrammes beschränkt. Noch nicht einmal dies, denn eine Sortier-Funktion sucht man ver-

gebens im Hauptmenü. Lediglich einfache Funktionen zum Suchen, Ändern, Eintragen, Abspeichern und Laden von Dateien wurden in das Programm integriert. Die einzelnen Datensätze eines Programmes bestehen immer aus 5 Kriterien, welche allerdings frei gewählt werden können. Die Eingaben beschränken sich auf 20 Zeichen pro Eingabefeld, was sicherlich als mager bezeichnet werden darf. Auch der Suchalgorithmus ist sehr begrenzt, lediglich nach dem ersten Eingabefeld kann eine Datei abgesucht werden.

Alles in allem läßt sich wohl nicht behaupten, das mit dem Programm DATEI C16 umfangreiche Datenmengen verwalten lassen, was wohl mit einem C16 auch undenbar wäre. Das Programm dürfte wohl für den Hobby-Anwender geschrieben sein, welcher seine Videofilme, LP's oder ähnliches per Computer verwalten möchte. Allerdings ist auch hier zu beachten, das die Sammlung nicht zu groß wird. Frank Brall

Positiv:

Einfache Bedienung

Negativ:

Wenig Speicherplatz für Daten
Arbeitet nur mit Kasette
Keine Sortierfunktion

stellt sei. Nun gut dachten wir, legen wir mal die Reinigungsdiskette ein und wählen den entsprechenden "Waschgang". Eine riesige Analog-Uhr zeigte wenige Sekunden danach an, das unser "Waschprogramm" im vollen Gange war. Als wir nun den Justage Test ein zweites mal durchführten, sah das Ergebnis tatsächlich besser aus, allerdings war laut Dokumentation das Laufwerk immer noch stark dejustiert. Obwohl im der Dokumentation zu Macic Disk auch erklärt wird, wie man das Laufwerk wieder mit Schraubendreher und etwas Fingerspitzengefühl in Gang setzt, haben wir uns entschlossen, unser Laufwerk verstellt zu lassen, denn schließlich hatten wir bisher nicht die geringsten Probleme. Frank Brall

Positiv:

Einfache Handhabung
Deutsche Dokumentation

Negativ:

Messkurve stimmt nicht immer mit Messwerten überein
Relativ hoher Preis

Programm- und Listingfragen beantwortet unser Hotline-service Montag bis Freitag 14 bis 16 Uhr Tel. (05651) 30013

Floppy-Jockey im Wunderland...

Programm: Eis und Feuer, **System:** C-64, **Preis:** 79,95, **Hersteller:** Euro Gold, **Vertrieb:** Rushware, Kaarst.

3 Jahre hat es gebraucht, bis dieses deutsche Adventure das Licht der Welt bei Euro Gold erblickt hat. So lange war nämlich die Erstellungszeit für Eis und Feuer. Ganz klar, daß ich auf dieses Spiel ganz besonders gespannt war.

Nach dem Öffnen der gut gestalteten Packung fielen mir gleich 3 Disketten entgegen, die jeweils beidseitig bespielt sind. Welch ein Aufwand! Zunächst muß die Hauptdiskette geladen werden und ich sah alle positiven Erwartungen von diesem Mammutwerk zunächst bestätigt. Während des Ladens ertönt ein flotter Sound, der wirklich hörenswert ist und die an der Erstellung des Adventures Beteiligten werden in einem Vorspann, ähnlich wie in Kino-Filmen, gezeigt.

Auch der Einstieg in das Adventure ist gut gelungen: Man hat eine "Mental-Reise" in eine andere Zeit gebucht. Nachdem ich über die Gefahren aufgeklärt wurde (unkontrollierbar) stimmte ich dennoch der Durchführung zu. Dann kommt das Kommando: "Bitte Diskette B einlegen". Wird gemacht! Ich befinde mich also auf einem Waldweg und sehe einen Wege- weiser. Klar daß ich den lesen will! Leider gibt mir der Computer zur Antwort, daß das unmöglich ist. Bin ich blind? Wie dem auch sei, jedenfalls merke ich sehr bald, daß man sich mit dem Text des Adventures sehr viel Mühe gegeben hat. Der Computer gibt mir nicht nur an, daß er einen bestimmten Befehl nicht verwerten kann, bestimmte Worte fehlen auch in seinem Befehlssatz oder er sagt mir z.B. "Suche Dir einen anderen Weg aus", wenn er meint: "Es gibt keinen Weg nach Westen". Auch das sagt er ab und zu. Die Antworten variieren also recht nett, was aber leider nichts daran ändert, daß sein Wortschatz wohl recht dürftig ist.

Nach etwas Suchen finde ich also ein Schloß mit sperrangelweit offener Tür, vor dem nur ein paar Wachen ihren Mittagsschlaf halten. Ist ja prima, denke ich mir und gebe ein: "betrete Schloß" ("gehe" versteht er nämlich nicht). Ganz niedergeschmettert erfahre ich, daß das "unmöglich" sei. Na, ganz so unmöglich kann das doch gar nicht sein, sonst wäre das Adventure kein Adventure, dachte ich mir dann und probierte diverse Satzstellungen aus neben anderen Möglichkeiten, wie z.B. zunächst die Schwelle zu betreten. Nichts klappt!

O.K., dumm wie ich nun einmal zu sein scheine, verlasse ich mich auf die in der Begleitinformationen angebotenen Hilfen, derer es zwei Sorten gibt: Zunächst wäre da der Systembefehl "hint". Systembefehle ändern nichts am Stand des Adventures sondern sollen den Bedienungskomfort erhöhen. Von Idee und Umfang recht brauchbar, helfen mir diese jedoch auch nicht weiter. Unter "hint" erfahre ich, daß ich die gute Gelegenheit ja mal nutzen könnte, mir das Schloß anzusehen. Ich könnte ja versuchen an den Wachen vorbeizuschleichen (was das Programm übrigens nicht kennt). Die zweite Möglichkeit besteht in der Nutzung eines weiteren Systembefehls, der laut Anleitung aber nur als "letzte Rettung" gedacht sein soll.

Ich bin gerne bereit, mich retten zu lassen und dafür auch die geforderte Leistung zu bringen. Nach Eingabe des "Clue"-Befehls erscheint nämlich eine Zeile mit Buchstaben, die nach den Vorgaben in der Anleitung "decodiert" werden müssen (eine Heidenarbeit!). So war ich denn auch gespannt, jetzt des Rätsels Lösung zu erhalten. Um so größer die Enttäuschung als ich erfahre, daß ich das Schloß betreten soll (wörtlich!), die Wachen schlafen lassen und mich selbst offen und ehrlich zeigen soll. Da lachen ja die Hühner!
Von wegen "Schloß betreten" war ja'n Witz! Na dann laufe ich

eben noch eine Runde durch den Wald, sehe abgeknickte Bäume, Zwerge, die ich nicht ansprechen kann, eine Kirche, die ich nicht betreten kann und so fort. Allerdings ist mir in der Zwischenzeit

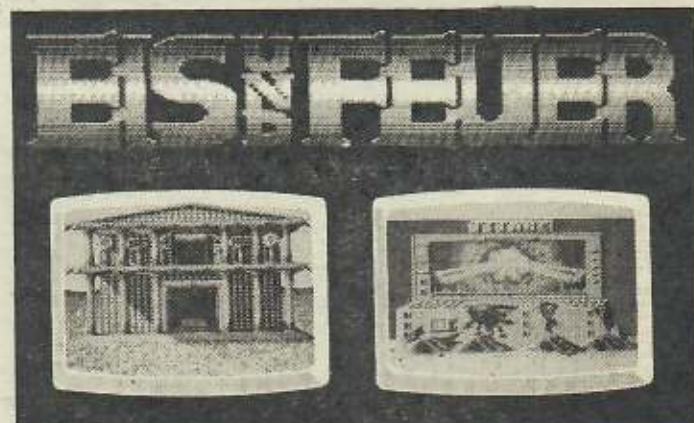
ein weiterer Systembefehl aufgefallen, nämlich "Pic on". Damit schaltet man die Bilder in das Adventure ein, die ich schon vermißt hatte. Die Grafik ist wirklich fantastisch!
Martina Strack

Die vorliegende
auszugsweise Besprechung
stammt aus unserem
Tronic-Magazin

»Aktueller Software-Markt«

Den vollständigen Bericht,
die aktuellsten Bildschirmfotos
finden Sie in der
neuesten Ausgabe ab...

...22.12.1986



Maschinensprache mit AMSMONIX

Programm: Amsmonix
System: CPC 464,664 und 6128
Preis: Kass. 49 DM, Disk. 59 DM
Hersteller/Vertrieb: R. Schuster Electronic, Obere Münsterstr. 33, 4620 Castrop-Rauxel

Programmieren in Assembler! Sicherlich kein neues Thema, jedoch gibt es immer wieder neue Hilfsprogramme, Monitor- oder Assembler-Programme auf dem Markt. Geräte beim Schneider zeigen sich sehr oft Unterschiede bei den Leistungen eines Maschinensprachemonitors. Durch die twas komplizierte RAM/ROM Struktur dieses Rechners lassen viele Programme nur das Editieren bzw. Bearbeiten von ROM oder RAM zu, selten können ROM und RAM gleichzeitig bearbeitet werden. Mit AMSMONIX läßt sich nun gerade dieser Nachteil umgehen. Um Ihnen einen Überblick über die Leistungen dieses Maschinensprachemonitors zu geben, habe ich die einzelnen Befehle kurz erläutert:

A Speicherinhalt ändern
 Hiermit läßt sich der Inhalt einer Speicherstelle (RAM) ändern. Der augenblickliche Wert der Speicherstelle wird angezeigt und kann durch einen anderen Wert überschrieben werden.

B Bildschirm löschen
 Zur besseren Übersicht läßt sich hiermit der Bildschirm löschen.

C Call
 Mit diesem Befehl können Sie ein Maschinenprogramm aufrufen. Trifft das Programm auf ein "RETURN" so wird in den Monitor verzweigt, und die entsprechenden Registerinhalte angezeigt.

D Disassembler
 Dies ist wohl der wichtigste und leistungsstärkste Befehl von AMSMONIX. Durch ihn werden Maschinensprache in disassemblierter Schreibweise ausgegeben. Es werden 24 Zeilen dargestellt, durch einen beliebigen Tastendruck werden die nächsten 24 Zeilen dargestellt. Durch die Taste ESC läßt sich jederzeit das Disassemblieren beenden.

F Fill
 Ein bestimmter Speicherbereich wird mit einem Wert gefüllt.

G GOTO
 Sprung in ein Maschinenprogramm.

H Hilfsmenü
 Dieser Befehl gibt eine Übersicht über alle Funktionen von AMSMONIX.

I Catalog Cass/Disc
 Dieser Befehl erklärt sich wohl selbst

K Kopieren
 Beliebige große Speicherblöcke können in einen anderen Bereich kopiert werden.

M Monitor
 Mit diesem leistungsstarken Befehl lassen sich Hexdumps von RAM oder ROM ausgeben. Neben den Hexzahlen können auch ASCII-Zeichen und Prüfsummen mit Ausgegeben werden.

P Druckerprotokoll
 Drucker wird ein bzw. ausgeschaltet.

R Binaer-Datei einlesen
 Dieser Befehl zeigt Ihnen die Anfangsadresse und Länge eines Binär-Files auf dem Bildschirm an. Danach läßt sich die Datei ab einer beliebigen Adresse einladen.

S Suchen
 Eine beliebige Bytefolge kann im Speicher gesucht werden. Alle Adressen mit der gesuchte Bytefolge werden auf dem Bildschirm ausgegeben.

T Text eingeben
 Mit diesem Befehl läßt sich auf komfortable Weise ein ASCII-Text eingeben, ohne ständig die ASCII-Tabelle vor sich zu haben.

W Binär-Datei speichern
 Ein bestimmter Speicherbereich kann als Binär-Datei abgelegt werden.

X Medium-Wahl
 Das Speicher-Medium Kasette oder Diskette kann gewählt werden.

Z Register-Anzeige
 Durch das Drücken der Taste Z wird der Inhalt aller Register ausgegeben.

Der Assembler Profi wird bei der obigen Aufstellung der Befehle sicherlich keine Besonderheiten feststellen, denn es stehen schon eine ganze Reihe von Programmen zur Verfügung, welche ähnliches leisten. Lediglich die Möglichkeit neben RAM auch ROM-Bänke zu adressieren, hebt den Monitor von der breiten Masse ab. Leider vermisst man dafür bei AMSMONIX einen LINE-ASSEMBLER zum schnellen Eingeben oder Verbessern von kleinen Maschinenprogrammen.

Für den Maschinensprache Anfänger welche lediglich Programme und Speicher des CPC untersuchen möchten, dürfte dieses Programm allerdings völlig ausreichen. Für Assembler-Programmierer bietet der Hersteller übrigens noch ein weiteres Programm, welches wir in einer der nächsten Ausgaben vorstellen werden.

Frank Brall

Positiv:
 Einfache Bedienung
 RAM und ROM-Bänke lassen sich auslisten
 Laden und Speichern auf Kasette/Diskette möglich
Negativ:
 Kein Line-Assembler

Hardcopy auf Bildschirm oder Drucker

Programm: Copyshop
System: CPC464/664/6128
Preis: 59 DM
Hersteller/Vertrieb: DMV Daten & Medien GmbH, Postfach 250, 3440 Eschwege

Obwohl dieses Programm eigentlich ausschließlich dafür entwickelt wurde, Bildschirmgrafiken auf das Papier zu bringen, unterscheidet es sich doch in der Anzahl der möglichen Optionen positiv von zahlreichen Konkurrenzprodukten. Neben der Möglichkeit, Bilder aller drei Schneider MODE's zu laden, speichern und darzustellen, können diese auch noch mit einem Grafikeditor verändert werden. Dieser Editor enthält zahlreiche Funktionen wie FILL, INVERTIERUNG und ähnliches. Zwar kann man diesen Editor nicht mit einem Grafikprogramm vergleichen, aber für eine Beschriftung oder eine kleine Modifikation des Bildes reicht es allemal aus. Besonders interessant und offensichtlich bisher auch Einmalig ist die Möglichkeit, über einen einzigen Tastendruck, eine Hardcopy auf dem Bildschirm auszugeben. Dabei werden die Farben der Originalgrafik in entsprechende Schwarz/Weiß Rasterungen umgesetzt. Das ganze dauert etwa 1-3 Sekunden bis die originalgetreue Bildschirm-Hardcopy auf der Mattscheibe erscheint. Das Aus-

drucken dauert da schon etwas länger, allerdings dürfte COPYSHOP auch hier zu den schnellsten Hardcopy-Programmen auf dem Markt gehören. In weniger als 4 Minuten wird eine komplette DIN A4 Seite bedruckt. COPYSHOP erlaubt den Benutzer das Hardcopy in 4 verschiedenen Größen DIN A4, DIN A5 und 2 kleineren Maßen auf das Papier zu bringen. Dabei kein frei gewählt werden welcher Bildausschnitt ausgegeben werden soll. Eine weitere Einstellmöglichkeit bezieht in der Wahl verschiedener RASTER. Praktisch jede Farbe läßt sich ein andere Raster zuteilen, welches später bei ausdrucken des Bildes benutzt wird. Ein spezielles Menü zur Druckerranpassung sorgt dafür das Sie praktisch jeden beliebigen Drucker mit dem Programm ansteuern können. Komfortable Pull-Down Menüs erlauben gute Übersicht aller Funktionen und einfache Bedienung des Programmes.

Neben COPYSHOP befindet sich auch noch zwei weitere kurze Maschinenprogramme auf der Diskette. Eines dieser Programme erlaubt über RSX-Befehle Hardcopies auf den Drucker auszugeben. Das andere Programm stellt eine Art "SCREEN-FREEZER" da. Über Tastendruck läßt sich damit die Bildschirmgrafik eines Spieles auf Diskette ablegen.

VC-20-Grundversion

TORCH

Versetzen Sie sich mal zurück in die Steinzeit. Sie sind der Neandertaler Roby und haben soeben das Feuer entdeckt. Aber allein haben Sie nichts davon. Also nehmen Sie sich ein Stück Holz und rennen zu ihrem Stamm. Aber Vorsicht!!! Auf Ihrem Weg lauern die teuflischen Geister aus der Unterwelt. Berührt Sie ein Geist, so sind Sie in die Hölle verdammt. Außerdem brennt die Fackel nicht ewig! Ihr Ziel ist es, Ihre Fackel am rechten Bildschirmrand zu erreichen, um dadurch das Feuer zu erhalten.

Die Geister können nicht zurück, wenn Sie einmal unten angekommen sind. Also lassen Sie möglichst viele Geister in den Abgrund stürzen, dies gibt nämlich 30 Punkte.

Haben Sie die Fackel erreicht, geht es auf den nächsten Weg. Auf jedem Weg aber kommt ein Geist hinzu (bis zum fünften), wodurch das Durchkommen natürlich immer schwieriger wird.

Ab dem fünften Weg denkt sich der Teufel etwas neues aus:

»Wenn ich diesen Typ mit meinen Geistern nicht einschüchtern kann, laß ich einfach von meinen Geistern 'Pseudo-Geister' hinstellen!« Diese Pseudo-Geister können Sie zwar berühren aber so nach und nach wird es immer schwieriger, echte Geister und Pseudo-Geistern zu unterscheiden.

Im übrigen bekommen Sie pro geschaffter Wegstrecke einen Bonus von 500 Punkten!!!

Das Spiel wird mit Joystick gespielt und ist für die Grundversion!!!

Haben Sie eine 3-kB-Erweiterung können Sie die beiden Teile zusammenfügen, so daß daraus ein Programm wird. Die Zeilennummern sind dafür vorgesehen.

Ansonsten muß einmal nachgeladen werden (aber erst starten). Und nun: Viel Glück!!!

Teil 1

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

1 REM ***** <128>
2 REM * COPYRIGHT * <249>
3 REM * BY * <226>
4 REM * TIMM HAASE * <204>
5 REM * UND * <111>
6 REM *BJOERN BROEGER* <209>
7 REM ***** <134>
8 : <66>
9 : <67>
10 PRINT " (CLEAR) ":PRINTCHR$(14)CHR$(8);POKE36878
,15 <147>
11 POKE51,0;POKE52,28;POKE55,0;POKE56,28;CLR:CB=
7168 <88>
12 READA:IFA=-1THENPOKE36869,255;GOTO15 <175>
13 FORN=0TO7:READB:POKECB+A*N+B,N;NEXT <178>
14 GOTO12 <143>
15 POKE36879,25;PRINT" (RED CLEAR DOWN3) " <17>
16 PRINT" (SPACE) [( (SPACE) [( (SPACE) [( (SPACE) [( (
(SPACE) [( (SPACE) [( <71>
17 PRINT" (BLACK SPACE)DDD (SPACE)DDD (SPACE)DDD (SP
ACE)DDD (SPACE)D (SPACE)D <206>
18 PRINT" (BLACK SPACE2)D (SPACE2)D (SPACE2)D (SPACE2)
D (RED) [( (BLACK)D (SPACE2)D (SPACE3)D (RED) [( (BLACK)D <179>
19 PRINT" (BLACK SPACE2)D (SPACE2)D (SPACE2)D (SPACE2)
DD^ (SPACE)D (SPACE RED) [( (BLACK SPACE)DDD <154>
20 PRINT" (BLACK SPACE2)D (SPACE2)D (RED) [( (BLACK)D (
SPACE)D (SPACE)D (SPACE)D (RED) [( (BLACK)D (SPACE)D (SP
ACE)D <145>
21 PRINT" (BLACK SPACE2)D (SPACE2)D \D] (SPACE)D (SPAC
E)D (SPACE)D \D] (SPACE)D (SPACE)D <136>
22 PRINT" (DOHNS BLACK RVSON SPACE2) (C) (SPACE) 198
6 (SPACE) BY (SPACE2) ZEUS" <175>
23 READX$:IFX$="*"THEN28 <105>
24 FORI=1TOLEN(X$):PRINT" (HOME DOWN) " <201>
25 PRINT" (HOME DOWN2] RVSON) ";TAB(20)MID$(X$,E,1
) <205>
26 PRINT" (HOME DOWN2] RVSON RIGHT) "CHR$(20);FORV
=1TO10:NEXT:POKE36877,200;POKE36877,0 <224>
27 NEXT:GOTO23 <115>
28 POKE198,0;WAIT198,1;POKE198,0;POKE36869,240 <54>
29 PRINT" (CLEAR BLACK DOWN) SPACE PLEASE (SPACE) L
DAD (SPACE) THE (SPACE) MAIN (DOHNS) " <147>
30 DATA0,16,56,56,120,253,255,255,255 <17>
31 DATA1,0,32,96,224,240,240,248,248 <223>
32 DATA2,0,0,0,0,1,3,7,1 <14>

```

```

33 DATA3,3,35,51,255,255,255,255,255 <97>
34 DATA4,255,255,255,255,255,255,255,255 <37>
35 DATA5,252,254,254,255,255,255,255,255 <254>
36 DATA6,0,0,0,192,248,240,224,240 <58>
37 DATA7,228,248,255,255,255,255,255,255 <70>
38 DATA8,231,253,255,255,255,255,255,255 <15>
39 DATA9,0,1,9,27,31,127,255 <131>
40 DATA10,0,0,192,236,253,254,255,255 <41>
41 DATA11,7,15,31,127,255,251,251,63 <67>
42 DATA12,132,207,255,255,255,247,243,255 <123>
43 DATA13,224,244,248,252,252,248,254,254 <105>
44 DATA14,31,71,255,255,247,247,127,63 <171>
45 DATA15,255,189,253,255,255,205,223,255 <18>
46 DATA16,252,248,240,248,252,252,252,240 <25>
47 DATA17,31,247,62,12,0,0,0 <41>
48 DATA18,255,255,124,124,254,254,252,254 <195>
49 DATA19,240,0,0,0,0,0,0,0 <123>
50 DATA20,254,255,255,127,126,254,255,255 <126>
51 DATA21,0,60,127,118,126,124,60,56 <65>
52 DATA22,56,60,124,82,220,231,120,126 <48>
53 DATA23,60,60,54,118,110,110,111,55 <195>
54 DATA24,60,60,60,60,60,60,30,31 <25>
55 DATA25,16,56,120,112,192,128,0,0 <20>
56 DATA26,60,126,86,126,99,63,63,118 <197>
57 DATA27,2,34,54,55,127,255,255,255 <106>
58 DATA28,255,255,127,127,63,63,15,3 <95>
59 DATA29,255,255,254,254,252,252,240,192 <178>
60 DATA30,255,254,252,240,192,240,252,255 <100>
61 DATA32,0,0,0,0,0,0,0,0 <220>
62 DATA-1 <174>
63 DATA"BEI (SPACE) DEM (SPACE) SPIEL (SPACE) 'TORCH' (
SPACE) GEHT (SPACE) ES (SPACE) DARUM (SPACE) MOEGLICHT
(SPACE) VIELE (SPACE) GEISTER (SPACE) UNTEN" <170>
64 DATA" (SPACE) ANKOMMEN (SPACE) ZU (SPACE) LASSEN. (S
PACE) JEDER (SPACE) GEIST (SPACE) MACHT (SPACE) 30 (SPAC
E) PUNKTE (SPACE) MEHR (SPACE) !!! (SPACE) IN (SPACE) SPA
ETEREN" <139>
65 DATA" (SPACE) LEVELN (SPACE) BLEIBEN (SPACE) BEISTE
R (SPACE) AUF (SPACE) DEM (SPACE) BILDSCHIRM (SPACE) STE
HEN, (SPACE) UM (SPACE) DEN (SPACE) SPIELER (SPACE) ZU" <245>
66 DATA" (SPACE) VERWIRREN. (SPACE) (C) (SPACE) 1986 (S
PACE) BY (SPACE3) TIMM (SPACE) HAASE (SPACE) & (SPACE) BJ
OERN (SPACE) BROEGER (SPACE) ..... (SPACE15) " <26>
67 DATA" (SPACE7) " <158>
68 DATA" * " <252>
ENDE DES LISTINGS

```

Teil 2

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

69 POKE36869,255;PRINTCHR$(8)CHR$(14);BOTO130 <125>
70 DATA7711,7702,7706,7710,7716 <84>
71 FORI=1TO5:READH(I);M(I)=H(I);W(I)=32;NEXT <71>
72 A=1;B=1;C=0;W=32;B=1;GF=8 <144>
73 DATA0,0,1,18,1,5,1,12,12,1,11,9,13,5,11,13,12
,18 <81>
74 FORU=1TO9:READT1(U),T2(U);NEXT:RESTORE <49>
75 POKE36879,B;PRINT" (BLUE) ":POKE36869,255 <163>
76 PRINT" (CLEAR RVSON) SCORE: (SPACE) 0 " <31>
77 PRINT" (HOME DOWN SPACES) @A (SPACE10) IGF <41>
78 PRINT" (SPACE3) BCDEF (SPACE6) I@IDDF <129>
79 PRINT" (SPACE2) IHDDGGJ (SPACE3) @CDDDDDDJ <7>
80 PRINT" (SPACE) IDDDDDDDJICDDDDDDDDJ "; <20>
81 PRINT" IDDDDDDDDDDDDDDDDDDDDD "; <113>
82 A$=" (GREEN) KLM (DOWN LEFT3) NOP (DOWN LEFT3) QRS (
DOWN LEFT2 RED) T (UP LEFT RED) T " <26>
83 PRINT" (GREEN) " <18>
84 FORI=1TO9 <203>
85 PRINT" (HOME) ":FORO=1TOT1(I)+5 <111>
86 PRINTCHR$(17);NEXTO:PRINTTAB(T2(I))A$;NEXT:P
RINT" (HOME DOWN14) " <195>
87 A$(1)=" (YELLOW) U (DOWN LEFT RED) V (PURPLE) Y (DOM
N LEFT2 CYAN) W " <176>
88 A$(2)=" (YELLOW) U (DOWN LEFT RED) V (PURPLE) Y (DOM
N LEFT2 CYAN) X " <201>
89 A$(3)=" (YELLOW SPACE DOWN LEFT RED SPACE DOWN
LEFT CYAN SPACE) " <11>
90 A$(4)=" (YELLOW SPACE DOWN LEFT RED SPACE2 DOM
N LEFT2 CYAN SPACE) " <173>
91 T1$="000000":POKE7987,25;POKE36865,1 <22>
92 REM *GAME* <93>
93 J9=PEEK(37137);POKE37154,127;J8=PEEK(37152);P
OKE37154,255;POKE36876,0;POKE36877,0 <26>
94 IF (J9AND16)=0THENA=A-1;B=1;GOSUB127 <52>

```



```

95 IF (JBAND128)=0 THEN A=A+1: B=2: GOSUB 127
96 C=C+1: IFC=1 THEN PRINT TAB(A) " (UP3) "A*(1)
97 IFC=2 THEN PRINT TAB(A) " (UP3) "A*(2): C=0
98 IFB=2 THEN PRINT TAB(A-1) " (UP3) "A*(3)
99 IFB=1 THEN PRINT TAB(A+1) " (UP3) "A*(4)
100 IFA=20 THEN A=1: PRINT " (HOME DOWN12 RIGHT20 SPA
CE DOWN LEFT SPACE2 DOWN LEFT2 SPACE) ": GOSUB 120
101 IFB=>6 THEN G=5
102 IFTI#="000010" THEN GH=54
103 IFBH=54 THEN GF=GF+1
104 IFGF=1 THEN POKE 36879, 24: GF=GF+1
105 IFGF>2 THEN GF=0
106 IFA<2 THEN A=1: PRINT " (HOME DOWN12 SPACE DOWN L
EFT SPACE DOWN LEFT SPACE) "
107 FOR I=1 TO G: D(I)=INT (RND (1)*3): NEXT
108 FOR I=1 TO G
109 IFD(I)=1 THEN E(I)--1
110 IFD(I)=2 THEN E(I)=1
111 POKEM(I), W(I): M(I)=PEEK (M(I)+E(I)+22): M(I)=M
(I)+E(I)+22: POKEM(I), 26
112 IFM(I)>8100 THEN FORV=1 TO G: POKEM(V), M(V): M(V)=
H(V): POKEM(V), W(V): W(V)=32: S=S+30: NEXT
113 IFPEEK (M(I)+E(I)+22)>20 AND PEEK (M(I)+E(I)+2
2)<26 THEN I41
114 IFPEEK (M(I)+E(I)+22)=21 THEN I41
115 IFPEEK (M(I)+E(I)+22)=22 THEN I41
116 IFPEEK (M(I)+E(I)+22)=23 THEN I41
117 IFPEEK (M(I)+E(I)+22)=24 THEN I41
118 NEXT: IFGF=2 THEN POKE 36879, B: GF=0
119 GOTO 93
120 REM WAHL
121 B=B+1: I=0: V=0: TI#="000000": GH=0: GF=0: S=S+500
122 PRINT " (HOME DOWN13 SPACE RVSON YELLOW) ENTER (
SPACE) NEXT (SPACE) LEVEL (SPACE) !! (SPACE) "
123 FOR PD=1 TO 2500: NEXT
124 PRINT " (HOME DOWN13 SPACE21) "
125 PRINT " (CYAN HOME RVSON) SCORE (SPACE) ": S: POKE
7987, 25: POKE 36885, 1
126 RETURN
127 REM GERAUSCH
128 POKE 36878, 15: POKE 36877, 200: POKE 36877, 0: POKE 3
<92>
<107>
<148>
<1>
<9>
<17>
<174>
<242>
<57>
<156>
<63>
<255>
<9>
<55>
<243>
<237>
<51>
<235>
<120>
<248>
<13>
<34>
<55>
<103>
<11>
<20>
<78>
<63>
<74>
<32>
<229>
<12>
<172>
6878, 0
129 RETURN
130 POKE 36879, 25: PRINT " (CLEAR DOWN4) "
131 PRINT " (RED SPACE) [ ( (SPACE) [ ( (SPACE) [ ( (SPA
CE) [ ( (SPACE) [ (SPACE) [
132 PRINT " (BLACK SPACE) DDD (SPACE) DDD (SPACE) DDD (S
PACE) DDD (SPACE) D (SPACE) D
133 PRINT " (BLACK SPACE2) D (SPACE2) D (SPACE2) D (SPACE
) D (RED) [ (BLACK) D (SPACE) D (SPACE3) D (RED) [ (BLACK) D
134 PRINT " (BLACK SPACE2) D (SPACE2) D (SPACE) D (SPACE
) DD (SPACE) D (SPACE RED) [ (BLACK SPACE) DDD
135 PRINT " (BLACK SPACE2) D (SPACE2) D (RED) [ (BLACK) D
(SPACE) D (SPACE) D (SPACE) D (RED) [ (BLACK) D (SPACE) D (S
PACE) D
136 PRINT " (BLACK SPACE2) D (SPACE2) \DJ (SPACE) D (SPA
CE) D (SPACE) \DJ (SPACE) D (SPACE) D
137 PRINT " (DOWN6 RIGHT2 RVSON BLACK) (C) (SPACE) 19
B6 (SPACE) BY (SPACE2) ZEUS"
138 POKE 36878, 15: FOR L=250 TO 2000 STEP -2: POKE 36876, L
: FORM=1 TO 100: NEXT: NEXT
139 FOR L=205 TO 250 STEP 2: POKE 36876, L: FORM=1 TO 100: N
EXT: NEXT: POKE 36876, 0: POKE 36878, 0
140 POKE 198, 0: WAIT 198, 1: POKE 198, 0: GOTO 70
141 PRINT " (HOME DOWN11 RIGHT5 RVSON GREEN) GAME (S
PACE YELLOW) OVER (SPACE PURPLE) !!! ": FORT=1 TO 5000:
NEXT: RUN
ENDE DES LISTINGS

```



VC-20 + 3k

Bombberman

Bombberman haßt Luftballons. Sein Haß treibt ihn dazu, Bomben zu legen, in der Hoffnung, die verhassten Luftballons mögen durch die Explosion vernichtet werden. Luftballons sind jedoch launische Gesellen. Mal fliegen sie ziellos umher, dann verfolgen sie unseren Bombberman zielstrebig...

Spielbeschreibung

Gesteuert wird mit dem Joystick, wobei der Feuerknopf dem Legen der Bomben dient. Es können maximal 5 Bomben gleichzeitig im Spiel sein. Die kleinen Mauersteine werden von deren Explosion durchgeschlagen - so kann man sich den Weg freimachen. In manchen dieser kleinen Mauersteine befindet sich ein Extraleben, welches durch eine Explosion freigelegt wird. Dieses Extraleben kann aber auch durch eine weitere Explosion oder durch einen Luftballon zerstört werden.

Verliert man ein Leben, weil ein Luftballon Bombberman erreichte oder man in eine Explosion geriet, so beginnt man den jeweiligen Level von vorn. Es gibt 4 verschiedene Level:

1. Normal

2. Kleine Mauersteine erscheinen und verschwinden; Nur allzu leicht tun sich Sackgassen vor einem auf...
3. Alle kleinen Mauersteine blinken; Die Übersicht wird dadurch noch mehr erschwert.
4. Bombberman legt kontinuierlich Bomben, ohne ihr Dazutun; Man muß ständig in Bewegung sein! Zur Warnung sieht man in der obersten Zeile: »AUTO-BOMB!«

Nach jedem vierten Level fängt man wieder mit dem ersten an, wobei jedoch ein Luftballon hinzukommt und die Zeit vom Legen bis zum Zünden der Bomben verringert wird.

Der Bonus, der ebenfalls im oberen Informationsblock angezeigt wird, wird ständig heruntergezählt. Gewinnt man einen Level, so wird der verbleibende Bonus zu der Punktzahl hinzugezählt.

Da dieses Spiel Ihnen auch ein wenig Strategie abverlangt, können Sie es mit F1 unterbrechen und mit F3 fortsetzen. Dadurch bekommt man die Chance, die Übersicht zu gewinnen.

Teil 1

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

0 :GOTO200
1 *****
2 * BOMBERMAN *
3 * WRITTEN BY: *
4 * OLIVER GUMTAU *
5 * KIRCHENWEG 25 *
6 * 2057 REINBEK *
7 * EXKLUSIV FUER D. *
8 * TRONIC-VERLAG *
9 *****
10 INTERRUPT :
12 DATA120,162,34,160,29,142,20,3,140,21,3,88,96
,120,162,191,160,234,142,20,3
14 DATA140,21,3,88,96,198,4,208,59,169,4,133,4,1
62,6,202,202,202,202,202,202
16 DATA189,90,3,133,247
18 DATA189,91,3,133,248,160,0,177,247,168,192,3,
208,2,169,4,192,4,208,2,169,5,192,5
20 DATA208,2,169,6,192,6,208,2,169,3,160,0,145,2
47,224,0,208,203,76,194,234
22 FARBE U. JOYSTICK :
24 DATA169,44,133,251,133,253,169,30,133,252,169
,150,133,254,160,0,177,251,170,189,195
26 DATA29,145,253,230,253,230,251,208,242,230,25
4,230,252,165,252,201,32,208,232
28 DATA24,173,17,145,41,60,133,5,162,127,142,34,
145,173,32,145,162,255
30 DATA142,34,145,41,128,101,5,133,5,96
32 BILDSCHIRM LOESCHEN
34 DATA160,0,169,44,133,251,169,30,133,252,169,1
6,145,251,230,251,208,248,230,252
36 DATA165,252,201,32,208,240,96
38 FARBLISTE
40 DATA2,6,8,4,4,4,4,3,5,5,5,5,5,5,7,4,7,7,7,7

```


programme

98 SONDERZEICHEN

```

100 DATA255,255,255,255,255,255,255,255
101 DATA126,255,255,255,255,255,255,126
102 DATA0,0,60,60,60,60,0,0
103 DATA0,62,107,127,127,62,1,0
104 DATA0,28,42,62,62,28,2,0
105 DATA28,62,42,62,62,28,2,0
106 DATA0,28,42,62,62,28,2,0
107 DATA16,56,76,92,124,124,56,0
108 DATA56,56,0,124,188,56,40,40
109 DATA56,56,0,124,188,56,40,8
110 DATA56,56,0,124,122,56,40,32
111 DATA56,56,0,56,56,24,40,36
112 DATA56,56,0,56,56,24,16,16
113 DATA56,56,0,56,56,48,40,72
114 DATA56,56,0,56,56,48,16,16
115 DATA120,76,76,124,98,98,124
116 DATA0,0,0,0,0,0,0,0
117 DATA187,254,223,247,126,223,253,111
118 DATA185,174,213,89,174,101,219,117
119 DATA37,178,4,81,138,64,18,162
120 DATA1,68,0,18,128,8,64,2
121 DATA0,3,6,4,15,15,15,31
122 DATA224,248,120,60,108,196,228,244
123 DATA17,28,30,31,31,47,112,16
124 DATA252,248,120,112,224,128,0,0
125 DATA7,31,30,60,54,35,39,47
126 DATA0,192,96,32,240,240,216
127 DATA63,31,28,14,7,1,0,0
128 DATA16,182,24,24,56,244,14,8
200 POKE56,25:POKE55,192:POKE52,25:POKE51,192
205 POKE36879,14:PRINT"(CLEAR)":POKE36869,255
210 FORI=7432T07639:READA:POKEI,A:NEXTI
215 FORI=7168T07399:READD:POKEI,D:NEXT
220 FORI=7424T07431:POKEI,0:NEXT
225 PRINT"(CLEAR DOWN4 RIGHT2 PURPLE)UV(RIGHT13)
VZ":PRINT"(RIGHT2)WX(RIGHT2 GREEN RVSON)BOMBERMA
N(RIGHT2 RVSOFF PURPLE)"
230 PRINT"(RVSON DOWN2 RIGHT8 RED)TEIL(SPACE)2":
PRINT"(RVSON DOWN RIGHT5)WIRD(SPACE)GELADEN(BLAC
K)"
235 POKE198,1:POKE631,131
ENDE DES LISTINGS

```

Teil 2

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

0 GOTO100
1 DATA165,197,201,39,208,6,165,197,201,47,208,25
0
2 LUFTBALLONS
3 DATA169,0,133,6
4 DATA162,6,202,202,202,202,202,189,86,3,208
,3,76,16,27,134,9,188,89,3,32,220,27
6 DATA24,105,30,24
8 DATA133,252,152,166,9,125,88,3,133,251,165,252
,105,0,133,252,160,0,177,251,24

```

```

<84> 10 DATA105,239,144,8,169,0,157,86,3,76,16,27,189
,86,3,201,1,208,83,56,189,87,3,233,1
<83> 12 DATA157,87,3,208,10,169,2,157,86,3,169,20,157
<242> ,87,3,169,0,133,2,56,189,88,3,229,0
<225> 14 DATA16,10,169,2,133,2,56,165,0
<186> 16 DATA253,88,3,133,9,169,1,133,3,56,189,89,3,22
<165> 9,1,16,10,169,3,133,3,56,165,1,253
<220> 18 DATA89,3,164,2,56,229,9,48,6,164,3,165,2,133,
<167> 3,24,144,34,56,189,87,3,233,1
<84> 20 DATA157,87,3,208,10,169,1,157,86,3
<74> 22 DATA169,20,157,87,3,134,9,24,32,148,224,165,1
<27> 41,41,3,168,166,9,189,88,3,133,181,189
<204> 24 DATA89,3,133,191,192,0,208,2,198,181,192,1,20
<8> 8,2,198,191,192,2,208,2,230,181
<28> 26 DATA192,3,208,2,230,191,134,9,164,191,32,220,
<106> 27,24,105,30,24,133,254,152,166,9
<130> 28 DATA101,181,133,253,165,254,105,0,133,254,160
<221> ,0,177,253,24,105
<75> 30 DATA239,144,8,169,0,157,86,3,24,144,33,56,177
<69> ,253,233,7,16,22,189,86,3,201,1,208,43
<22> 32 DATA165,3,201,255,240,37,164,3,169,255,133,3,
<66> 76,138,26,177,251,145,253
<110> 34 DATA169,16,145,251,165,181,157,88,3,165,191,1
<174> 57,89,3,165,253,157,90,3,165,254
<113> 36 DATA157,91,3,189,86,3,240,2,230,6,224,0,240,3
<226> ,76,211,25
<153> 37 EXPLOSIONSKONTR.
<148> 38 DATA160,0,169,44,133,251,169,30,133,252,177,2
<172> 51,170,201,17,208,2,162,18,201,18,208,2
<13> 40 DATA162,19,201,19,208,2,162,20,201,20,208,2,1
<172> 62,16,138,145,251,230,251,208,222
<156> 41 DATA230,252,165,252,201,32,208,214
<132> 42 BOMBENKONTOLLE
<52> 43 DATA162,15,202,202,202,189,60,3,240,118,189,6
<100> 1,3
<54> 44 DATA133,251,189,62,3,133,252,160,46,177,251,2
<200> 01,16,208,4,169,7,145,251,56,189,60,3
<73> 46 DATA233,1,157,60,3,208,85,169,17,145,251,138,
<244> 72,162,9,202,202,224,255,240,69
<186> 48 DATA188,78,3,177,251,240,243,201,1,240,239,20
<91> 1,2,208,29,169,18,145,251,134,9,152,72
<73> 49 DATA32,148,224,166,9,104,168,165,141,41,63,20
<23> 8,4,169,15,145,251,202,24,144,20
<172> 50 DATA169,18,145,251,202,188,78,3,177,251,240,8
<96> ,201,1,240,4
<59> 52 DATA169,19,145,251,202,224,255,208,187,104,17
<132> 0,224,0,240,3,76,84,27,76,100,29
<51> 54 REM U.P. Y-REGISTER * 22
<91> 56 DATA169,0,133,98,132,99,162,144,56,32,73,220,
<17> 32,12,220,169,0,133,98,160,22,132,99
<172> 58 DATA162,144,56,32,73,220,32,43,218,32,170,209
<231> ,96
<27> 100 POKE36879,14:PRINT"(CLEAR)":POKE36869,255
<206> 105 FORI=6593T07167:READA:POKEI,A:NEXTI
110 PRINT"(CLEAR DOWN4 RIGHT2 PURPLE)UV(RIGHT13)
VZ":PRINT"(RIGHT2)WX(RIGHT2 GREEN RVSON)BOMBERMA
N(RIGHT2 RVSOFF PURPLE)"
<85> 115 PRINT"(RVSON DOWN2 RIGHT6 RED)HAUPTTEIL"
<222> 120 PRINT"(RVSON DOWN RIGHT5)WIRD(SPACE)GELADEN(
BLACK)"
<182> 125 POKE198,1:POKE631,131
<134> ENDE DES LISTINGS

```

Teil 3

VC20 LISTING + CHECKSUMMEN (V V1.0)

```

0 POKE37877,255:DIMZ(3,1):Z(0,0)=8:Z(0,1)=8:Z(1,
0)=9:Z(1,1)=10:Z(2,0)=11:Z(2,1)=12
1 Z(3,0)=13:Z(3,1)=14:FORI=828T01023:POKEI,0:NEX
T
2 FORI=1T09:READT(I):NEXT:FORI=846T0853:READA:PO
KEI,A:NEXT
3 POKE36879,8:POKE36869,255:TA#="30202220"
4 T#="1414654564546890988664545654221401414654565
446890988664545988668909886645456542214000"
5 L#="231223121131224223122312113122422312231211
312242231223122312242231223121131224244"
6 PRINT"(CLEAR DOWN2 RIGHT2 PURPLE)UV(LEFT2 DOM
N)WX(RIGHT2 GREEN RVSON)BOMBERMAN(RIGHT2 RVSOFF
PURPLE)"

```

```

18 IFSC>H8THENH8=SC
20 PRINT"(YELLOW RVSON DOWN4 RIGHT2)YOUR(SPACE)S
CORE(SPACE)":SC:PRINT"(RVSON DOWN RIGHT2)HIGHSCO
RE(SPACE2)":H8
22 PRINT"(DOWN3 RIGHT2 PURPLE)C(SPACE YELLOW RV
S ON)=(SPACE)50(SPACE)PUNKTE":PRINT"(DOWN RIGHT2 Y
ELLOW)D(SPACE RVSON)=(SPACE)EXTRALEBEN":POKE7467
,6:POKE658,32
24 POKE859,31:SYS7432:PRINT"(RVSON RED DOWN4 RIG
HT6)FEUERKNOPF":T=0:R=1:TA=0:POKE36878,15
26 TA=TA+1:IFTA=9THENTTA=1
28 POKE36877,B0#VAL(MID#(TA#,TA,1)):POKE36878,PE
EK(36878)-3
30 R=R-1:POKE36877,0:IFR>0THENR=36
32 T=T+1:IFT=82THENT=1
34 POKE36875,T(VAL(MID#(T#,T,1))):POKE36878,15:R
=VAL(MID#(L#,T,1))
36 FORI=1T0120:NEXT:IF(PEEK(37137)AND32)=32THENG
OTO26
38 POKE36875,0:SYS7445:T#="":L#="":TA=0:R=0:LE=3
:AL=1:L=0:M=0:Z=8:SC=0
40 L=L+1:M=M+1:IFM<5THEN46

```



```

42 M=1:AL=AL+1:IFAL>25THENAL=25
44 ZZ=ZZ-1:IFZZ=4THENZZ=5
46 TW=0:TN=0:B=500+20*(L-1):POKE6610,AL*6:POKE74
67,AL*6:POKE36878,239
48 FORI=39TO160:POKE36865,I:NEXT
50 PRINT"(CLEAR RVSON YELLOW)LEVEL:";L:PRINT"(RV
SON)BONUS:(WHITE)";B:SYS7524
51 IFM=4THENPRINT"(HOME)"TAB(11)"(RED RVSON)AUTO
-BOMB!"
53 PRINT"(HOME DOWN)"TAB(13)"(GREEN SPACE8)";
55 FORI=7724TO7744:POKEI,0:POKEI+440,0:NEXT:FORI
=7746TO81428STEP22:POKEI,0:POKEI+20,0
60 NEXTI:FORI=0TO8:FORJ=0TO8:POKE7770+I*2+J*44,1
:NEXTJ,I
64 FORI=1TO90+(M=4)*20
65 Y=INT(RND(1)*19)+J=0:IFY/2=INT(Y/2)THENJ=1
66 X=J+2*INT(RND(1)*(10-J)+J):IFX>18THEN65
67 POKE7747+X+Y*22,2:NEXTI
70 FORI=0TOAL-1:POKE854+I*6,INT(RND(1)*2+1):POKE
855+I*6,INT(RND(1)*20+1)
72 X=(18*INT(RND(1)*2))+1:Y=INT(RND(1)*18+3):IFP
EEK(7680+X+22*Y)<>16THEN72
74 POKE7680+X+22*Y,4:POKE854+I*6,X:POKE857+I*6,Y
75 POKE858+I*6,(7680+X+22*Y)AND255:POKE859+I*6,(
7680+X+22*Y)/256:NEXTI:SYS7432
76 FORI=9TO11:FORJ=10TO12:IFPEEK(7680+I+22*J)=2T
HENPOKE7680+I+22*J,16
77 NEXTJ,I
78 POKE7932,8:X=10:Y=11:POKE0,X:POKE1,Y:SYS7524
80 FORI=1TO8:POKE7723-I,16:IFLE=>1THENPOKE7723-I
,8
85 NEXT
90 FORI=828TO845:POKEI,0:NEXT:FORI=160TO388STEP-1
:POKE36865,I:NEXTI
95 IFPEEK(197)<>47THEN95
100 POKE36876,0:IFBTHENB=B-5:PRINT"(HOME DOWN RI
GHT)RVSON WHITE SPACES LEFT5)"B
102 IFM=3ANDW=0THENW=1:POKE36878,15:GOTO105
103 IFM=3THENW=0:POKE36878,239
104 IFM=2THENW=0:GOSUB300:GOTO105
105 SYS6593:P=PEEK(7680+X+22*Y):IFP<7DRP>16THEN4
00
106 V=0:RX=0:RY=0:J=PEEK(5):IFM=4OR(JAND32)=0THE
NGOSUB180
107 IF(JANDB)=0THENRY=1:V=1:GOSUB200:GOTO111
<16>
<120>
<60>
<194>
<8>
<12>
<8>
<67>
<85>
<66>
<38>
<237>
<50>
<24>
<137>
<178>
<34>
<60>
<11>
<96>
<170>
<215>
<217>
<5>
<182>
<23>
<199>
<90>
<146>
<50>
<29>
108 IF(JAND16)=0THENRX=-1:V=2:GOSUB200:GOTO111
109 IF(JAND128)=0THENRX=1:V=3:GOSUB200:GOTO111
110 IF(JAND4)=0THENRY=-1:V=1:GOSUB200:GOTO111
111 IF(P>2ANDP<7)DRP>16THEN400
112 POKE7680+X+22*Y,Z(V,U):SYS7524
114 IFPEEK(6)=0THEN450
116 IFTW=1THENTN=TN+2:POKE36876,8:IFTN>230THEN
W=0:TN=0
130 GOTO100
180 IFPEEK(828+BZ*3)<>0THENRETURN
182 TW=1:TN=230-ZZ*2:D=7634+X+22*Y:POKE829+BZ*3,
DAND255:POKE830+BZ*3,D/256
184 POKE828+BZ*3,ZZ:BZ=BZ+1:IFBZ=5THENBZ=0
185 RETURN
200 IFU=0THENU=1:GOTO210
205 U=0
210 P=PEEK(7680+X+RX+22*(Y+RY)):IFP<3THENRETURN
215 POKE7680+X+22*Y,16:X=X+RX:Y=Y+RY:POKE0,X:POK
E1,Y:IFP<>15THENRETURN
216 POKE7680+X+22*Y,Z(V,U):SYS7524:FORT=128TO245
STEP(256-T)/128
217 POKE36874,T:NEXTT:POKE36874,0:LE=LE+1:IFLE>8
THENLE=8
220 POKE7723-LE,8:RETURN
300 P=INT(RND(1)*19)+J=0:IFP/2=INT(P/2)THENJ=1
302 D=J+2*INT(RND(1)*(9-J)+J):D=7747+D+P*22
304 IFPEEK(D)=2THENPOKE0,16:RETURN
306 IFPEEK(D)=16THENPOKE0,2:RETURN
308 RETURN
400 POKE36878,239:FORI=1TO11:SYS6593:FORT=1TO150
:NEXTI,I:FORT=1TO5
405 FORI=245TO160STEP-T:POKE36875,I:NEXTI,T:FORT
=4TO1STEP-.5:FORT=245TO160STEP-T
408 POKE36875,I:NEXTI,T:POKE36875,0:SC=SC+(AL-PE
EK(6))*50:LE=LE-1:IFLE<0THEN4
410 GOTO46
450 POKE36878,239:FORT=1TO5:FORI=200TO245STEP:T:P
OKE36876,I:NEXTI,T
455 FORT=4TO1STEP-.5:FORT=200TO245STEP:T:POKE3687
6,I:NEXTI,T:POKE36876,0
460 SC=SC+(AL-PEEK(6))*50+D:FORT=1TO500:NEXTT:BO
TO40
500 DATA191,198,204,207,213,217,221,223,227
505 DATA2,24,48,47,90,68,44,45
ENDE DES LISTINGS
<85>
<141>
<14>
<92>
<113>
<72>
<124>
<189>
<141>
<133>
<39>
<71>
<192>
<22>
<138>
<218>
<163>
<109>
<125>
<113>
<112>
<86>
<195>
<12>
<111>
<89>
<46>
<56>
<53>
<47>
<130>
<114>

```

C 16/116: Note Screen

Das Maschinenprogramm "NOTE-SCREEN" stellt einen zweiten Bildschirm zur Verfügung, den man als Notizblock benutzen kann. Dies ist besonders nützlich bei der Programmentwicklung. Auf dem zweiten Bildschirm kann man sich Notizen der angewandten Variablen und deren Funktionen machen. Da meistens ein Programm nicht am gleichen Tag beendet wird, kann man sich die Notizen auf Diskette oder Kassette abspeichern und sie bei der Weiterbearbeitung des Programms wieder laden. Die angewandte Tastenkombination und ihre Bedeutung befinden sich am Anfang des Basicladers. Für Maschinensprache-Freaks, oder die es werden wollen, befindet sich ab Zeile 300 ein ausführlich dokumentiertes Assemblerlisting. Beim Abtippen des Basicladers kann man selbstverständ-

lich die "Remarks" weglassen. Nun zur Funktionsbeschreibung. Nach dem Starten des "Laders" werden die Daten in den Speicher eingelesen und zugleich überprüft. Befindet sich ein Fehler in den Daten, so wird sofort angezeigt, in welcher Zeile er sich befindet. Sind die Daten korrekt, wird gefragt, ob man ein Kassetten- oder Disketten-Lautwerk benutzt. Danach wird in den Monitor gesprungen und der Text zur Abspeicherung der Maschinenroutine auf den Bildschirm gegeben. Jetzt ist nur noch die "RETURN"-Taste zu drücken und das Maschinenprogramm wird gespeichert. Danach ist "X" zu drücken, um zurück ins Basic zu kommen. Die Routine kann nun zu jeder Zeit mit der Eingabe von "Monitor" und "L" N-SCREEN/SYS 1525", 8" (bei Kassetten ..., 1) geladen werden. Nun doch "X"

eingeben und die Routine mit "SYS 1525" starten. Nach dem gleichzeitigen Drücken der "CTRL"-und "COMMODORE"-Taste wird auf den zweiten Bildschirm umgeschaltet. Wird zum ersten mal umgeschaltet, ist der Bildschirm wahrscheinlich mit Zeichen gefüllt und ist durch Drücken von SHIFT/CLEAR HOME zu löschen. Unten rechts in der Bildecke befindet sich ein invertiertes Zeichen das anzeigt, daß der "NOTE-SCREEN" eingeschaltet ist. Jetzt kann man sich beliebige Notizen machen und mit gleichzeitigem Drücken von "CTRL" und "SHIFT" wird der Bildschirm in den geschützten RAM-Speicher kopiert und zurück zum normalen Bildschirm gesprungen. Will man den Bildschirm speichern, ist der "NOTE-SCREEN" einzuschalten, in die obere linke Ecke des Bildschirms eine Zahl zwischen 0 und 9 zu schreiben und dann gleichzeitig die "CTRL"- und die "S"-Taste drücken. Da die Zahl als "FILENAME" benutzt wird, kann man mit

verschiedenen Bildschirmen arbeiten. Beim Abspeichern ist jedoch darauf zu achten, daß sich mit einer schon einmal benutzten Datei, der Bildschirm nicht abspeichern läßt. Laden kann man den Bildschirm wieder, indem man den "NOTE-SCREEN" einschaltet, links oben die Kennzahl eingibt und dann gleichzeitig die "CTRL"- und die "L"-Taste drückt. Befindet man sich im "NOTE-SCREEN" ist das Scrollen nur im Direkt-Modus abgeschaltet. Läßt man ein Programm Listen oder lädt man ein Directory, so hat der Bildschirm die normale Scroll-Funktion. Um einen Syntax Error zu vermeiden, hat die "RETURN"-Taste ihre Funktion als Eingabe-Taste verloren. Sie kann nur als Zeilensprung benutzt werden. Befindet sich der "CURSOR" in der unteren linken Ecke auf dem invertierten Zeichen, so behalten nur noch die Tasten "CURSOR UP" und "CURSOR LEFT" ihre Funktion. Alle anderen Eingaben werden ignoriert.

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

```

10 REM *****
15 REM "(SPACE17)"
20 REM "(SPACE)* (SPACE) NOTE-SCREEN (SPACE) * (SPACE)
E)"
25 REM "(SPACE17)"
30 REM "(SPACE2) (BASICLADER) (SPACE3)"
35 REM "(SPACE17)"
40 REM "(SPACE2) STARTADRESSE (SPACE3)"
45 REM "(SPACE4) SYS1525 (SPACE6)"
50 REM "(SPACE17)"
55 REM "(SPACE3) WRITTEN (SPACE) BY (SPACE4)"
60 REM "(SPACE) GUNTER (SPACE3) MASSAR (SPACE)"
65 REM "(SPACE17)"
70 REM *****
75 :
80 REM N.SCREEN ON = 'CTRL'+ 'COMMODORE'
85 REM
90 REM N.SCREEN OFF = 'CTRL'+ 'SHIFT'
95 REM
100 REM SAVE SCREEN =
105 REM N.SCREEN ON + 'CTRL' + 'S'
110 REM
115 REM LOAD SCREEN =
120 REM N.SCREEN ON + 'CTRL' + 'L'
125 REM
130 SCNCLR:FLASH0,11,6,"(FLASHON)BITTE (SPACE) WART
EN (SPACE) ! (FLASHOFF)"
135 FORK=1525TO1765:READD:CHAR1,11,10,"ADRESSE (S
PACE)";:PRINTK
140 CHAR1,11,12,"DATA (SPACE4): (SPACE2)";:PRINTUSI
NG"###";D
<91>
<234>
<7>
<244>
<3>
<254>
<171>
<122>
<13>
<208>
<31>
<28>
<151>
<133>
<20>
<228>
<82>
<238>
<185>
<195>
<253>
<139>
<63>
<12>
<156>
<51>
<224>

```


programme

145 Z=Z+1;S=S+D;POKEK,D;GS=GS+D	<123>	ZUM(SPACE)BASIC	<40>
150 IFZ=12THENREADCH;GOTO160	<181>	370 REM	<2>
155 NEXTK:IFGS<>27514THEN165:ELSE170	<174>	375 REM *** IRQ-ROUTINE ***	<106>
160 IFS=CHTHENS=0;Z=0;NEXTK	<46>	380 REM	<12>
165 SCNCLR:CHAR1,7,10,"":PRINT(RVSON)DATA(SPACE)		385 REM "060A(SPACE2)LDA(SPACE)#E5(SPACE4)MERE(SPACE)HOLEN	<183>
1FEHLER(SPACE)IN(SPACE)ZEILE"PEEK(63)+PEEK(64)*2		390 REM "060C(SPACE2)BNE(SPACE)#062B(SPACE2)MERKER(SPACE)GESETZT,(SPACE)DANN(SPACE)SPRUNG(SPACE)	<13>
56:END		E)ZUR(SPACE)PAGE-ROUTINE	
170 SCNCLR:CHAR1,10,10,"(RVSON)DATAS(SPACE)IN(SPACE)	<46>	395 REM "060E(SPACE2)LDA(SPACE)#06(SPACE3)'CD	<124>
ACE)ORDNUNG(RVSOFF)"	<96>	NTRDL'+COMMDDRE'(SPACE)...	
175 CHAR1,8,15,"(RVSON FLASHON)K(RVSOFF FLASHOFF)		400 REM "0610(SPACE2)CMP(SPACE)#0543(SPACE2)..	<172>
1)ASBETTE(SPACE)/(SPACE RVSON FLASHON)D(RVSOFF FL	<219>	.(SPACE)TASTE(SPACE)GEDRUECKT(SPACE)?	
ASHOFF)ISKETTE(SPACE)?":GETKEYG#:IFG#"K"THENG=1	<31>	405 REM "0613(SPACE2)BNE(SPACE)#062B(SPACE2)NE	<158>
:GOTO185	<145>	IN,(SPACE)DANN(SPACE)SPRUNG(SPACE)ZUR(SPACE)IRQ-	<4>
180 IFG#"D"THENG=8:ELSEGOTO175	<26>	ROUTINE(SPACE)D.(SPACE)BETRIEBSSYSTEMS	
185 SCNCLR:CHAR1,5,10,"BITTE(SPACE RVSON)RETURN(RVSOFF	<85>	410 REM "0615(SPACE2)INC(SPACE)#E5(SPACE4)MERE(SPACE)SETZEN	<244>
SPACE)TASTE(SPACE)DRUECKEN"	<199>	415 REM "0617(SPACE2)LDX(SPACE)#0C(SPACE3)HIG	
190 CHAR1,0,20,"":PRINT"8";CHR\$(34);"N.SCREEN/SY	<85>	H-BYTE(SPACE)DER(SPACE)EMPFANGSADRESSE(SPACE)HOLEN(SPACE)	<32>
S1525";CHR\$(34);",G,05F5,06E8"	<147>	EN(SPACE)(SCREEN)	<61>
195 POKE3872,8+48;POKE1746,8;SYS1525:CHAR1,0,15,"":MONITOR	<49>	420 REM "0619(SPACE2)LDA(SPACE)#3C(SPACE3)HIG	
200 DATA120,169,59,133,56,160,10,162,6,140,20,3,1038	<122>	H-BYTE(SPACE)DER(SPACE)SENDEADRESSE(SPACE)HOLEN(SPACE)	<244>
205 DATA142,21,3,169,0,133,229,88,96,165,229,208,1483	<158>	425 REM "061B(SPACE2)JSR(SPACE)#064E(SPACE2)ZUR(SPACE)KOPIERROUTINE	<32>
210 DATA29,169,6,205,67,5,208,19,230,229,162,12,1341	<154>	430 REM "061E(SPACE2)LDY(SPACE)#69(SPACE3)LOW	<61>
215 DATA169,60,32,78,6,160,105,162,6,140,69,5,992	<225>	-BYTE(SPACE)DES(SPACE)NEUEN(SPACE)TASTATUR-VEKTO	<216>
220 DATA142,70,5,76,14,206,142,231,15,173,67,5,1146	<29>	RS(SPACE)HOLEN	<91>
225 DATA201,5,208,22,198,229,160,122,162,219,140,69,1735	<4>	435 REM "0620(SPACE2)LDX(SPACE)#06(SPACE3)ENT	<240>
230 DATA5,142,70,5,162,60,169,12,32,78,6,32,773	<180>	SPRECHENDES(SPACE)HIGH-BYTE(SPACE)HOLEN	<87>
235 DATA139,216,76,14,206,160,0,132,3,132,5,134,1217	<130>	440 REM "0622(SPACE2)STY(SPACE)#0544(SPACE2)LD	<47>
240 DATA4,133,6,162,4,177,5,145,3,200,208,249,1296	<39>	W-BYTE(SPACE)UND(SPACE)...	<97>
245 DATA230,4,230,6,202,208,242,96,173,67,5,201,1664	<78>	445 REM "0625(SPACE2)BTX(SPACE)#0546(SPACE2)..	
250 DATA4,208,10,165,198,201,13,240,55,201,42,240,1577	<200>	.(SPACE)ENTSPRECHENDES(SPACE)HIGH-BYTE(SPACE)SETZEN	<216>
255 DATA71,160,64,165,205,201,24,16,12,165,198,201,1482	<217>	450 REM "062B(SPACE2)JMP(SPACE)#CE0E(SPACE2)ZUR(SPACE)IRQ-ROUTINE(SPACE)DES(SPACE)BETRIEBSSYSTEMS	<91>
260 DATA1,208,3,141,67,5,76,122,219,165,198,201,1406	<41>	455 REM	<240>
265 DATA1,240,4,201,40,208,2,132,198,166,202,224,1618	<64>	460 REM *** PAGE-ROUTINE ***	<87>
270 DATA39,48,10,201,48,240,6,201,43,240,2,132,1210	<6>	465 REM	<47>
275 DATA198,76,122,219,32,201,6,169,0,133,230,169,1555	<83>	470 REM "062B(SPACE2)STX(SPACE)#0FE7(SPACE2)ZE	<97>
280 DATA12,133,231,169,230,162,1,160,16,76,216,255,1661	<188>	ICHEN(SPACE)UNTEN(SPACE)RECHTS(SPACE)IN(SPACE)BILD	<153>
285 DATA32,201,6,169,0,76,213,255,173,0,12,141,1778	<95>	DSCHIRM(SPACE)SCHREIBEN	<88>
290 DATA230,6,169,2,162,8,160,1,32,186,255,169,1380	<198>	475 REM "062E(SPACE2)LDA(SPACE)#0543(SPACE2)'CONTROL'+SHIFT'(SPACE)...	<153>
295 DATA1,162,230,160,6,32,189,255,160,0,132,239,1566,96	<47>	480 REM "0631(SPACE2)CMP(SPACE)#05(SPACE3)..	<88>
300 REM	<115>	(SPACE)TASTE(SPACE)GEDRUECKT?	<186>
305 REM *** NEUER IRQ-VEKTOR EINSTELLEN ***	<222>	485 REM "0633(SPACE2)BNE(SPACE)#064B(SPACE2)NE	<42>
310 REM	<39>	IN,(SPACE)DANN(SPACE)ZUR(SPACE)IRQ-ROUTINE(SPACE)DES(SPACE)BETRIEBSSYSTEMS	<198>
315 REM "05F5(SPACE2)SEI(SPACE8)INTERRUPT(SPACE)SPERREN	<88>	490 REM "0635(SPACE2)DEC(SPACE)#E5(SPACE4)MERE(SPACE)RUECKSETZEN	<115>
320 REM "05F6(SPACE2)LDA(SPACE)#3B(SPACE3)HIG	<24B>	495 REM "0637(SPACE2)LDY(SPACE)#7A(SPACE3)LOW	<82>
H-BYTE(SPACE)DES(SPACE)BASISPEICHER-ENDE(SPACE)HOLEN	<118>	-BYTE(SPACE)DES(SPACE)NORM.(SPACE)TASTATUR-VEKTO	<143>
325 REM "05FB(SPACE2)STA(SPACE)#3B(SPACE4)UND(SPACE)SETZEN	<216>	RS(SPACE)HOLEN	<229>
330 REM "05FA(SPACE2)LDY(SPACE)#0A(SPACE3)LOW	<169>	500 REM "0639(SPACE2)LDX(SPACE)#DB(SPACE3)ENT	<82>
-BYTE(SPACE)DES(SPACE)NEUEN(SPACE)IRQ-VEKTORS(SPACE)HOLEN	<111>	SPRECHENDES(SPACE)HIGH-BYTE(SPACE)HOLEN	<143>
335 REM "05FC(SPACE2)LDX(SPACE)#06(SPACE3)ENTSPRECHENDES(SPACE)HIGH-BYTE(SPACE)HOLEN		505 REM "063B(SPACE2)STY(SPACE)#0545(SPACE2)LD	
340 REM "05FE(SPACE2)STY(SPACE)#0314(SPACE2)LOW-BYTE(SPACE)UND(SPACE)...		W-BYTE(SPACE)UND(SPACE)...	
345 REM "0601(SPACE2)STX(SPACE)#0315(SPACE2)..		510 REM "063E(SPACE2)STX(SPACE)#0546(SPACE2)..	
.(SPACE)ENTSPRECHENDES(SPACE)HIGH-BYTE(SPACE)SETZEN		.(SPACE)ENTSPRECHENDES(SPACE)HIGH-BYTE(SPACE)SETZEN	
350 REM "0604(SPACE2)LDA(SPACE)#00(SPACE3)MERKER(SPACE)...		515 REM "0641(SPACE2)LDX(SPACE)#3C(SPACE3)HIG	<1>
355 REM "0606(SPACE2)STA(SPACE)#E5(SPACE4)...(SPACE)ZURUECKSETZEN		H-BYTE(SPACE)DER(SPACE)EMPFANGSADRESSE(SPACE)HOLEN(SPACE)	<161>
360 REM "060B(SPACE2)CLI(SPACE8)IRQ(SPACE)WIEDER(SPACE)FREIBEBEN		EN(SPACE)(RAM)	<115>
365 REM "0609(SPACE2)RTS(SPACE8)ZURUECK(SPACE)		520 REM "0643(SPACE2)LDA(SPACE)#0C(SPACE3)HIG	<207>
		H-BYTE(SPACE)DER(SPACE)SENDEADRESSE(SPACE)HOLEN(SPACE)	<157>
		(SCREEN)	
		525 REM "0645(SPACE2)JSR(SPACE)#064E(SPACE2)ZUR(SPACE)KOPIER-ROUTINE	
		530 REM "064B(SPACE2)JSR(SPACE)#DBBB(SPACE2)ZUR(SPACE)SCNCLR	
		535 REM "064B(SPACE2)JMP(SPACE)#CE0E(SPACE2)ZUR(SPACE)IRQ-ROUTINE(SPACE)DES(SPACE)BETRIEBSSYSTEMS	
		540 REM	<144>
		545 REM *** KOPIER-ROUTINE ***	<173>
		550 REM	<98>
		555 REM "064E(SPACE2)LDY(SPACE)#00(SPACE3)LOW	<183>
		-BYTE(SPACE)DES(SPACE)SENDE-(SPACE)UND(SPACE)EMPFANGSADRESSE(SPACE)HOLEN	<137>
		560 REM "0650(SPACE2)STY(SPACE)#03(SPACE4)LOW-BYTE(SPACE)INB(SPACE)EMPFANG-(SPACE)UND(SPACE)...	<78>

565 REM "0652 (SPACE2) STY (SPACE) #05 (SPACE4) ... (SPACE) SENDEREGB. (SPACE) BETZEN	<247>	765 REM "069C (SPACE2) CPX (SPACE) #27 (SPACE3) ... (SPACE) SPALTE?	<52>
570 REM "0654 (SPACE2) STX (SPACE) #04 (SPACE4) HIGH-BYTE (SPACE) INS (SPACE) EMPFANGREG. (SPACE) BETZEN	<233>	770 REM "069E (SPACE2) BMI (SPACE) #06AA (SPACE2) NE IN, DANN (SPACE) SPRUNG (SPACE) Z. TASTATUR-ROUTINE (SPACE) D. (SPACE) BETRIEBBYST.	<93>
575 REM "0656 (SPACE2) STA (SPACE) #06 (SPACE4) HIGH-BYTE (SPACE) INS (SPACE) SENDEREGB. (SPACE) BETZEN	<225>	775 REM "06A0 (SPACE2) CMP (SPACE) #30 (SPACE3) 'CURSOR (SPACE) LEFT' (SPACE) BEDRUECKT?	<126>
580 REM "065B (SPACE2) LDX (SPACE) #04 (SPACE3) ZAEHLER (SPACE) LADEN	<249>	780 REM "06A2 (SPACE2) BEQ (SPACE) #06AA (SPACE2) JA, (SPACE) DANN (SPACE) SPRUNG (SPACE) Z. (SPACE) TAST. -R. (SPACE) D. (SPACE) BETRIEB.	<48>
585 REM "065A (SPACE2) LDA (SPACE) #05, Y (SPACE) ZEICHEN (SPACE) VON (SPACE) SENDEADRESSE (SPACE) IN (SPACE) ...	<91>	785 REM "06A4 (SPACE2) CMP (SPACE) #2B (SPACE3) 'CURSOR (SPACE) UP' (SPACE) BEDRUECKT?	<68>
590 REM "065C (SPACE2) STA (SPACE) #03, Y (SPACE) ... (SPACE) EMPFANGSADRESSE (SPACE) KOPIEREN	<177>	790 REM "06A6 (SPACE2) BEQ (SPACE) #06AA (SPACE2) JA, (SPACE) DANN (SPACE) SPRUNG (SPACE) Z. (SPACE) TAST. -R. (SPACE) D. (SPACE) BETRIEB.	<82>
595 REM "065E (SPACE2) INY (SPACE8) LOW-BYTE (SPACE) DER (SPACE) REGISTER (SPACE) +1	<199>	795 REM "06AB (SPACE2) STY (SPACE) #C6 (SPACE4) BEDRUECKTE (SPACE) TASTE (SPACE) LOESCHEN	<37>
600 REM "065F (SPACE2) BNE (SPACE) #065A (SPACE2) UNGLEICH (SPACE) NULL? (SPACE) DANN (SPACE) NAECHSTES (SPACE) ZEICHEN (SPACE) KOPIEREN	<111>	800 REM "06AA (SPACE2) JMP (SPACE) #DB7A (SPACE2) ZUR (SPACE) TASTATUR-ROUTINE (SPACE) DES (SPACE) BETRIEB SYSTEMS	<5>
605 REM "0661 (SPACE2) INC (SPACE) #04 (SPACE4) HIGH-BYTE (SPACE) DES (SPACE) EMPFANGS- (SPACE) UND (SPACE) ...	<58>	805 REM	<183>
610 REM "0663 (SPACE2) INC (SPACE) #06 (SPACE4) ... (SPACE) SENDEREGB. (SPACE) ERHOEHEN	<182>	810 REM *** SAVE-ROUTINE ***	<7>
615 REM "0665 (SPACE2) DEX (SPACE8) ZAEHLER (SPACE) -1	<251>	815 REM	<193>
620 REM "0666 (SPACE2) BNE (SPACE) #065A (SPACE2) UNGLEICH (SPACE) NULL? (SPACE) DANN (SPACE) NAECHSTES (SPACE) ZEICHEN (SPACE) KOPIEREN	<40>	820 REM "06AD (SPACE2) JSR (SPACE) #06C9 (SPACE2) ZUR (SPACE) SETLFS- (SPACE) UND (SPACE) SETNAM-ROUTINE	<10>
625 REM "066B (SPACE2) RTS (SPACE8) RUECKSPRUNG	<216>	825 REM "06B0 (SPACE2) LDA (SPACE) #00 (SPACE3) LOW-BYTE (SPACE) DER (SPACE) ANFANGSADRESSE (SPACE) HOLEN (SPACE) ...	<251>
630 REM	<7>	830 REM "06B2 (SPACE2) STA (SPACE) #E6 (SPACE4) ... (SPACE) UND (SPACE) SETZEN	<53>
635 REM *** TASTATUR-ROUTINE ***	<19>	835 REM "06B4 (SPACE2) LDA (SPACE) #0C (SPACE3) HIGH-BYTE (SPACE) DER (SPACE) ANFANGSADRESSE (SPACE) HOLEN (SPACE) ...	<88>
640 REM	<17>	840 REM "06B6 (SPACE2) STA (SPACE) #E7 (SPACE4) ... (SPACE) UND (SPACE) SETZEN	<102>
645 REM "0669 (SPACE2) LDA (SPACE) #0543 (SPACE2) 'CONTROL' -TASTE (SPACE) ...	<126>	845 REM "06BB (SPACE2) LDA (SPACE) #E6 (SPACE3) ZEIGER (SPACE) DER (SPACE) ANFANGSADRESSE (SPACE) HOLEN	<90>
650 REM "066C (SPACE2) CMP (SPACE) #04 (SPACE3) ... (SPACE) BEDRUECKT?	<243>	850 REM "06BA (SPACE2) LDX (SPACE) #01 (SPACE3) LOW-BYTE (SPACE) DER (SPACE) ENDRADRESSE (SPACE) HOLEN	<114>
655 REM "066E (SPACE2) BNE (SPACE) #067A (SPACE2) NE IN, (SPACE) DANN (SPACE) 'B' (SPACE) UND (SPACE) 'L' (SPACE) TASTENABFRAGE (SPACE) LIEBERSPRINGEN	<253>	855 REM "06BC (SPACE2) LDY (SPACE) #10 (SPACE3) ENTSPRECHENDES (SPACE) HIGH-BYTE (SPACE) HOLEN	<12>
660 REM "0670 (SPACE2) LDA (SPACE) #C6 (SPACE4) 'B' (SPACE) TASTE (SPACE) ...	<101>	860 REM "06BE (SPACE2) JMP (SPACE) #FFDB (SPACE2) ZUR (SPACE) SAVE-ROUTINE (SPACE) DES (SPACE) BETRIEBBYST EMS	<122>
665 REM "0672 (SPACE2) CMP (SPACE) #0D (SPACE3) ... (SPACE) BEDRUECKT?	<161>	865 REM	<243>
670 REM "0674 (SPACE2) BEQ (SPACE) #06AD (SPACE2) JA, (SPACE) DANN (SPACE) ZUR (SPACE) SAVE-ROUTINE	<62>	870 REM *** LOAD-ROUTINE ***	<217>
675 REM "0676 (SPACE2) CMP (SPACE) #2A (SPACE3) 'L' (SPACE) TASTE (SPACE) BEDRUECKT?	<222>	875 REM	<253>
680 REM "0678 (SPACE2) BEQ (SPACE) #06C1 (SPACE2) JA, (SPACE) DANN (SPACE) ZUR (SPACE) LOAD-ROUTINE	<47>	880 REM "06C1 (SPACE2) JSR (SPACE) #06C9 (SPACE2) ZUR (SPACE) SETLFS- (SPACE) UND (SPACE) SETNAM-ROUTINE	<222>
685 REM "067A (SPACE2) LDY (SPACE) #40 (SPACE3) LOESCH-KODE (SPACE) HOLEN	<42>	885 REM "06C4 (SPACE2) LDA (SPACE) #00 (SPACE3) FLAG (SPACE) FUER (SPACE) LOAD (SPACE) HOLEN	<183>
690 REM "067C (SPACE2) LDA (SPACE) #CD (SPACE4) CURSOR (SPACE) IN (SPACE) LETZTER (SPACE) ...	<170>	890 REM "06C6 (SPACE2) JMP (SPACE) #FFD5 (SPACE2) ZUR (SPACE) LOAD-ROUTINE (SPACE) DES (SPACE) BETRIEBBYST EMS	<137>
695 REM "067E (SPACE2) CMP (SPACE) #1B (SPACE3) ... (SPACE) ZEILE?	<44>	895 REM	<17>
700 REM "0680 (SPACE2) BPL (SPACE) #068E (SPACE2) JA, DANN (SPACE) ZUR (SPACE) 'RETURN' (SPACE) U. (SPACE) 'CURSOR (SPACE) DOWN' (SPACE) SPERREN	<50>	900 REM *** SETLFS- U. SETNAM-ROUTINE ***	<223>
705 REM "06B2 (SPACE2) LDA (SPACE) #C6 (SPACE4) 'RETURN' (SPACE) TASTE (SPACE) ...	<40>	905 REM	<27>
710 REM "06B4 (SPACE2) CMP (SPACE) #01 (SPACE3) ... (SPACE) BEDRUECKT?	<175>	910 REM "06C9 (SPACE2) LDA (SPACE) #0C00 (SPACE2) FILENAME (SPACE) AUF (SPACE) BILDschirm (SPACE) ECKE (SPACE) HOLEN (SPACE) ...	<204>
715 REM "06B6 (SPACE2) BNE (SPACE) #06BB (SPACE2) NE IN, (SPACE) DANN (SPACE) SPRUNG (SPACE) ZUR (SPACE) TASTATUR-ROUTINE (SPACE) D. (SPACE) BETRIEB.	<90>	915 REM "06CC (SPACE2) STA (SPACE) #06E6 (SPACE2) ... (SPACE) UND (SPACE) SETZEN	<149>
720 REM "06B8 (SPACE2) STA (SPACE) #0543 (SPACE2) 'BHIFT' -KODE (SPACE) ERZEUGEN	<131>	920 REM "06CF (SPACE2) LDA (SPACE) #01 (SPACE3) FILENUMBER (SPACE) HOLEN	<51>
725 REM "06BB (SPACE2) JMP (SPACE) #DB7A (SPACE2) ZUR (SPACE) TASTATUR-ROUTINE (SPACE) DES (SPACE) BETRIEB SYSTEMS	<203>	925 REM "06D1 (SPACE2) LDX (SPACE) #0B (SPACE3) BEZUGSADRESSE (SPACE) HOLEN	<179>
730 REM "06BE (SPACE2) LDA (SPACE) #C6 (SPACE4) 'RETURN' (SPACE) TASTE (SPACE) ...	<179>	930 REM "06D3 (SPACE2) LDY (SPACE) #01 (SPACE3) BEKUNDAERADRESSE (SPACE) HOLEN	<218>
735 REM "0690 (SPACE2) CMP (SPACE) #01 (SPACE3) ... (SPACE) BEDRUECKT?	<181>	935 REM "06D5 (SPACE2) JSR (SPACE) #FFBA (SPACE2) FILEPARAMETER (SPACE) SETZEN	<119>
740 REM "0692 (SPACE2) BEQ (SPACE) #069B (SPACE2) JA, (SPACE) DANN (SPACE) SPRUNG (SPACE) ZUR (SPACE) KODE-LUECKUNG	<33>	940 REM "06DB (SPACE2) LDA (SPACE) #01 (SPACE3) LANGE (SPACE) DES (SPACE) FILENAMENS	<188>
745 REM "0694 (SPACE2) CMP (SPACE) #2B (SPACE3) 'CURSOR (SPACE) DOWN' (SPACE) BEDRUECKT?	<70>	945 REM "06DA (SPACE2) LDX (SPACE) #E6 (SPACE3) LOW-BYTE (SPACE) DER (SPACE) FILENAMENADRESSE (SPACE) HOLEN	<68>
750 REM "0696 (SPACE2) BNE (SPACE) #069A (SPACE2) NE IN, (SPACE) DANN (SPACE) SPRUNG (SPACE) ZUR (SPACE) ABFRAGE (SPACE) DER (SPACE) CURSOR (SPACE) SPALTE	<160>	950 REM "06DC (SPACE2) LDY (SPACE) #06 (SPACE3) ENTSPRECHENDES (SPACE) HIGH-BYTE (SPACE) HOLEN	<198>
755 REM "069B (SPACE2) STY (SPACE) #C6 (SPACE4) BEDRUECKTE (SPACE) TASTE (SPACE) LOESCHEN	<212>	955 REM "06DE (SPACE2) JSR (SPACE) #FFBD (SPACE2) FILENAMEN (SPACE) SETZEN	<204>
760 REM "069A (SPACE2) LDX (SPACE) #CA (SPACE4) CURSOR (SPACE) IN (SPACE) LETZTER (SPACE) ...	<190>	960 REM "06E1 (SPACE2) LDY (SPACE) #00 (SPACE3) TASTATURPUFFER	<187>
		965 REM "06E3 (SPACE2) STY (SPACE) #EF (SPACE4) ... (SPACE) LOESCHEN	<56>
		970 REM "0635 (SPACE2) RTS (SPACE8) RUECKSPRUNG ENDE DES LISTINGS	<17>

C 16/116: **Champ**

Hier handelt es sich um ein spannendes Boxspiel. Hervorragende Programmierung in Maschinensprache garantiert schnelle Bewegungen und fordert vom Spieler rasches Reagieren. Beide Teile werden getrennt abgetippt und gespeichert.

Nun können Sie die Programme wieder getrennt einladen und starten. Gespielt wird mit Joystick oder Tastatur.

Nach dem Start können Sie alles weitere aus dem Programm heraus erfahren.

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

```

10 POKE56,47:CLR:COLOR0,1:COLOR4,1 <33>
20 PRINT"BLACK CLEAR DOWN3"(SO DOWN6)T0000(SPA <73>
CE)D3FF(SPACE)3000(DOWN2)X(DOWN2)BOT040(HOME)" <112>
30 FORI=0T03:POKE1319+I,13:NEXT:POKE239,4:END <220>
40 POKE65298,195:POKE65299,48:PRINT"WHITE CLEAR <19>
J" <26>
50 RESTORE70:FORI=14472T014503:READZ#:POKEI,DEC <141>
(Z#):NEXT <80>
60 RESTORE110:FORI=14336T014423:READZ#:POKEI,DEC <204>
(Z#):NEXT <250>
70 :DATA0,00,B1,D6,99,00,32,CB <105>
80 :DATA0,4B,D0,F6,60,A0,00,B1 <203>
90 :DATA0B,99,4B,32,CB,C0,4B,D0 <172>
100 :DATAF6,60,00,00,00,00,00,00 <173>
110 DATA7B,A2,16,A0,3B,8E,14,03 <199>
120 DATA0C,15,03,5B,A2,00,86,D1 <214>
130 DATA2,14,8E,11,FF,60,C6,D0 <32>
140 DATAF0,03,4C,0E,CE,A6,D1,BD <104>
150 DATA00,35,8D,0E,FF,8D,00,36 <219>
160 DATA0D,12,FF,8D,00,37,85,D0 <161>
170 DATAE4,D2,F0,05,E6,D1,4C,0E <76>
180 DATA0E,A2,00,86,D1,4C,0E,CE <194>
190 DATA7B,A2,0E,A0,CE,8E,14,03 <49>
200 DATA0C,15,03,5B,A2,00,8E,11 <59>
210 DATAFF,B6,D1,A2,01,86,D0,60 <215>
220 RESTORE230:FORI=13568T013695:READZ#:POKEI,DE <221>
C(Z#):NEXT <210>
230 DATA0D,E3,02,FC,02,1E,AD,FC <193>
240 DATAAD,E3,02,FC,02,1E,AD,FC <203>
250 DATA42,2A,42,2A,42,56,FC,2A <190>
260 DATA42,1E,2A,02,1E,E3,60,56 <37>
270 DATAFC,FC,FC,FC,FC,FC,FC,FC <116>
280 DATA04,FC,04,54,83,FC,83,FC <192>
290 DATA04,FC,04,54,83,FC,83,FC <159>
300 DATA54,FC,54,AD,C0,FC,C0,E3 <219>
310 DATAFC,E3,02,FC,02,1E,FC,1E <126>
320 DATA2A,FC,2A,FC,C0,AD,C0,AD <61>
330 DATA54,04,83,04,FC,54,C0,E3 <180>
340 DATA02,2A,02,83,AD,83,FC,83 <196>
350 DATA54,83,54,20,FC,10,02,10 <74>
360 DATA02,E3,FC,02,C0,10,02,C0 <84>
370 DATA02,F3,C0,10,F3,C0,FC,9B <124>
380 DATA54,C0,FC,00,EA,EA,EA,EA <116>
390 RESTORE400:FORI=13824T013951:READZ#:POKEI,DE <144>
C(Z#):NEXT <94>
400 DATA2,C2,C3,C3,C3,C3,C2,C3 <104>
410 DATA2,C2,C3,C3,C3,C3,C2,C3 <90>
420 DATA3,C3,C3,C3,C3,C3,C3,C3 <178>
430 DATA3,C3,C3,C3,C3,C2,C3,C3 <116>
440 DATA3,C3,C3,C3,C3,C3,C3,C3 <111>
450 DATA2,C3,C2,C2,C2,C3,C2,C3 <145>
460 DATA2,C3,C2,C2,C2,C3,C2,C3 <195>
470 DATA2,C3,C2,C2,C2,C3,C2,C2 <190>
480 DATA3,C2,C3,C3,C3,C3,C3,C3 <173>
490 DATA3,C3,C3,C3,C2,C2,C2,C2 <170>
500 DATA2,C2,C2,C2,C3,C2,C2,C2 <74>
510 DATA3,C3,C3,C2,C2,C2,C3,C2 <246>
520 DATA2,C2,C2,C2,C3,C3,C3,C3 <190>
530 DATA3,C2,C3,C3,C2,C3,C3,C2 <173>
540 DATA3,C2,C2,C3,C2,C2,C3,C2 <170>
550 DATA2,C2,C3,C0,EA,EA,EA,EA <74>
560 RESTORE570:FORI=14080T014207:READZ#:POKEI,DE <246>
C(Z#):NEXT:POKE210,122 <74>
570 DATA10,30,70,30,10,30,70,40 <246>

```

```

580 DATA10,30,70,30,10,30,70,80 <92>
590 DATA10,10,30,10,10,70,10,12 <129>
600 DATA14,20,14,20,14,40,20,80 <73>
610 DATA40,80,01,01,01,01,01,01 <30>
620 DATA1E,02,20,20,1E,02,80,40 <113>
630 DATA1E,02,20,20,1E,02,80,40 <123>
640 DATA1E,02,20,20,1E,02,40,3E <22>
650 DATA02,20,20,02,40,3E,02,20 <40>
660 DATA1E,02,80,80,80,20,20,40 <184>
670 DATA40,20,40,80,80,20,80,20 <177>
680 DATA20,40,70,20,20,70,80,60 <244>
690 DATA40,20,60,40,20,60,40,20 <73>
700 DATA60,40,20,60,40,20,60,40 <133>
710 DATA20,60,40,20,60,80,80,20 <241>
720 DATA20,80,FF,00,EA,EA,EA,EA <219>
730 PRINT"BLACK CLEAR DOWN3"NEW(DOWN3)DLOAD"CHR <55>
$(34)"C2"CHR$(34)"(HOME)" <55>
740 POKE1319,13:POKE1320,13:POKE1321,82:POKE1322 <5>
,117:POKE1323,13:POKE239,5:END <5>
ENDE DES LISTINGS

```

Teil 2

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

```

10 SYS14400:SYB14336:V0LB <161>
20 RESTORE30:FORI=12944T013031:READZ#:POKEI,Z:NEX <2>
T <51>
30 DATA20,8B,94,8B,104,4B,252,252 <32>
40 DATA20,37,181,37,41,12,63,63 <202>
50 DATA05,44,44,44,60,150,150,130 <123>
60 DATA170,150,182,182,182,182,150,170 <1>
70 DATA144,20B,16,16,144,20B,16,16 <143>
80 DATA6,7,4,4,6,7,4,4 <73>
90 DATA255,255,255,255,255,255,255,255 <241>
100 DATA0,0,0,255,255,0,0,0 <13>
110 DATA24,24,24,24,24,24,24,24 <219>
120 DATA10B,19B,177,27,180,30,225,135 <91>
130 DATA40,174,190,182,182,150,154,40 <159>
140 RESTORE150:FORI=14790T016373:READZ#:POKEI,DE <5>
C(Z#):NEXT <23B>
150 DATA,,,,,,,,,05,16,17,16,1A,0C,37,36,,,80,,,, <53>
,45 <53>
160 DATA,,,,,,,,,3E,3E,3F,3F,15,1D,1D,2A,45,A4,A9 <12B>
,,,,, <0>
170 DATA,,,,,,,,,22,20,20,E0,40,C0,C0,50,80,A0,2B <194>
,0C,04,0C,0C,05 <8>
180 DATA,,,,,,,,,05,16,17,16,1A,0C,,,,,80,,, <139>
190 DATA,,,,,01,01,01,37,37,3A,3E,BE,BF,55,1D,,, <30>
,80,A4,14,04 <34>
200 DATA,,,,,07,07,04,1D,2A,0A,0A,0B,7B,6B,,,,,AB <253>
,AC,04,0C,0C,05 <149>
210 DATA,,,,,,,,,01,05,05,05,06,03,2D,AD,40,80,E0 <7B>
,80,80,,C0,C0 <42>
220 DATA,,01,,,,,BE,BF,4F,4F,45,07,07,0A,B5,A5, <199>
E4,C1,40,40,40,80 <78>
230 DATA,,,,,,,,,02,02,02,76,77,41,03,01,,,,,4 <78>
0 <42>
240 DATA,,,,,,,,,14,5B,5E,5B,6B,30,DC,DA,,,,,05 <253>
,05 <149>
250 DATA,,,,,,,,,FA,FC,FC,FC,54,74,74,AB,A4,01,,, <78>
,,, <42>
260 DATA,,,,,,,,,20,20,20,20,30,10,30,14,,,,,0 <78>
270 DATA,,,,,,,,,14,5B,5E,5B,6B,30,DB,DA,,,,,05,05 <42>
,24,A9,80

```


programme

```

1490 RESTORE1500:FORI=13536TD13548:READZ:POKEI,Z
:NEXT <253>
1500 DATA38,142,38,174,214,110,182,254,70,190,10
2,150,222 <109>
1510 RESTORE1520:FORI=13552TD13564:READZ:POKEI,Z
:NEXT <209>
1520 DATA61,62,61,63,62,61,61,61,62,59,63,60,60 <235>
1530 FORI=BTDO0STEP-1:VDL1:FORN=OTD250:NEXT:NEXT <37>
1540 SYS14400:PRINT"BLACK CLEAR DOWN3"NEW(DOWN3
)DLOAD"CHR*(34)"C3"CHR*(34)"(HOME)" <231>
1550 POKE1319,13:POKE1320,13:POKE1321,82:POKE132
2,117:POKE1323,13:POKE239,5:END <51>
ENDE DES LISTINGS

```

Teil 3

```

C-16 LISTING + CHECKSUMMEN (OC V1.0)
10 PRINTCHR*(142)CHR*(8) <79>
20 FORI=1TD10:NM*(I)="---":NEXT <60>
30 NM(1)=6000:NM(2)=6400:NM(3)=6800:NM(4)=7200:NM
M(5)=7600:NM(6)=8000:NM(7)=8400 <12>
40 NM(8)=8800:NM(9)=9200:NM(10)=9600 <207>
50 POKE19,1:INPUT"(WHITE CLEAR)ENTER(SPACE)YOUR(C
SPACE)NAME(SPACE):":N#:POKE19,0 <163>
60 IFLEN(N#)>3THENIN#="LEFT*(N#,3):ELSEIN#="N# <53>
70 IFLEN(N#)>16THENN#="LEFT*(N#,15)+".(SPACE)":GO
T090 <21>
80 FORI=LEN(N#)+1TD17:N#="N#+"(SPACE)":NEXT <102>
90 TRAP2200 <219>
100 SCNCLR:POKE65287,24:POKE65302,103:POKE65303,
56:F=9:POKE65286,11:GOSUBB90 <60>
110 POKE65286,27 <231>
120 G*(1)="TEETHLESS(SPACE)SAM(SPACE4)":GF(1)=57 <156>
130 G*(2)="JOHN(SPACE)"MILKY"(SPACE5)":GF(2)=60 <95>
140 G*(3)="LUIGI(SPACE)KLANOTTI(SPACE3)":GF(3)=6
1 <212>
150 G*(4)="SERGEJ(SPACE)BRECHOV(SPACE3)":GF(4)=5
8 <74>
160 GOTO1210:REM LAUFCHRIFT <62>
170 TI#="000000" <150>
180 P1=0:P2=0:ZE=T:GOSUB1660 <225>
190 GOSUB1670:SZ=5-B <141>
200 FORI=BTDO33:CHAR,I,22,"(D.BLUE C-)":CHAR,I,24
,"(YL.GRN C-)":NEXT <52>
210 FORI=33TD0STEP-1:CHAR,I,22,"(SPACE)":CHAR,I,
24,"(SPACE)":NEXT <34>
220 CHAR,8,22,"(D.BLUE C-)":CHAR,8,24,"(YL.GRN C
-)":PRINT"(WHITE HOME)" <74>
230 CHAR,21,8,"O":K=5 <222>
240 CHAR,17,8,RIGHT*(STR*(RU),1) <203>
250 POKE211,0:SYS13312:POKE212,0:SYS13328 <246>
260 SYS14400:FORI=OTD500:NEXT:FORI=BTDO0STEP-1:VD
LI:SOUND1,600,5:SOUND2,800,5:NEXT <103>
270 CHAR,17,15,"(D.BLUE S* SA SB DOWN LEFT3 SC S
D SE DOWN LEFT3 SF SG SH WHITE)" <171>
280 POKE1339,GF(8):CHAR,20,15,"(SI SJ SK DOWN LE
FT3 SL SM SN DOWN LEFT3 SO SP SQ WHITE)":FORI=OT
D500:NEXT:SYS14336 <143>
290 IFS1THENJ1=INT(RND(1)*8)+1:GOTO310 <35>
300 J1=JOY(1):IFJ1>127THENJ1=J1-128 <184>
310 IFS2THENJ2=INT(RND(1)*8)+1:GOTO340 <130>
320 J2=JOY(2):IFJ2>127THENJ2=J2-128 <6>
330 GOTO360 <156>
340 SZ=SZ-1:IFSZTHENJ2=(INT(RND(1)*4)+1)-1:GOTO3
60 <27>
350 SZ=5-B:GOTO360 <38>
360 POKE211,J1:SYS13312:POKE212,J2:SYS13328 <12>
370 TU=J1+1:ONTUBGOTO560,560,390,430,470,560,510,
560,560 <188>
380 GOTO560 <210>
390 IFJ2=0THENP2=P2+1:GOTO560 <20>
400 IFJ2=8THENP2=P2+1:GOTO560 <70>
410 IFJ2=4THENP2=P2+1:GOTO560 <60>
420 GOTO560 <250>
430 IFJ2=0THENP2=P2+1:GOTO560 <60>
440 IFJ2=7THENP2=P2+1:GOTO560 <105>
450 IFJ2=4THENP2=P2+1:GOTO560 <100>
460 GOTO560 <34>
470 IFJ2=0THENP2=P2+1:GOTO560 <100>
480 IFJ2=6THENP2=P2+1:GOTO560 <140>
490 IFJ2=4THENP2=P2+1:GOTO560 <140>
500 GOTO560 <74>
510 IFJ2=0THENP2=P2+1:GOTO560 <140>
520 IFJ2=8THENP2=P2+1:GOTO560 <191>
530 IFJ2=7THENP2=P2+1:GOTO560 <196>

```

```

540 IFJ2=5THENP2=P2+1:GOTO560 <196>
550 GOTO560 <125>
560 TU=J2+1:ONTUBGOTO750,750,750,750,700,750,660,
620,580 <124>
570 GOTO750 <146>
580 IFJ1=0THENP1=P1+1:GOTO750 <204>
590 IFJ1=4THENP1=P1+1:GOTO750 <234>
600 IFJ1=6THENP1=P1+1:GOTO750 <254>
610 GOTO750 <186>
620 IFJ1=0THENP1=P1+1:GOTO750 <244>
630 IFJ1=3THENP1=P1+1:GOTO750 <13>
640 IFJ1=6THENP1=P1+1:GOTO750 <38>
650 GOTO750 <226>
660 IFJ1=0THENP1=P1+1:GOTO750 <28>
670 IFJ1=2THENP1=P1+1:GOTO750 <48>
680 IFJ1=6THENP1=P1+1:GOTO750 <78>
690 GOTO750 <10>
700 IFJ1=0THENP1=P1+1:GOTO750 <68>
710 IFJ1=2THENP1=P1+1:GOTO750 <88>
720 IFJ1=3THENP1=P1+1:GOTO750 <103>
730 IFJ1=5THENP1=P1+1:GOTO750 <123>
740 GOTO750 <60>
750 IFP1<26THENCHAR,B+P1,22,"(D.BLUE C- WHITE)":
ELSE1760 <200>
760 IFP2<26THENCHAR,B+P2,24,"(YL.GRN C- WHITE)":
ELSE1680 <210>
770 POKE211,0:SYS13312:POKE212,0:SYS13328 <1>
780 K=K+1:IFK=6THENK=0:ZE=ZE-1:GOTO800 <90>
790 GOTO810 <101>
800 CHAR,23,8,RIGHT*(STR*(ZE),2) <145>
810 IFZETHEN290 <119>
820 SYS14400:FORI=OTD250:NEXT:FORI=BTDO0STEP-1:VD
LI:SOUND1,600,5:SOUND2,800,5:NEXT <184>
830 FORN=OTD150:NEXT <159>
840 GOSUB1120 <133>
850 RU=RU+1:IFRU<4THEN180 <242>
860 REM <238>
870 BOTO1800 <172>
880 POKE65298,195:POKE65287,8:END <172>
890 PRINT"(CLEAR SPACE12 YL.GRN ST15 YL.GRN <147>
900 PRINT"(DOWN SPACES CA SY21 CS <222>
910 PRINT"(SPACES SZ CA SY19 CS SZ <184>
920 PRINT"(SPACES SZ2 YELLOW)1:(YL.GRN SPACE17 S
Z <234>
930 PRINT"(SPACES SZ2 SPACE19 SZ <50>
940 PRINT"(SPACES SZ2 YELLOW)2:(YL.GRN SPACE17 S
Z <12>
950 PRINT"(SPACES SZ2 SPACE19 SZ <70>
960 PRINT"(SPACES SZ2 SPACE19 SZ <80>
970 PRINT"(L.BLU SR YL.GRN SPACE7 SZ2 SHIFTSPEAC
E SZ2 L.BLU SPACE7 SS <19>
980 PRINT"(YL.GRN SR2 YL.GRN SPACE6 SZ CZ SY19 C
X SZ YL.GRN SPACE6 SS <225>
990 PRINT"(D.BLUE SR3 YL.GRN SPACES CZ CR2 SY17
CR2 CX D.BLUE SPACES SS <199>
1000 PRINT"(BL.GRN SR4 D.BLUE SY5 CE2 SY17 CE2 S
Y5 BL.GRN SS <242>
1010 PRINT"(BRN SR5 D.BLUE S*22 BRN SS <200>
1020 PRINT"(PINK SR6 D.BLUE S*7 SPACE13 S*7 PINK
SS <184>
1030 PRINT"(L.BLU SR7 D.BLUE S*5 SPACE15 S*5 L.B
LU SS <124>
1040 PRINT"(YL.GRN SR8 D.BLUE S*3 SPACE L.GRN SV
SPACE13 SH D.BLUE SPACE S*3 YL.GRN SS <230>
1050 PRINT"(BL.GRN SR2 SPACE3 L.GRN SV SPACE13 S
H BL.GRN SPACE3 SS <122>
1060 PRINT"(BL.GRN SU9 SPACE3 L.BLU SU15 SPACE3
BL.GRN SU <189>
1070 PRINT"(BL.GRN SU9 SPACE3 ORNG SU15 SPACE3 B
L.GRN SU <71>
1080 PRINT"(BRN SX39)" <102>
1090 PRINT"(YELLOW DOWN)BOXER(SPACE)1:(SPACE26 G
REEN)K.D." <250>
1100 PRINT"(CYAN DOWN)BOXER(SPACE)2:(SPACE26 PUR
PLE)K.D.(HOME)" <170>
1110 RETURN <232>
1120 SYS14400 <224>
1130 I=0:POKE65296,126:POKE65297,68 <77>
1140 RESTORE1170:FORN=OTD15:READZ:POKE12944+N,Z:
NEXT:POKE65295,INT(RND(1)*64)+1 <252>
1150 FORN=OTD15:READZ:POKE12944+N,Z:NEXT <101>
1160 I=I+1:IFI<5THEN1140:ELSEPOKE65297,0:RETURN <69>
1170 DATA0,0,20,88,94,88,104,48 <27>
1180 DATA0,0,20,37,181,37,41,12 <169>
1190 DATA20,88,94,88,104,48,252,252 <191>
1200 DATA20,37,181,37,41,12,63,63 <172>

```



```

1210 TRAP2200:SYS14400:SYS14336:P=0:B1=0:B2=1:FO
RI=0T016:CHAR,28-1,4,"(SPACE)":NEXT
1220 FORI=0T016:CHAR,28-1,6,"(SPACE)":NEXT
1230 CHAR,13,4,"(GREEN)I(SPACE)PLAYER(WHITE)":B=
1:BZ=4
1240 POKE211,0:SYS13312:POKE212,0:SYS13328
1250 CHAR,17,15,"(D.BLUE)S* SA SB DOWN LEFT3 SC
SD SE DOWN LEFT3 SF SG SH WHITE)"
1260 POKE1339,BF(B):CHAR,20,15,"(SI SJ SK DOWN L
EFT3 SL SM SN DOWN LEFT3 SO SP SQ WHITE)"
1270 CHAR,9,23,"(SPACE24)":RU=1
1280 DATA(SPACE)H&H(SPACE)BOFT(SPACE)PRESENTS(S
PACE)
1290 DATA(SPACE)*(SPACE)'THE(SPACE)CHAMPS'(SPAC
E)*(SPACE2)
1300 DATA(SPACE)A(SPACE)BOXING-GAME(SPACE)FOR(S
PACE)
1310 DATA(SPACE)THE(SPACE)COMMODORE(SPACE)16(SP
ACE2)
1320 DATA(SPACE)FUNKION(SPACE)KEY'S:(SPACE3)
1330 DATA(SPACE)'*'=CHOOSE(SPACE)PLAYER(SPACE)"
1340 DATA(SPACE)'S'=START(SPACE)GAME(SPACE4)
1350 DATA(SPACE)PRESS(SPACE)'RUN/STOP'(SPACE2)
1360 DATA(SPACE)FOR(SPACE)NEW(SPACE)NAME(SPACE6
)
1370 DATA(SPACE)YOU(SPACE)MUST(SPACE)DEFEAT(SPA
CE3)
1380 DATA(SPACE)4(SPACE)OPPONENTS(SPACE):(SPACE
5)
1390 DATA(SPACE)1.(SPACE)TEETHLESS(SPACE)SAM(SP
ACE2)
1400 DATA(SPACE)-(SPACE)GREAT(SPACE)BRITAIN(SPA
CE3)
1410 DATA(SPACE)2.(SPACE)JOHN(SPACE)'MILKY'(SPA
CE3)
1420 DATA(SPACE)-(SPACE)TEXAS(SPACE)/(SPACE)USA
(SPACE)
1430 DATA(SPACE)3.(SPACE)LUIGI(SPACE)KLANOTTI(S
PACE)
1440 DATA(SPACE)-(SPACE)ITALY(SPACE1)
1450 DATA(SPACE)4.(SPACE)SERGEJ(SPACE)BRECHOV(S
PACE)
1460 DATA(SPACE)-(SPACE)SOVIETUNION(SPACE)
1470 DATA(SPACE)THEY(SPACE)ALL(SPACE)FIGHT(SPA
CE4)
1480 DATA(SPACE)VERY(SPACE)WELL(SPACE)!!!(SPACE
5)
1490 DATA(SPACE)*** (SPACE)GOOD(SPACE)LUCK(SPACE
)*** (SPACE)
1500 DATA(SPACE)PRESS'S'TO(SPACE)START(SPACE2)
1510 DATA(SPACE)FOLLOWING(SPACE)A(SPACE)DEMO(SP
ACE2)
1520 RESTORE1280:PRINT(WHITE)":FORN=0T023:READL
$
1530 FORI=0T018:CHAR,28-1,8,LEFT*(L$,I+1):NEXT:F
ORN=0T0150
1540 GETT$:IFT$="S" THENGOSUB1630:T=60:GOTO170
1550 IFT$="*" THENGOSUB1600
1560 NEXT
1570 FORI=18T00STEP-1:CHAR,10,8,RIGHT*(L$,I+1):N
EXT:NEXT:T=30:B1=1:B2=1
1580 CHAR,9,23,"(YELLOW)PRESS(SPACE)'RUN/STOP'(S
PACE)TO(SPACE)END(WHITE)":TRAP1210
1590 GOSUB1630:GOTO170
1600 P=1-P
1610 IFF THENCHAR,13,4,"(SPACEB)":CHAR,13,6,"(GRE
EN)2(SPACE)PLAYER(WHITE)":S1=0:S2=0:RETURN
1620 CHAR,13,6,"(SPACEB)":CHAR,13,4,"(GREEN)I(S
PACE)PLAYER(WHITE)":S1=0:S2=1:RETURN
1630 L$="(SPACE)ROUND:(SPACE3)01:00(SPACE3)"
1640 FORI=0T018:CHAR,28-1,8,LEFT*(L$,I+1):NEXT:C
HAR,10,8,"(SPACE RED)ROUND:(WHITE SPACE3)01:00"
1650 RETURN
1660 POKE1339,BF:FORI=0T016:CHAR,28-1,4,LEFT*(N$,
I+1):NEXT:PRINT(WHITE)":RETURN
1670 POKE1339,BF(B)+B:FORI=0T016:CHAR,28-1,6,LEF
T*(B*(B),I+1):NEXT:PRINT(WHITE)":RETURN
1680 SYS14400:TM=TI
1690 POKE212,9:SYS13328:FORI=0T0500:NEXT:POKE212
,10:SYS13328
1700 FORI=BT00STEP-1:VOLI: SOUND3,300,3:NEXT
1710 POKE211,11:SYS13312:FORI=0T0100:NEXT
1720 GOSUB1120:RU=1:B=B+1:IFB<4 THEN180
1730 POKE211,11:SYS13312:GOSUB1120:POKE212,12:SY
S13328:GOSUB1120:GOSUB1120
1740 IFS1=0ANDS2=1ANDTM<NM(10) THEN1830
1750 GOTO1210
1760 SYS14400
1770 POKE211,9:SYS13312:FORI=0T0500:NEXT:POKE211
,10:SYS13312
1780 FORI=BT00STEP-1:VOLI: SOUND3,300,3:NEXT:FORI
=0T0500:NEXT
1790 POKE212,11:SYS13328:GOSUB1120:POKE211,12:SY
S13312:GOSUB1120:GOSUB1120:GOTO1210
1800 IFF2>P1 THENSYS14400:GOTO1710
1810 IFF1>P2 THENSYS14400:GOTO1790
1820 IFF1=P2 THENSYS14400:GOTO180
1830 FORI=0T0500:NEXT
1840 FORI=0T012:POKE2023,12-1:POKE2022,12-1:POKE
2024,27+I:POKE2021,12+I
1850 SCNCLR
1860 NEXT
1870 SYS14400
1880 L$(0)="(LEFT7)DEINE(SPACE)INITIALEN"
1890 L$(1)="(UP LEFT UP LEFT UP LEFT UP LEFT UP
LEFT UP LEFT UP LEFT)D(DOWN)E(DOWN)I(DOWN)N(DOWN
)E(DOWN SPACE DOWN)I(DOWN)N(DOWN)I(DOWN)T(DOWN)I
(DOWN)A(DOWN)L(DOWN)E(DOWN)N"
1900 L$(2)="(UP)D(DOWN LEFT)E(DOWN LEFT)I(DOWN
LEFT)N(DOWN LEFT)E(DOWN LEFT SPACE DOWN LEFT)I(D
OWN LEFT)N(DOWN LEFT)I(DOWN LEFT)T(DOWN LEFT)I(D
OWN LEFT)A(DOWN LEFT)L(DOWN LEFT)E(DOWN LEFT)N"
1910 L$(3)="(LEFT7 DOWN)N(UP)E(UP)L(UP)A(UP)I(UP
)T(UP)I(UP)N(UP)I(UP SPACE UP)E(UP)N(UP)I(UP)E(
UP)D"
1920 L$(4)="(LEFT7)NELAITINI(SPACE)ENIED"
1930 L$(5)="(LEFT7 UP)N(DOWN)E(DOWN)L(DOWN)A(DO
WN)I(DOWN)T(DOWN)I(DOWN)N(DOWN)I(DOWN SPACE DOWN
)E(DOWN)N(DOWN)I(DOWN)E(DOWN)D"
1940 L$(6)="(DOWN7)D(UP LEFT)E(UP LEFT)I(UP LEFT
)N(UP LEFT)E(UP LEFT SPACE UP LEFT)I(UP LEFT)N(UP
P LEFT)I(UP LEFT)T(UP LEFT)I(UP LEFT)A(UP LEFT)L
(UP LEFT)E(UP LEFT)N"
1950 L$(7)="(LEFT7 DOWN)D(UP)E(UP)I(UP)N(UP)E(UP
P SPACE UP)I(UP)N(UP)I(UP)T(UP)I(UP)A(UP)L(UP)E(
UP)N":B=1:POKE65287,B
1960 I=0
1970 SCNCLR:CHAR,19,12,L$(1)
1980 FORN=0T09:NEXT
1990 IFF<7 THENI=I+1:GOTO1970
2000 IFF<8 THEND=D+20:GOTO1960
2010 SCNCLR:CHAR,19,12,L$(0)+"(SPACE):(SPACE YEL
LOW)" + IN$
2020 FORI=0T01500:NEXT:FORI=0T011:PRINT(NONE)"C
HR$(27)"V":NEXT
2030 CHAR,16,2,"(RED)TOP(SPACE)TEN(SPACE)":
2040 FORI=1T09:CHAR,11,3+2*I,"(YELLOW)" + STR$(I)+
".":NEXT
2050 CHAR,11,23,"(YELLOW)10."
2060 NM(10)=TM:NM$(10)=IN$
2070 GOSUB2120
2080 FORI=1T010:COLOR1,3+I,4:CHAR,16,3+2*I,NM$(I
):NEXT
2090 PRINT(WHITE)":FORI=1T010:CHAR,21,3+2*I,STR
$(NM(I)):NEXT
2100 CHAR,4,7,"(GREEN)P(DOWN LEFT)R(DOWN LEFT)E(
DOWN LEFT)S(DOWN LEFT)S(DOWN LEFT SPACE DOWN LEF
T)K(DOWN LEFT)E(DOWN LEFT)Y(DOWN LEFT SPACE DOWN
LEFT) (DOWN LEFT)S(DOWN LEFT)'(WHITE)"
2110 GETT$:IFT$<"S" THEN2110:ELSE100
2120 CC=0
2130 FORI=1T09:IFNM(I+1)>NM(I) THEN2180
2140 N=NM(I):H$=NM$(I)
2150 NM(I)=NM(I+1):NM$(I)=NM$(I+1)
2160 NM(I+1)=N:NM$(I+1)=H$
2170 CC=1
2180 NEXT
2190 IFCC>0 THEN2120:ELSERETURN
2200 SYS14400:TRAP2200
2210 FORI=0T012:POKE2023,12-1:POKE2022,12-1:POKE
2024,27+I:POKE2021,12+I
2220 SCNCLR
2230 NEXT:POKE65287,B
2240 POKE19,1:INPUT(WHITE CLEAR)ENTER(SPACE)YOU
R(SPACE)NAME(SPACE)":N$:POKE19,0
2250 IFFN$="END" THENEND
2260 IFFLEN(N$)>3 THENIN$=LEFT$(N$,3):ELSEIN$=N$
2270 IFFLEN(N$)>16 THENN$=LEFT$(N$,15)+"(SPACE)":
GOTO100
2280 FORI=LEN(N$)+1T017:N$=N$+"(SPACE)":NEXT
2290 GOTO100
ENDE DES LISTINGS

```


Der Checksummer für alle C-16/116 - oder Plus 4-Besitzer!

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

SYS 62158 : SYS 33047

Aktiviert wird der Checksummer mit SYS 818. Die Funktion und Arbeitsweise entnehmen Sie bitte der C64-Checksummer-Version (siehe Ausgabe Computronic Nr. 3). Eine Ausnahme bei C-16/116 sind

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

Arbeitsweise und Aufbau unseres Checksummers:

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

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

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

Zahl im Heft hinter der Zeile. Stimmt die Zahl überein, ist die Zeile richtig eingegeben.

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

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

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

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

Achtung: nach dem Starten des Checksummers muß noch »NEW (RETURN)« eingegeben werden.

```

1 REM ***** <4>
2 REM * * <99>
3 REM * C16-CHECKSUMMER * <140>
4 REM * * * <101>
5 REM * COPYRIGHT BY FRANK BRALL * <182>
6 REM * * * <103>
7 REM * (C) 18.04.86 * <181>
8 REM * * * <105>
9 REM ***** <12>
10 PRINT "CLEAR DOWN SPACE2" ***** <103>
CHECKSUMMER(SPACE)OC(SPACE)1.0(SPACE) **
** <DOWN> "
11 PRINT "SPACE2" ** <SPACE2> COPYRIGHT (SPA
CE)FRANK(SPACE)BRALL(SPACE3) ** <DOWN> " <110>
12 PRINT "SPACE8" FUER(SPACE)TRONIC-VERLA
B" <196>
60000 DIMH(75):FORI=0TD9 <123>
60010 H(4B+I)=I:H(65+I)=I+10:NEXT <145>
60020 FORI= B18 TO 1010 :READA# <151>
60030 H=ASC(LEFT*(A#,1)) <234>
60040 L=ASC(RIGHT*(A#,1)) <253>
60050 D=H(H)*16+H(L):S=B+D:POKEI,D <13>
60060 A=A+1:IFA<20THENNEXT:A=-1 <97>
60070 READV:Z=Z+1:IFV=BTHEN60085 <177>
60080 PRINT"DATAFEHLER(SPACE)IN(SPACE)ZE
ILE(SPACE):":60200+Z:END <39>
60085 IFA<0THEN60100 <116>
60090 S=0:A=0:NEXT <246>
60100 PRINT"DOWN2" <172>
60110 PRINT"CHECKSUMMER(SPACE)EIN(SPACE)
=(SPACE)SYS(SPACE)818(DOWN)" <140>
60120 PRINT"CHECKSUMMER(SPACE)AUB(SPACE)
=(SPACE)SYS(SPACE)62158+8YS(SPACE)33047<
DOWN" <122>

```

```

60130 PRINT"DOWN)SAVE/LOAD(SPACE)IST(SPA
CE)WAEREND(SPACE)DER(SPACE)CHECK-" <61>
60140 PRINT"SUMMER(SPACE)AKTIV(SPACE)IST
,(SPACE)NICHT(SPACE)MOEGLICH(SPACE)!" <221>
60150 PRINT"DOWN)BEACHTEN(SPACE)SIE(SPA
CE)DIE(SPACE)HINWEISE(SPACE)IN(SPACE)DEN
" <189>
60170 PRINT"HEFTEN(SPACE)COMPUTE(SPACE)M
IT(SPACE)UND(SPACE)COMPUTRONIC" <234>
60180 PRINT"DOWN)VIEL(SPACE)SPASS(SPACE)
):(SPACE)(AUTOR:F.BRALL764438ONTRA) <116>
60190 POKE B14,147:POKEB15,251:POKE B16,
147:POKE B17,251 <109>
60195 SYS 818:END <177>
60201 DATA A9,3D,8D,02,03,A9,03,8D,03,03
,60,A2,FF,86,3A,20,5A,8B,86,3B,1851 <56>
60202 DATA B4,3C,20,73,04,AA,F0,EF,90,09
,20,53,89,20,79,04,4C,D9,8B,20,2018 <28>
60203 DATA 3E,8E,20,53,89,84,0B,A9,00,8D
,E4,03,8D,E5,03,8D,E6,03,18,A5,2076 <40>
60204 DATA 14,65,15,8D,E6,03,A0,FF,CB,B1
,3B,F0,2C,C9,22,D0,0A,AD,E4,03,2508 <18>
60205 DATA 49,01,8D,E4,03,B1,3B,AE,E4,03
,D0,04,C9,20,F0,E4,EE,ES,03,AE,2644 <87>
60206 DATA E5,03,18,B1,3B,6D,E6,03,8D,E6
,03,CA,D0,F4,4C,76,03,38,20,39,2204 <107>
60207 DATA DB,8E,E2,03,BC,E3,03,A2,00,A0
,00,18,20,39,DB,A9,5B,20,D2,FF,2365 <115>
60208 DATA A9,00,AE,E6,03,20,5F,A4,A9,5D
,20,D2,FF,20,0B,FB,20,0B,FB,AC,2380 <169>
60209 DATA E3,03,AE,E2,03,18,20,39,DB,A2
,00,86,FF,4C,36,87,00,00,00,00,1778 <133>
60210 DATA 00,00,00,00,00,00,00,00,00,00
,00,00,00,0 <104>

```


Kennen Sie schon?

Doppelausgabe Nov./Dez. '86 Nr. 6 4. Jahrgang

Computronia

+ Homecomputer
Software
Für Homecomputer

OS 55,- Str 6,50 DM 6,50

Der verfluchte Punkt-Gewinnspiel

Wo? Seite 3

CR64	Wick
C-16	IT-9644
Schneider	apple
ATARI	Spectrum

1/87

checksummer

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 Prüfzahl. 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>
1 REM VERSION C 1.0 BEI FRANK BRALL <10>
2 REM ----- <148>
3 FOR I=832 TO 1008:READ A:S=S+A <6B>
4 POKE I,A:NEXT I:IF S<>16397 THEN PRINT
"FEHLER(SPACE) IN(SPACE)DATA-ZEILEN(SPACE
)1(SPACE) (10-19)":STOP <32>
5 SYS 832:S=0:FOR I=58464 TO 58603:READ
A:S=S+A <21>
6 POKE I,A:NEXT I:IF S<>18919 THEN PRINT
"FEHLER(SPACE) IN(SPACE)DATA-ZEILEN(SPACE
)1(SPACE) (29-36)":STOP <219>
7 S=0:FOR I=48979 TO 49005:READ A:S=S+A <33>
8 POKE I,A:NEXT I:IF S<>2888 THEN PRINT
"FEHLER(SPACE) IN(SPACE)DATA-ZEILEN(SPACE
)1(SPACE) (46-47)":STOP <23>
9 PRINT"(DOWN2)AKTIVIEREN(SPACE):(SPACE)
POKE(SPACE)1,53(SPACE18)AUSSCHALTEN:(SPA
CE)POKE(SPACE)1,55":NEW <26>
10 DATA169,0,133,254,162,1,189,137,3,133
,255,160,0,177,254,145,254,136 <176>
11 DATA208,249,230,255,165,255,221,139,3
,208,238,202,16,230,169,96,141,49 <147>
12 DATA165,169,228,141,50,165,169,53,133
,1,169,141,133,254,162,3,134,255 <225>
13 DATA160,0,177,254,240,7,32,202,241,20
0,76,120,3,169,2,141,32,208 <140>
14 DATA96,160,224,192,0,147,17,32,32,35,
35,35,35,32,84,82,79,78 <158>
15 DATA73,67,45,86,69,82,76,65,71,32,67,
72,69,67,75,83,85,77 <129>
16 DATA77,69,82,32,35,35,35,35,139,13,13
,32,32,32,32,32,32 <147>
17 DATA32,32,32,32,32,86,69,82,83,73,79,
78,32,67,32,49,46,48 <79>
18 DATA13,13,32,32,32,32,32,32,40,67,41,
32,70,82,65,78,75,32 <94>
19 DATA66,82,65,76,76,32,40,49,48,46,56,
53,41,13,0 <12>
20 DATA169,0,141,107,191,141,108,191,141
,109,191,160,2,24,177,95,200,113 <7>
21 DATA95,141,109,191,160,3,200,177,95,2
,40,44,201,34,208,10,173,107,191 <91>
22 DATA73,1,141,107,191,177,95,174,107,1
91,208,4,201,32,240,228,238,108 <128>
23 DATA191,174,108,191,24,177,95,109,109
,191,141,109,191,202,208,244,76,120 <244>
24 DATA228,56,76,183,228,72,32,201,255,1
70,104,144,1,138,96,32,240,255 <101>
25 DATA142,105,191,140,106,191,162,0,160
,0,24,32,240,255,169,91,32,12 <89>
26 DATA225,169,0,174,109,191,32,205,189,
169,93,76,83,191,173,33,208,145 <239>
27 DATA243,96,105,2,164,145,200,208,4,19
7,161,208,247,96 <202>
28 DATA32,12,225,32,63,171,32,63,171,172
,106,191,174,105,191,24,32,240 <213>
29 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

PYRAMID



Meinst du, wir finden das Geheimnis der Pyramiden?



geschafft, müssen Sie nur noch in die vier Diagonalen laufen und Sie sind der neue Pharao!

Im nächsten Level haben sie natürlich weniger Zeit zur Verfügung, um die Ihnen gestellte Aufgabe zu meistern. Auch andere Gefahren stellen sich noch zusätzlich ein!?

Sie dürfen übrigens weder Spinne, Monster oder einen Stein berühren, dann ist Ihre Mission vorzeitig beendet!

Die tausend Leiden eines werdenden Pharao

Stellen Sie sich vor, Sie befinden sich im alten Ägypten und sind in der engeren Auswahl für die Nachfolge des eben verstorbenen Pharaos. Alle Bewerber werden einer harten Probe unterworfen. Ihre Aufgabe ist es, in eine Pyramide einzudringen. Und das ist wirklich nicht so einfach. Herab-

fallende Steine erschweren das Erklimmen der Pyramide. Die einzige Möglichkeit, die es gibt, ist, sich ganz nahe an die Pyramidenwand zu stellen. Haben Sie das geschafft, können Sie in die Pyramide eindringen. Dort allerdings erwarten sie zwei Spinnen, mehrere Totenköpfe und Mumien. Die

Totenköpfe und Mumien müssen Sie einsammeln, aber den Spinnen sollten Sie aus dem Weg gehen. Haben Sie alles eingesammelt, so wird (unten links) ein Weg freigegeben, der in die tiefste Ebene führt. Dort müssen Sie ein Monster überwinden und sich durch die Grabkammer graben. Ist dies

Tips zur Steuerung

Zum Erklimmen der Pyramide Joystick nach links

Zum Zerhacken der Grabkammer Joystick und Trigger nach rechts

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

```

0 REM *****
1 REM * BACKGROUND & VORPROGRAMM *
2 REM *****
3 FORI=1536TO1536+173
4 READA
5 POKEI,A
6 NEXTI
7 FORI=1712TO1712+119STEP3
8 READA,B
9 POKEI,A:POKEI+1,B-(INT(B/256)*256)
10 POKEI+2,INT(B/256)
11 D=D+A*B
12 NEXTI
13 POKE208,176:POKE209,6
14 VOL4:SYS1536
15 :
16 DATA 120,169,36,141,20,3,169,6,141,21,3,169,2
55,141,252,4
17 DATA 141,254,4,88,96,120,169,14,141,20,3,169,
206,141,21,3
18 DATA 88,96,255,0,173,252,4,201,255,240,3,76,1
4,206,160,0
19 DATA 162,0,177,208,149,210,200,232,138,201,3,
208,245,165,208,105
20 DATA 2,133,208,144,2,230,209,165,210,201,255,
208,6,32,21,6
21 DATA 24,144,216,201,254,208,19,165,208,133,21
3,165,211,133,208,165
22 DATA 209,133,214,165,212,133,209,24,144,196,2
01,253,208,11,165,213
23 DATA 133,208,165,214,133,209,24,144,181,201,2
52,208,13,173,17,255
24 DATA 41,240,5,211,141,17,255,24,144,164,73,25
5,141,252,4,169
25 DATA 255,141,254,4,173,18,255,41,252,5,212,14
1,18,255,165,211
26 DATA 141,14,255,173,17,255,9,16,141,17,255,24
,144,162
27 :
28 DATA 254,1727,254,1796,254,1796,254,1727,254,
1712
29 DATA 10,685,10,739,20,739,10,685,10,643,20,59
6,10,643,10,685
30 DATA 10,739,10,685,40,643,10,685,10,739,20,73
9
31 DATA 10,685,10,643,20,596,10,643,10,685,10,64
3
32 DATA 10,596,40,596,253,0
33 DATA 10,770,10,810,20,810,10,796,10,739,20,77
0
34 DATA 10,770,10,810,10,798,10,739,40,770,253,0
35 COLOR0,1:COLOR4,3,3:GRAPHIC1,1
36 COLOR1,2,7
37 DRAW1,100,100TO180,100
38 DRAW1,100,100TO130,70

```

Handwritten annotations in the listing include: '174' next to line 4, '0600' and '0620' next to line 22, '0480' and '3232' next to line 25, and '80' next to line 9.

programme

```

39 DRAW1,130,70TD210,70
40 DRAW1,210,70TD180,100
41 DRAW1,100,100TD155,15
42 DRAW1,180,100TD155,15
43 DRAW1,130,70TD155,15
44 DRAW1,210,70TD155,15
45 COLOR1,7,4
46 CHAR1,14,13,"PYRAMID"
47 CHAR1,12,15,"VORPROGRAMM"
48 CHAR1,2,17,"WRITTEN(SPACE)1986(SPACE)BY(SPACE)
BJOERN(SPACE)THIEMANN"
49 CHAR1,13,19,"DRUECKE(SPACE)F1"
50 KEY1,"LOAD"+CHR$(13)
51 FORT=1TD3000:NEXT
52 GRAPHICCLR
53 NEW
ENDE DES LISTINGS

```

Teil 2

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

```

0 TRAP 277
1 REM
2 REM
3 REM
4 REM
5 REM
6 REM
7 REM
8 REM
9 REM
10 REM
11 REM
12 REM
13 REM
14 REM
15 REM
16 ZF=3:LE=1:TM=0:VOL4
17 COLOR,1:COLOR4,1
18 RESTORE
19 POKE51,0:POKE52,60:POKE55,0:POKE56,50:POKE652
98,0:POKE65299,60
20 FORT=15300TD15333
21 READA1:POKET,A1:NEXT
22 DATA169,000,133,003,133,005,169,211,133,004,1
69,063,133,006,160,000,177
23 DATA003,145,005,136,208,249,198,004,198,006,1
65,006,201,059,208,237,096
24 BY815300
25 FORT=0TO8:READZN
26 V1=15360+8*ZN:FORC1=0TD7
27 READH1:POKEV1+C1,H1:NEXT:NEXT
28 DATA83,60,90,255,189,195,235,36,102
29 DATA102,180,190,255,230,150,121,210,130
30 DATA42,255,0,255,0,255,0,255,0
31 DATA35,1,2,4,8,16,32,64,128
32 DATA0,60,90,255,195,126,36,36,231
33 DATA65,126,153,153,231,102,60,36,24
34 DATA86,60,66,126,60,36,66,129,0
35 DATA81,56,60,126,127,153,126,253,120
36 DATA88,24,60,60,24,126,90,24,60
37 X=38:Y=17:M1=23:M2=1:G1=20:S2=14:G1=15:G2=12:
W1=31:W2=23:T1=0:TS=500-TM
38 D=0:U=0
39 REM SPIELFELD
40 SCNCLR
41 L$=STR$(LE)
42 CHAR1,0,1,"(SPACE2 RED SPACE)PYRAMID(D.BLUE S
U SI SPACE18 RED)LEVEL"+L$
43 CHAR1,0,2,"(SPACE D.BLUE SU SC2 SK SPACE4 BRN
SPACE C+3 SPACE C+3"
44 CHAR1,0,3,"(D.BLUE SPACE S2 SK SPACE13 BRN SP
ACE C+3 SPACE C+3 SPACE6 GREEN)PHASE"
45 CHAR1,0,4,"(SPACE15 BRN C+5 SPACE4 C+3"
46 CHAR1,0,5,"(D.BLUE)((WHITE SA D.BLUE))(SPACE)
1985(SPACE)BY(SPACE BRN SPACE3 C+5 SPACE C+2 SPA
CE C+2 SPACE4 BLUE)SCORE"
47 CHAR1,0,6,"(SPACE13 BRN C+7 SPACE C+2 SPACE C
+4)"
48 CHAR1,0,7,"(YL.GRN)EAGLE-SOFT(SPACE BRN SPACE
2 C+2 SPACE6 C+2 SPACE C+4 SPACE3)CLIMBING

```

```

<92>
<118>
<44>
<85>
<188>
<183>
<234>
<226>
<215>
<22>
<222>
<199>
<246>
<74>
<215>
49 CHAR1,0,8,"(SPACE11 C+4 SPACE WHITE SA BRN C+
3 SPACE C+2 SPACE C+6)"
50 CHAR1,0,9,"(SPACE11 C+4 SHIFTSPACE C+4 SPACE
C+2 SPACE2 C+5)"
51 CHAR1,0,10,"(SPACE9 C+5 SPACES C+ SPACE C+3 S
PACE PURPLE SX BRN C+3"
52 CHAR1,0,11,"(SPACE9 C+5 SPACE C+3 SPACE C+ SP
ACE C+3"
53 CHAR1,0,12,"(SPACE7 C+3 SPACES C+3 SPACE C+ S
PACE9 C+5)"
54 CHAR1,0,13,"(SPACE7 C+3 SPACE C+3 SPACE C+3 S
PACE C+ SPACE C+6 SPACE CYAN SX BRN C+5)"
55 CHAR1,0,14,"(SPACE5 C+5 SPACE C+3 D.BLUE SX B
RN C+3 WHITE SA BRN C+8 WHITE SA SPACE BRN C+7)"
56 CHAR1,0,15,"(SPACE3 C+3)"
57 CHAR1,0,16,"(SPACE3 C+3 SPACE29 C+3"
58 CHAR1,0,17,"(SPACE3 C+3 SPACE4 C+6 SPACE6 C+4
SPACE6 PINK)****(BRN C+3"
59 CHAR1,0,18,"(SPACE C+3 SPACE8 C+ SPACE9 C+4 S
PACE2 C+2 SPACE2 PINK)****(BRN C+3"
60 CHAR1,0,19,"(SPACE C+ SPACE4 C+ SPACES C+ SPA
CE11 C+2 SPACE2 C+ SPACE3 PINK)***(SPACE4 BRN C+
61 CHAR1,0,20,"(SPACE C+ SPACE4 C+ SPACES C+ SPA
CE11 C+2 SPACE2 C+ SPACE3 PINK)***(SPACE4 BRN C+
62 CHAR1,0,21,"(SPACE C+ SPACE4 C+ SPACES C+ SPA
CE8 C+ SPACE10 PINK)***(SPACE PURPLE)##(SPACE BR
N C+3"
63 CHAR1,0,22,"(SPACE C+ SPACE4 C+ SPACE14 C+ SP
ACE10 PINK)***(SPACE PURPLE)##(SPACE BRN C+3"
64 CHAR1,0,23,"(SPACE C+ SPACE30 PINK)***(SPACE4
BRN C+3"
65 CHAR1,0,24,"(SPACE C+38)"
66 CHAR1,35,3,"(SPACE)1"
67 POKE4071,102
68 REM PHASE 1
69 J=JOY(1)
70 BETA$:IF A$="(SPACE)"THENGETKEYB$
71 IFJ=7 THENP=P+1:B=B+50:GOSUB84
72 TS=TS-1:IFTS=0THEN97
73 PRINT"(RED)":P$=STR$(P):CHAR1,35,5,P$
74 CHAR1,X,Y,"(BLUE SS)"
75 IFX=20ANDJ=5THENGOSUB103
76 ZS=INT(RND(1)*ZF)+1
77 IFZ8=1THENA=1
78 IFA=1THENCHAR1,M1,M2,"(SPACE)":M1=M1+2:CHAR1,
M1,M2,"(SPACE)":M2=M2+2:VOL8:SOUND3,10,3:VOL4
79 IFM1=37THENCHAR1,M1,M2,"(SPACE)":M1=23:M2=1:A
=0
80 IFPEEK(3072+M1+40*M2)=83THENGOSUB97
81 CHAR1,M1,M2,"(BRN S2)"
82 CHAR1,31,7,"C"
83 GOTO69
84 CHAR1,X,Y,"(SPACE)"
85 X=X-1:IFPEEK(3072+X+40*Y)=102THENX=X+1:Y=Y-2:
X=X-1
86 IFX=18THEN88
87 RETURN
88 FORT=1TO8
89 CHAR1,X,Y,"(SPACE)":X=X-2
90 CHAR1,X,Y,"(BLUE SS)"
91 CHAR1,X,Y,"(SPACE)":Y=Y+2
92 CHAR1,X,Y,"(BLUE SS)":VOL8:SOUND3,1,5:VOL4
93 FORF=1TO25:NEXTF
94 NEXTT:FORB=80TO600STEP-50
95 SOUND1,5,2:NEXT
96 REM VERLOREN
97 SCNCLR
98 FORT=1TO100:GETA$:A$="":NEXT
99 FORV=8TDOSTEP-1
100 VOLV:SOUND3,930,10
101 NEXTV
102 INPUT"WOLLEN(SPACE)SIE(SPACE)NOCH(SPACE)EINM
AL(SPACE)(J/N)":A$:IFA$="J"THENRUN:ELSEEND
103 REM PHASE 2
104 FORT=1TO10
105 VOLB
106 SOUND3,800,5
107 SOUND3,700,5:VOL4
108 CHAR1,35,3,"(SPACE)2"
109 NEXT
110 CHAR1,31,7,"(GREEN)THE(SPACE)CAVE"
111 Y=Y+1
112 CHAR1,19,0,"(BRN C+3)"
113 CHAR1,19,1,"(C+3)"
114 J=JOY(1)
115 BETA$:IFA$="(SPACE)"THENGETKEYB$

```


114 IFJ=1THENGOSUB142	<183>	200 CHAR1,W1,W2,"(GREEN)@"	<188>
117 IFJ=3THENGOSUB146	<228>	201 GOTO180	<28>
118 IFJ=5THENGOSUB150	<191>	202 REM MAENCHEN BEWEGEN	<173>
119 IFJ=7THENGOSUB154	<236>	203 CHAR1,X,Y,"(SPACE)"	<155>
120 TS=TS-1:IFTS=0THEN97	<13>	204 Y=Y-1	<53>
121 IFPEEK(3072+X+40*Y)=66THEN96	<182>	205 IFPEEK(3072+X+40*Y)=102THENY=Y+1	<21>
122 IFPEEK(3072+X+40*Y)=65THENGOSUB158	<98>	206 IFPEEK(3072+X+40*Y)=42THENY=Y+1	<209>
123 IFPEEK(3072+X+40*Y)=88THENGOSUB158	<187>	207 RETURN	<93>
124 CHAR1,X,Y,"(BLUE SS)"	<163>	208 CHAR1,X,Y,"(SPACE)"	<160>
125 IFX=10ANDY=15THEN166	<145>	209 X=X+1	<50>
126 REM SPINNEN BEWEGEN	<100>	210 IFPEEK(3072+X+40*Y)=102THENX=X-1:P=P-1	<219>
127 IFD=0THENCHAR1,S1,S2,"(SPACE)":S2=S2-1	<183>	211 IFPEEK(3072+X+40*Y)=42THENX=X-1:P=P-1	<48>
128 IFS2=1THEND=1	<96>	212 RETURN	<98>
129 IFD=1THENCHAR1,S1,S2,"(ORNG SB)":S2=S2+1	<138>	213 CHAR1,X,Y,"(SPACE)"	<165>
130 IFS2=14THEND=0	<105>	214 Y=Y+1	<59>
131 IFPEEK(3072+S1+40*S2)=83THENGOSUB97	<159>	215 IFPEEK(3072+X+40*Y)=102THENY=Y-1	<55>
132 IFU=0THENCHAR1,G1,G2,"(SPACE)":G2=G2-1	<224>	216 IFPEEK(3072+X+40*Y)=42THENY=Y-1	<242>
133 IFG2=6THENU=1	<144>	217 RETURN	<103>
134 IFU=1THENCHAR1,G1,G2,"(D.BLUE SB)":G2=G2+1	<43>	218 CHAR1,X,Y,"(SPACE)"	<170>
135 IFG2=12THENU=0	<122>	219 X=X-1	<64>
136 IFPEEK(3072+G1+40*G2)=83THENGOSUB97	<164>	220 IFPEEK(3072+X+40*Y)=102THENX=X+1	<248>
137 IFPEEK(3072+G1+40*G2)=83THENGOSUB97	<133>	221 IFPEEK(3072+X+40*Y)=42THENX=X+1	<182>
138 CHAR1,S1,S2,"(GREEN SV)"	<90>	222 RETURN	<108>
139 CHAR1,G1,G2,"(RED SV)"	<193>	223 REM GRABWACHE BEWEGEN	<77>
140 GOTO114	<218>	224 CHAR1,W1,W2,"(SPACE)"	<200>
141 REM MAENCHEN BEWEGEN	<112>	225 W1=W1+1	<84>
142 CHAR1,X,Y,"(SPACE)"	<94>	226 IFPEEK(3072+W1+40*W2)=42THENW1=W1-1	<8>
143 Y=Y-1	<248>	227 IFPEEK(3072+W1+40*W2)=102THENW1=W1-1	<7>
144 IFPEEK(3072+X+40*Y)=102THENY=Y+1	<216>	228 RETURN	<114>
145 RETURN	<31>	229 CHAR1,W1,W2,"(SPACE)"	<205>
146 CHAR1,X,Y,"(SPACE)"	<98>	230 W2=W2+1	<96>
147 X=X+1	<244>	231 IFPEEK(3072+W1+40*W2)=42THENW2=W2-1	<62>
148 IFPEEK(3072+X+40*Y)=102THENX=X-1	<200>	232 IFPEEK(3072+W1+40*W2)=102THENW2=W2-1	<63>
149 RETURN	<35>	233 RETURN	<119>
150 CHAR1,X,Y,"(SPACE)"	<102>	234 CHAR1,W1,W2,"(SPACE)"	<210>
151 Y=Y+1	<252>	235 W1=W1-1	<100>
152 IFPEEK(3072+X+40*Y)=102THENY=Y-1	<248>	236 IFPEEK(3072+W1+40*W2)=42THENW2=W2-1	<67>
153 RETURN	<39>	237 IFPEEK(3072+W1+40*W2)=102THENW1=W1+1	<245>
154 CHAR1,X,Y,"(SPACE)"	<106>	238 RETURN	<124>
155 X=X-1	<0>	239 CHAR1,W1,W2,"(SPACE)"	<215>
156 IFPEEK(3072+X+40*Y)=102THENX=X+1	<184>	240 W2=W2-1	<112>
157 RETURN	<43>	241 IFPEEK(3072+W1+40*W2)=42THENW2=W2+1	<45>
158 CHAR1,X,Y,"(SPACE)"	<110>	242 IFPEEK(3072+W1+40*W2)=102THENW2=W2+1	<45>
159 REM GEGENSTAENDE NEHMEN	<79>	243 RETURN	<129>
160 VOLB: SOUND3,800,10:VOL4	<203>	244 REM SPIEL BESCHAFFT	<114>
161 P=P+1:T1=T1+1	<44>	245 FORR=1TO42	<190>
162 IFT1=6THENCHAR1,10,15,"(SPACE)"	<1>	246 READS	<35>
163 P#STR\$(P)	<40>	247 VOLB: SOUND3,5,5	<225>
164 CHAR1,35,5,P#	<82>	248 NEXT:RESTORE:VOL4	<249>
165 RETURN	<51>	249 DATA100,100,500,500,600,600,500	<246>
166 REM PHASE 3	<83>	250 DATA400,400,300,300,200,200,100	<215>
167 VOLB	<242>	251 DATA500,500,400,400,300,300,200	<58>
168 SOUND3,800,10	<200>	252 DATA500,500,400,400,300,300,200	<59>
169 FORT=1TO125:NEXT	<42>	253 DATA100,100,500,500,600,600,500	<250>
170 SOUND3,900,5	<62>	254 DATA400,400,300,300,200,200,100	<219>
171 FORT=1TO75:NEXT	<1>	255 SCNCLR	<231>
172 SOUND3,900,5	<64>	256 TM=TM-50:ZF=ZF-1	<46>
173 FORT=1TO75:NEXT	<3>	257 PRINT:PRINT:PRINT	<191>
174 FORT=8TO1STEP-1	<215>	258 PRINT"(SPACE15 RED)PYRAMID	<51>
175 VOLT: SOUND3,900,5:NEXT:VOL4	<160>	259 PRINT	<157>
176 Y=Y+1	<21>	260 PRINT"(SPACE15 D.BLUE)(WHITE SA D.BLUE)(SPACE)	<232>
177 CHAR1,10,15,"(BRN C#)"	<251>	BY.	<81>
178 CHAR1,31,7,"(RED)TREASURY"	<86>	261 PRINT"(SPACE13 L.GRN)EAGLE-SOFT	<196>
179 CHAR1,35,3,"(SPACE)3"	<86>	262 PRINT:PRINT:PRINT	<102>
180 J=JOY(1)	<90>	263 FORT=1TO100:BETA#:A#=""	<143>
181 BETA#:IFA#=""(SPACE)"THENGETKEYB#	<154>	264 INPUT"(PURPLE)NAME":N#	<226>
182 IFJ=1THENGOSUB203	<233>	265 PRINT:PRINT	<40>
183 IFJ=3THENGOSUB208:P=P+1	<102>	266 IFF:PUTHEN268	
184 IFJ=5THENGOSUB213	<3>	267 PRINTN#"(SPACE BLUE)SIE(SPACE)GIND(SPACE)MIT	
185 IFJ=7THENGOSUB218:P=P-1	<142>	(SPACE)"P"(SPACE)PUNKTEN(SPACE)DER(SPACE)NEUE(SPACE)	
186 IFJ=131ANDPEEK(3072+X+1+40*Y)=42THENCHAR1,X+1,Y,"(SPACE)"	<137>	ACE)PHARAD(SPACE)VON(SPACE)PYRAMID":PU=P	<251>
187 TS=TS-1:IFTS=0THEN97	<80>	268 PRINT:PRINTN#"(SPACE)SIE(SPACE)HABEN(SPACE)L	
188 P#STR\$(P):CHAR1,35,5,P#	<239>	EVEL(SPACE)"LE"(SPACE)BESCHAFT"	<196>
189 BW=BW+1:IFBW=2THENBW=0:ELSEGOTO197	<5>	269 PRINT	<167>
190 REM RICHTUNG BESTIMMEN	<143>	270 LE=LE+1	<215>
191 VOLB: SOUND3,500,2:VOL4	<87>	271 PRINT"WOLLEN(SPACE)SIE(SPACE)LEVEL(SPACE)":L	
192 IFX>W1THENGOSUB224	<126>	E: (SPACE)SPIELEN": INPUTS#	<216>
193 IFY>W2THENGOSUB229	<184>	272 IFLEFT\$(S#,1)="J"THEN37	<37>
194 IFX<W1THENGOSUB234	<143>	273 IFLEFT\$(S#,1)="N"THENP=0:GOTO15:ELSEGOTO274	<22>
195 IFY<W2THENGOSUB239	<201>	274 PRINT:PRINT:PRINT	<208>
196 IFPEEK(3072+W1+1+40*W2)=102THENGOSUB229:GOTO		275 PRINT"(FLASHON)WIE(SPACE)BITTE(SPACE)?(FLASH	
196		OFF)"	<109>
197 IFW1=XANDW2=YTHENGOSUB97	<37>	276 GOTO271	<107>
198 IFPEEK(3072+X+40*Y)=35THEN245	<136>	277 POKE 65299,208:HELP	<13>
199 CHAR1,X,Y,"(BLUE SS)"	<150>	278 REM ENDE	<131>
	<238>	ENDE DES LISTINGS	

TURBO DISC

C64

Turbo-Load ist ein Floppyspeicher für die 1541. Er liegt im Speicherbereich von CDOO - DOOO und kann somit nur Programme mit einer Länge von 197 Blocks einladen. Die 1541 wird bis zu 6,4 mal schneller als normal, dadurch entfallen lästige Ladezeiten. Aktiviert wird der

Turbo-Load mit SYS 52480 und abgeschaltet wird er durch Reset oder Run-Stop/Restore. Bevor man ihn startet muß man "NEW" eingeben. Der Basic-Loader speichert ein 4-Block langes Programm ab, das man dann absolut (,8,1) lädt und startet wie oben beschrieben.

Bis zu 7 x schnellere Floppy-Operationen

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

10 REM *****
11 REM *****
12 REM ***
13 REM *** TURBO-LOAD VERSION 3.1 ***
14 REM ***
15 REM *** COPYRIGHT 03.05.85 ***
16 REM ***
17 REM *** BY ***
18 REM ***
19 REM *** LARS HENNIG ***
20 REM ***
21 REM *** AND ***
22 REM ***
23 REM *** SOENKE HENNIG ***
24 REM ***
25 REM *** FOR ***
26 REM ***
27 REM *** THE T R O N I C ***
28 REM ***
29 REM *** V E R L A B ***
30 REM ***
31 REM *****
32 REM *****
33 :
34 REM *** PROGRAM ***
35 :
36 B=53281 : REM BACKGROUND
37 F=53280 : REM FOREGROUND
38 PRINTCHR$(147)
39 POKEB,0:POKEF,0
40 PRINT" (YELLOW)" : REM GELB
41 PRINTCHR$(17)+CHR$(17)+CHR$(17)+CHR$(17)
42 PRINTCHR$(17)+CHR$(17)+CHR$(17)
43 PRINTCHR$(17)+CHR$(17)+CHR$(17)+CHR$(14)
44 PRINT" (SPACE14)BITTE (SPACE)WARTEN"
45 PRINTCHR$(142)
46 GOSUB 10 1: REM MASCHINENROUTINEN
47 PRINTCHR$(147)"(SPACE2 SC CR SC SPACE CR SPAC
E CR SPACE SU SC SI SPACE CA SC SI SPACE SU SC S
I SPACE2 CR SPACE3 SU SC SI SPACE SU SC SI SPACE
CA SC SI SPACE2)";
48 PRINT"(SPACE3 SB SPACE2 SB SPACE SB SPACE SB
SPACE SB SPACE SB SPACE SB SPACE SB SPACE SB SPA
CE2 SB SPACE3 SB SPACE SB SPACE SB SPACE SB SPAC
E SB SPACE SB SPACE2)";
49 PRINT"(SPACE3 SB SPACE2 SB SPACE SB SPACE CQ
<207> SC SK SPACE CQ SC CM SPACE SB SPACE SB CD CF SB
<208> SPACE3 SB SPACE SB SPACE CQ SC CM SPACE SB SPACE
<9> SB SPACE2)";
<19> 50 PRINT"(SPACE3 SB SPACE2 SB SPACE SB SPACE SB
<11> SH SPACE2 SB SPACE SB SPACE SB SPACE SB SPACE2 S
<130> B SPACE3 SB SPACE SB SPACE SB SPACE SB SPACE SB
<13> SPACE SB SPACE2)";
<106> 51 PRINT"(SPACE3 CE SPACE2 SJ SC SK SPACE CE SPA
<15> CE CE SPACE CZ SC SK SPACE SJ SC SK SPACE2 SJ SC
<167> 2 SPACE SJ SC SK SPACE CE SPACE CE SPACE CZ SC S
<17> K SPACE2)";
<129> 52 PRINTCHR$(150)"(DOWN3 RIGHT11)";
<19> 53 PRINTCHR$(67)+CHR$(79)+CHR$(80)+CHR$(89)+CHR$(
<98> (82)+CHR$(73);
<21> 54 PRINTCHR$(71)+CHR$(72)+CHR$(84)+CHR$(32)+CHR$(
<6> (49)+CHR$(57);
<23> 55 PRINTCHR$(56)+CHR$(53)+CHR$(32)+CHR$(66)+CHR$(
<130> (89)+CHR$(13);
<25> 56 PRINT" (RIGHT14)";
<7> 57 PRINTCHR$(76)+CHR$(65)+CHR$(82)+CHR$(83)+CHR$(
<27> (32)+CHR$(72);
<228> 58 PRINTCHR$(69)+CHR$(78)+CHR$(78)+CHR$(73)+CHR$(
<229> (71)+CHR$(13);
<91> 59 PRINT" (RIGHT12)";
<41> 60 PRINTCHR$(38)+CHR$(32)+CHR$(83)+CHR$(79)+CHR$(
<93> (69)+CHR$(78);
<204> 61 PRINTCHR$(75)+CHR$(69)+CHR$(32)+CHR$(72)+CHR$(
<242> (69)+CHR$(78);
<246> 62 PRINTCHR$(78)+CHR$(73)+CHR$(71)+CHR$(13);
<44> 63 PRINTCHR$(31)" (SPACE7)BITTE (SPACE)EINE (SPACE)
<60> TASTE (SPACE)DRUECKEN";
<244> 64 POKE19B,0:WAIT19B,1:POKE19B,1
<132> 65 :
<177> 66 :
<126> 67 REM **** ANLEITUNG ****
<223> 68 :
<10> 69 :
70 PRINTCHR$(147)+CHR$(30)
71 PRINT" (DOWN2 SPACE3)ANLEITUNG:"
72 PRINT" (SPACE3 CT10)";
<3> 73 PRINT" (DOWN SPACE)TURBO-LOAD (SPACE)MACHT (SPAC
E)DIE (SPACE)FLOPPY (SPACE)CA. (SPACE)6.4 (SPACE)";
<151> 74 PRINT" (DOWN SPACE)MAL (SPACE)SCHNELLER. (SPACE)
<103> NACH (SPACE)RUN-STOP (SPACE)UND (SPACE)RE-";
75 PRINT" (DOWN SPACE)STORE (SPACE)TASTE (SPACE)KAN
<47> N (SPACE)TURBO-LOAD (SPACE)MIT (SPACE)BYS (SPACE)";

```



```

76 PRINT"(DOWN SPACE)52480(SPACE)WIEDER(SPACE)AK
TIVIERT(SPACE)WERDEN.(SPACE)DIE(SPACE)" <227>
77 PRINT"(DOWN SPACE)PROGRAMME(SPACE)DURFEN(SPA
CE)BIS(SPACE)ZU(SPACE)RVSDN)197(RVSOFF SPACE)BLD
ECKE" <72>
78 PRINT"(DOWN SPACE)BETRAGEN.(SPACE)WEIL(SPACE)
SIE(SPACE)SONGT(SPACE)TURBO-LOAD(SPACE)" <162>
79 PRINT"(DOWN SPACE)UEBERSCHREIBEN(SPACE)WUERDE
N(SPACE)!" <53>
80 PRINTCHR$(129)"(DOWN3 SPACE)BITTE(SPACE)DISKE
TTE(SPACE)EINLEGEN(SPACE)UND(SPACE)RVSDN)RETURN(
RVSOFF)" <110>
81 PRINT"(SPACE)DRUECKEN." <194>
82 GETA$:IFA$(CHR$(13))THENB2 <13>
83 GOSUBB9 <251>
84 : <142>
85 : <143>
86 REM **** ABSPEICHERN **** <38>
87 : <145>
88 : <146>
89 POKEB69,84:POKEB70,85:POKEB71,82:POKEB72,66:P
DKEB73,79:POKEB74,45:POKEB75,76 <113>
90 POKEB76,79:POKEB77,65:POKEB78,68:POKEB79,32:P
DKEB80,86:POKEB81,51:POKEB82,46 <12>
91 POKEB83,49 <121>
92 PRINTCHR$(147)+CHR$(156) <158>
93 PRINT"(DOWN SPACE)DER(SPACE)NUN(SPACE)ABGESPE
ICHERTE(SPACE)TURBO-LOAD(SPACE)" <106>
94 PRINT"(DOWN SPACE)MUSS(SPACE)MIT(SPACE)'8,1'
(SPACE)GELADEN(SPACE)UND(SPACE)MIT(SPACE)" <171>
95 PRINT"(DOWN SPACE)RVSDN)BY(SPACE)52480(RVSOFF
SPACE)AKTIVIERT(SPACE)WERDEN." <202>
96 PRINT"(DOWN SPACE)VOR(SPACE)DEM(SPACE)STARTEN
(SPACE)RVSDN)'NEW'(RVSOFF SPACE)EINGEBEN":SYSB32
97 PRINTCHR$(158)"(DOWN5 SPACE2)VIEL(SPACE)SPASS
(SPACE)BEIM(SPACE)SCHNELLEREN(SPACE)LADEN(SPACE)
!" <57>
98 PRINTCHR$(159)"(DOWN3 SPACE2)BY(SPACE)LARS(SP
ACE)&(SPACE)GÖENKE <112>
99 FORX=0TO6000:NEXT <60>
100 POKEB,246:POKEF,254:PRINTCHR$(147)+CHR$(154)
:END <249>
101 FORI=52480TO53247:READA:POKEI,A <247>
102 P=P+A <247>
103 NEXT <233>
104 : <162>
105 : <163>
106 REM UEBERPRUEFEN, OB EIN <12>
107 REM FEHLER IN DEN DATA'S <133>
108 REM AUFGETRETEN IST ODER <84>
109 REM NICHT! <198>
110 : <168>
111 : <169>
112 IF<>B9904THENPRINT"DATA-FEHLER(SPACE)!" <85>
113 IFP=B9904THENGOSUB199:RETURN <128>
114 END <242>
115 : <173>
116 : <174>
117 REM **** TURBO-LOAD DATA'S **** <50>
118 : <176>
119 : <177>
120 DATA169,0,141,0,208,169,75,141,1 <107>
121 DATA208,169,56,141,2,208,169,208,141,3,208 <113>
122 DATA169,131,141,2,3,169,164,141,3 <2>
123 DATA3,32,213,207,96,0,0,0,0,0,0,0,0,0,169 <131>
124 DATA3,133,49,32,10,245,80,254,184 <8>
125 DATA173,1,28,153,0,3,200,208,244,160,186 <32>
126 DATA80,254,184,173,1,28,153,0,1 <30>
127 DATA200,208,244,32,224,248,165,56,197,71,240 <113>
128 DATA5,169,4,76,105,249,32,233,245 <28>
129 DATA197,58,240,5,169,5,76,105,249,173,0,3 <231>
130 DATA240,42,162,0,189,0,3,32,134,6 <157>
131 DATA232,208,247,173,0,3,197,12,208,12,173 <123>
132 DATA1,3,133,13,173,0,3,133,12,208 <65>
133 DATA166,133,12,173,1,3,133,13,169,1,76,105 <132>
134 DATA249,162,0,238,1,3,189,0,3,32 <212>
135 DATA134,6,232,236,1,3,208,244,169,127,76,105 <219>
136 DATA249,0,0,0,0,0,0,133,133,44,0 <65>
137 DATA24,16,251,169,16,141,0,24,44,0,24,48,251 <181>
138 DATA169,0,6,133,42,10,6,133,42,10 <197>
139 DATA141,0,24,169,0,6,133,42,10,6,133,42,10 <120>
140 DATA141,0,24,169,0,6,133,42,10,6 <129>
141 DATA133,42,10,141,0,24,169,0,6,133,42,10,6 <204>
142 DATA133,42,10,141,0,24,234,234,234 <169>
143 DATA169,15,141,0,24,96,0,32,0,193,165,24 <189>
144 DATA166,25,133,12,134,13,141,0,3 <232>
145 DATA142,1,3,169,224,133,3,165,3,48,252,201 <126>
146 DATA2,144,244,201,127,240,3,76,10 <168>
147 DATA230,76,158,193,0,0,0,0,169,11,141,0,221 <189>
148 DATA173,0,221,16,251,169,3,141,0 <196>
149 DATA221,32,114,206,234,234,234,173,0,221,10 <163>
150 DATAB,10,38,164,40,38,164,173,0 <24>
151 DATA221,10,8,10,38,164,40,38,164,173,0,221 <66>
152 DATA10,8,10,38,164,40,38,164,173 <96>
153 DATA0,221,10,8,10,38,164,40,38,164,165,164 <24>
154 DATA73,255,96,234,234,234,234,234 <27>
155 DATA234,96,32,44,206,133,174,32,44,206,133 <236>
156 DATA175,165,2,208,8,165,195,133 <22>
157 DATA174,165,196,133,175,96,0,0,0,0,0,0,0,0 <54>
158 DATA133,147,160,0,177,187,201,36 <41>
159 DATA208,3,76,167,244,169,8,133,184,32,175 <151>
160 DATA245,32,231,255,166,185,134,2 <29>
161 DATA169,96,133,185,32,192,255,165,186,32,9 <112>
162 DATA237,165,185,32,199,237,32,19 <47>
163 DATA238,165,186,32,195,255,165,144,74,74,144 <65>
164 DATA3,76,4,247,32,210,245,169,44 <128>
165 DATA133,3,169,205,133,4,169,0,133,5,169,6 <53>
166 DATA133,6,165,186,32,12,237,169 <82>
167 DATA111,32,185,237,169,77,32,221,237,169,45 <91>
168 DATA32,221,237,169,87,32,221,237 <165>
169 DATA165,5,32,221,237,165,6,32,221,237,169 <206>
170 DATA32,32,221,237,160,0,177,3,32 <46>
171 DATA221,237,200,192,32,144,246,32,254,237 <104>
172 DATA24,165,3,105,32,133,3,144,2 <36>
173 DATA230,4,24,165,5,105,32,133,5,144,2,230,6 <13>
174 DATA166,6,224,7,144,173,234,234 <166>
175 DATA234,234,165,186,32,12,237,169,111,32,185 <201>
176 DATA237,169,77,32,221,237,169,45 <64>
177 DATA32,221,237,169,69,32,221,237,169,213,32 <238>
178 DATA221,237,169,6,32,221,237,32 <75>
179 DATA254,237,173,17,208,41,239,141,17,208,120 <167>
180 DATA162,4,32,44,206,240,33,32,44 <55>
181 DATA206,224,2,240,3,32,121,206,160,0,32,44 <204>
182 DATA206,145,174,230,174,208,2,230 <49>
183 DATA175,232,208,242,162,2,208,221,234,234 <80>
184 DATA234,32,44,206,224,2,240,5,72 <62>
185 DATA32,121,206,104,170,202,202,160,0,32,44 <182>
186 DATA206,145,174,230,174,208,2,230 <53>
187 DATA175,202,208,242,165,186,32,12,237,169 <93>
188 DATA111,32,185,237,169,73,32,221 <79>
189 DATA237,32,254,237,173,17,208,9,16,141,17 <108>
190 DATA208,166,174,164,175,24,96,169 <252>
191 DATA152,141,48,3,169,206,141,49,3,96,234 <214>
192 DATA234,234,234,234,234,234,234 <138>
193 DATA234,234,234,234,234,234,234,234,234,234 <86>
194 DATA234,234,234,234,234,234,234 <140>
195 DATA234,234,234,234,234,234,234 <141>
196 : <254>
197 REM ENDE DES TURBO-LOADER'S <250>
198 : <0>
199 FORZ=832TO868:READY:POKEZ,Y:NEXT:RETURN <5>
200 : <2>
201 : <3>
202 REM **** SAVER DATA'S **** <22>
203 : <5>
204 : <6>
205 DATA169,001,162 <44>
206 DATA008,169,255 <168>
207 DATA032,186,255 <144>
208 DATA169,015,162 <86>
209 DATA101,160,003 <230>
210 DATA032,189,255 <171>
211 DATA169,000,133 <21>
212 DATA251,169,205 <110>
213 DATA133,252,162 <74>
214 DATA000,160,208 <53>
215 DATA234,169,251 <126>
216 DATA032,216,255 <110>
217 DATA096 <63>
218 : <20>
219 REM ENDE DES SAVER'S <230>
220 : <22>
ENDE DES LISTINGS

```

...echt Turbo!

Hot-line:

Bei Fragen zu den Listings steht Ihnen unsere Programmierabteilung für telefonische Anfragen zur Verfügung. Bitte wählen Sie nur diese Nummer und rufen Sie nur zu den angegebenen Zeiten an: Montag-Freitag, 14-16 Uhr!

0 56 51 / 3 00 13

C-64

IRQ-Routine

Wer schon einmal eine Dateiverwaltung o. ä. selbst geschrieben hat, wird sich sicherlich schon über das Scrollen des Bildschirms nach oben geärgert haben, denn die erste Zeile mit dem (überaus wichtigen) Copyright-Vermerk und dem Namen des Autors geht bei fast jeder Eingabe flöten. Solche Probleme treten mit diesem kleinen Maschinenprogramm (ca. 40 Bytes) nicht auf. Es ist vollständig in den IRQ (Interrupt) eingebaut,

der jede 60tel Sekunde aufgerufen wird. Der Text, der in die oberste(n) Zeile(n) aufgerufen werden soll, ist variabel und liegt im Bereich ab 49192 aufwärts. Die Anzahl der auszugebenden Zeichen vom Text müssen Sie in Speicherstelle 252 POKEn. Die Farbe des Textes können Sie auch verändern: Sie brauchen den Farbcode (0 = schwarz, 1 = weiß, 2 = rot, usw.) des Textes nur in Speicherstelle

251 zu schreiben. Noch ein Tip am Schluß: Damit das Eingeben nicht mühselig ist, können Sie sich folgenden Trick bedienen:

- (1) - Bildschirm löschen
- (2) - Den gewünschten Text in

- die obersten Bildschirmzeilen schreiben.
- (3) - Eingeben: FORI= 0TO255: POKE 49192+1,PEEK (1024+I): NEXT (Return)
- (4) - Einen Moment warten und schon kann's losgehen.

C-64 LISTING + CHECKSUMMEN (C V1.0)

```
100 CLR:FORI=49152TO49191:READX:POKEI,X:PS=PS+X:
NEXT
110 IFFB<>5026THENPRINT"FEHLER (SPACE) IN (SPACE) DA
TAB (SPACE) !":END
120 PRINT"OK (SPACE) - (SPACE) START (SPACE) MIT (SPACE
) 'SYS (SPACE) 49152'":END
130 DATA120,169,13,141,20,3,169,192,141,21,3,88,
96,165,252,201,0,240,18,160
140 DATA0,185,40,192,153,0,4,165,251,153,0,216,2
00,196,252,208,240,76,49,234
ENDE DES LISTINGS
```

RYLAN

C-64

Hallo!

Dies ist also das erste Programm, das nur mit Ex-UMH-System läuft. 'RYLAN' benutzt außer dem UMH-SYS noch eine abso-

lut winzige M/C-Routine, die das Scrolling des Hintergrundes bewirkt. Jegliche Spriteverwaltung wird mit dem UMH-SYS

getätigt. In 'RYLAN' geht es darum, eine zehnteilige Strecke mit den zur Verfügung stehenden Leben zu durchfliegen. Die zehn Levels haben natürlich steigenden Schwierigkeitsgrad. 'RYLAN's Besonderheit ist, daß ein zweiter Spieler die Flugbahn eines der störenden 'Satelliten' (ahem) beeinflussen kann.

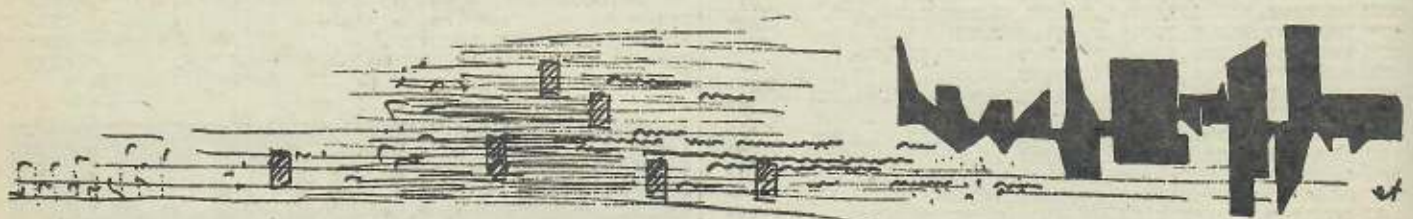
- Der Satellit wird mit einem Joystick in Port 2 gelenkt

Das Ausmaß des Einflusses der Lenkbewegungen auf die Figuren kann in einem Menü individuell eingestellt werden.

Ps: Ex-UMH-System erschien in "Compute mit" 12/86!

Die Steuerung erfolgt wie folgt:

- Der Gleiter wird mit einem Joystick in Port 1 gelenkt



Teil 1

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 REM * UNTER 'LADER' SPEICHERN (FLOPPY) <56>
1 A#CHR$(34) <106>
4 PRINT"(WHITE CLEAR)LOADING...(BLACK DOWN2)":PO
KE 532B1,0:POKE 532B0,0 <12B>
5 PRINT"LOAD"A#"PRG2"A#",B" <80>
6 PRINT"(DOWN4)RUN" <13>
7 PRINT"(DOWN2)LOAD"A#"PRG3"A#",B" <255>
8 PRINT"(DOWN4)RUN" <15>
9 PRINT"(UP17)":POKE631,13:POKE632,13:POKE633,13
:POKE634,13 <7>
10 PDKE 19B,4 <162>
    ENDE DES LISTINGS
    
```

Teil 2

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 REM VEKTOREN AENDERN : BASIC-START <236>
1 POKE163B5,0:POKE 163B6,0:POKE43,2:POKE44,64:NE
W <211>
    ENDE DES LISTINGS
    
```

Teil 3

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 REM ***** 'RYLAN' ***** <115>
1 REM * POSTAL SERVICES * <93>
2 REM * WRITTEN BY : ULRICH MUEHL * <58>
5 REM * (C) 1986 PARAVISION COMPANY * <179>
8 REM **EX - UMH - SYSTEM REQUIRED !** <41>
9 REM ***** <12>
20 PRINT"(GREEN CLEAR RIGHT5)AUF(SPACE)WUNSCH(SP
ACE)DER(SPACE)TRONIC-VERLAGES":POKE532B0,0:POKE5
32B1,0 <60>
30 PRINT"(CYAN RIGHT11)PRAESENTIERT(SPACE)VON(SP
ACE)!" <39>
40 PRINT"(DOWN RIGHT3 YELLOW)P(SPACE)A(SPACE)R(S
PACE)A(SPACE)V(SPACE)I(SPACE)B(SPACE)I(SPACE)O(S
PACE)N(SPACE)C(SPACE)O(SPACE)M(SPACE)P(SPACE)A(
SPACE)N(SPACE)Y" <166>
50 PRINT"(DOHNS RIGHT13 RED)R(SPACE2)Y(SPACE2)L(
SPACE2)A(SPACE2)N" <144>
100 PRINT"(DOHNS RIGHT13 YELLOW)READING(SPACE)DA
TA!" <65>
110 FORX=B192TO10559:READD:POKEX,DT:NEXTX <195>
12000 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <49>
12001 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <50>
12002 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <51>
12003 DATA 0,4B,0,0,4B,0,0,0,0,0,0,0,0,0,0,0 <28>
12004 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <53>
12005 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <54>
12006 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <59>
12008 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <57>
12009 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 <58>
12010 DATA 0,0,0,0,0,0,0,0,16,0 <61>
12011 DATA 0, 112, 0, 0, 56, 0, 0, 32, 0 <80>
12012 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <61>
12013 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <62>
12014 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <67>
12016 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <65>
12017 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <66>
12018 DATA 0, 0, 0, 0, 16, 0, 0, 16, 0 <72>
12019 DATA 0, 240, 0, 0, 60, 0, 0, 32, 0 <29>
12020 DATA 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <207>
12021 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <70>
12022 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <75>
12024 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <73>
12025 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <74>
12026 DATA 0, 16, 0, 0, 16, 0, 0, 16, 0 <84>
12027 DATA 1, 240, 0, 0, 62, 0, 0, 32, 0 <65>
12028 DATA 0, 32, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <192>
    
```

```

12029 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <78>
12030 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <83>
12032 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <82>
12033 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 16, 0, 0, 0, 0, 0 <85>
12034 DATA 0, 16, 0, 0, 16, 0, 0, 16, 0, 0, 16, 0, 0, 0, 0, 0 <93>
12035 DATA 3, 240, 0, 0, 63, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0 <91>
12036 DATA 0, 32, 0, 0, 32, 0, 0, 32, 0, 0, 32, 0, 0, 0, 0, 0 <17>
12037 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <87>
12038 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <92>
12040 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <90>
12041 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 32, 0, 0, 0, 0, 0, 0 <57>
12042 DATA 0, 32, 0, 0, 16, 0, 0, 16, 0, 0, 16, 0, 0, 0, 0, 0 <89>
12043 DATA 0, 243, 0, 3, 60, 0, 0, 32, 0, 0, 0, 0, 0, 0, 0, 0 <102>
12044 DATA 0, 32, 0, 0, 16, 0, 0, 16, 0, 0, 16, 0, 0, 0, 0, 0 <91>
12045 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <95>
12046 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <100>
12048 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <98>
12049 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <99>
12050 DATA 0, 64, 0, 0, 32, 0, 0, 18, 0, 0, 0, 0, 0, 0, 0, 0 <131>
12051 DATA 0, 116, 0, 0, 184, 0, 1, 32, 0, 0, 0, 0, 0, 0, 0, 0 <10>
12052 DATA 0, 16, 0, 0, 8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <84>
12053 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <103>
12054 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <108>
12056 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <106>
12057 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <107>
12058 DATA 0, 0, 0, 0, 0, 0, 0, 0, 120, 0, 0, 0, 0, 0, 0, 0 <230>
12059 DATA 0, 120, 0, 0, 120, 0, 0, 120, 0, 0, 120, 0, 0, 0, 0, 0 <175>
12060 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <110>
12061 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <111>
12062 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <116>
12064 DATA 0, 0, 0, 1, 255, 128, 6, 10, 224, 0, 0, 0, 0, 0, 0, 0 <147>
12065 DATA 8, 13, 80, 31, 6, 168, 16, 195, 88, 0, 0, 0, 0, 0, 0 <44>
12066 DATA 63, 1, 252, 32, 252, 4, 33, 47, 194, 0, 0, 0, 0, 0, 0 <146>
12067 DATA 114, 158, 63, 68, 127, 193, 68, 127, 193, 0, 0, 0, 0, 0, 0, 0 <29>
12068 DATA 114, 158, 63, 33, 47, 196, 32, 252, 4, 0, 0, 0, 0, 0, 0, 0 <194>
12069 DATA 63, 1, 252, 16, 195, 88, 31, 6, 168, 0, 0, 0, 0, 0, 0, 0 <50>
12070 DATA 8, 13, 80, 6, 10, 224, 1, 255, 128, 0, 0, 0, 0, 0, 0, 0 <140>
12072 DATA 0, 0, 0, 0, 0, 0, 0, 0, 48, 0, 0, 0, 0, 0, 0, 0 <206>
12073 DATA 0, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <39>
12074 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <124>
12075 DATA 24, 48, 96, 24, 48, 96, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <246>
12076 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <126>
12077 DATA 0, 0, 0, 0, 48, 0, 0, 48, 0, 0, 48, 0, 0, 0, 0, 0 <15>
12078 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <132>
12080 DATA 0, 0, 0, 0, 0, 0, 0, 0, 76, 0, 0, 0, 0, 0, 0, 0 <4>
12081 DATA 0, 76, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <57>
12082 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 96, 0, 0, 0, 0, 0 <40>
12083 DATA 0, 48, 96, 24, 48, 0, 24, 0, 0, 0, 0, 0, 0, 0, 0 <208>
12084 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <134>
12085 DATA 0, 0, 0, 0, 24, 0, 0, 24, 0, 0, 0, 0, 0, 0, 0, 0 <109>
12086 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <140>
12088 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <138>
12089 DATA 0, 192, 0, 0, 192, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <109>
12090 DATA 0, 0, 0, 0, 0, 192, 0, 0, 192, 0, 0, 192, 0, 0, 0, 0 <110>
12091 DATA 0, 48, 0, 0, 48, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0, 0 <143>
12092 DATA 12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <6>
12093 DATA 0, 12, 0, 0, 12, 0, 0, 12, 0, 0, 0, 0, 0, 0, 0, 0 <228>
12094 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <148>
12096 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <146>
12097 DATA 0, 0, 0, 1, 128, 0, 1, 128, 0, 0, 0, 0, 0, 0, 0, 0 <59>
12098 DATA 0, 1, 128, 0, 1, 128, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <188>
12099 DATA 0, 48, 0, 0, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <125>
12100 DATA 6, 0, 0, 6, 0, 0, 0, 0, 6, 0, 0, 0, 0, 0, 0, 0 <50>
12101 DATA 0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <175>
12102 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <156>
12104 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <154>
12105 DATA 0, 0, 0, 0, 0, 0, 0, 1, 128, 0, 0, 0, 0, 0, 0, 0 <179>
12106 DATA 1, 131, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <243>
12107 DATA 0, 48, 0, 0, 48, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <133>
12108 DATA 3, 0, 0, 3, 6, 0, 0, 6, 0, 0, 6, 0, 0, 0, 0, 0 <88>
12109 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <159>
12110 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <164>
12112 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <162>
12113 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <163>
12114 DATA 0, 12, 0, 1, 140, 0, 1, 128, 0, 0, 0, 0, 0, 0, 0, 0 <86>
12115 DATA 0, 48, 0, 0, 48, 0, 0, 6, 0, 0, 6, 0, 0, 0, 0, 0 <249>
12116 DATA 0, 198, 0, 0, 192, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <172>
12117 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <167>
12118 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <172>
12120 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <170>
12121 DATA 0, 0, 0, 0, 24, 0, 0, 24, 0, 0, 24, 0, 0, 0, 0, 0 <145>
12122 DATA 0, 0, 0, 0, 0, 0, 0, 6, 0, 0, 6, 0, 0, 0, 0, 0 <0>
12123 DATA 6, 48, 0, 0, 49, 128, 0, 1, 128, 0, 0, 0, 0, 0, 0, 0 <57>
12124 DATA 0, 0, 0, 0, 0, 0, 0, 0, 96, 0, 0, 0, 0, 0, 0, 0 <48>
12125 DATA 0, 96, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 <101>
    
```


programme

12126 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<180>	20015 DATA 13, 92, 0, 53, 112, 0, 53, 112, 0	<168>
12128 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<178>	20016 DATA 213, 192, 0, 213, 192, 0, 255, 192, 0	<29>
12129 DATA 0, 48, 0, 0, 48, 0, 0, 0, 0, 0	<180>	20020 DATA 255, 0, 0, 235, 0, 0, 235, 0, 0	<182>
12130 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<177>	20021 DATA 235, 0, 0, 235, 0, 0, 235, 0, 0	<177>
12131 DATA 12, 48, 192, 12, 48, 192, 0, 0, 0, 0	<182>	20022 DATA 235, 0, 0, 235, 0, 0, 235, 0, 0	<178>
12132 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<159>	20023 DATA 235, 0, 0, 219, 0, 0, 215, 0, 0	<167>
12133 DATA 0, 48, 0, 0, 48, 0, 0, 0, 0, 0	<188>	20024 DATA 215, 0, 0, 215, 0, 0, 215, 0, 0	<114>
12134 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	<102>	20025 DATA 215, 255, 252, 213, 85, 92, 213, 85, 92	<1>
12136 DATA 0, 0, 0, 0, 0, 0, 0, 20, 0	<45>	20026 DATA 213, 85, 92, 213, 85, 92, 255, 255, 2	<14>
12137 DATA 0, 119, 0, 0, 247, 128, 1, 235, 192	<61>	52, 0	<121>
12138 DATA 3, 136, 224, 7, 8, 112, 7, 8, 112	<51>	20030 DATA 0, 60, 0, 0, 235, 0, 0, 235, 0	<212>
12139 DATA 14, 8, 56, 1, 255, 192, 14, 8, 56	<134>	20031 DATA 0, 235, 0, 0, 235, 0, 3, 170, 192	<3>
12140 DATA 7, 8, 112, 7, 8, 112, 3, 136, 224	<156>	20032 DATA 3, 190, 192, 3, 190, 192, 3, 190, 192	<92>
12141 DATA 1, 235, 192, 0, 247, 128, 0, 119, 0	<18>	20033 DATA 14, 190, 176, 14, 166, 112, 13, 85, 1	<31>
12142 DATA 0, 20, 0, 0, 0, 0, 0, 0, 0, 0	<108>	12	<186>
12144 DATA 0, 0, 0, 0, 0, 0, 0, 12, 0	<77>	20034 DATA 13, 125, 112, 53, 195, 92, 53, 195, 9	<75>
12145 DATA 0, 111, 0, 0, 239, 128, 1, 243, 192	<163>	2	<3>
12146 DATA 3, 144, 224, 7, 16, 112, 7, 16, 112	<162>	20035 DATA 53, 195, 92, 53, 195, 92, 215, 0, 215	<88>
12147 DATA 14, 11, 192, 14, 28, 56, 1, 232, 56	<137>	20036 DATA 215, 0, 215, 215, 0, 215, 255, 0, 255	<203>
12148 DATA 7, 4, 112, 7, 4, 112, 3, 132, 224	<254>	20040 DATA 255, 0, 255, 235, 0, 235, 235, 0, 235	<62>
12149 DATA 1, 231, 192, 0, 251, 128, 0, 123, 0	<6>	20041 DATA 235, 0, 235, 234, 192, 235, 234, 192,	<222>
12150 DATA 0, 24, 0, 0, 0, 0, 0, 0, 0	<211>	235	<29>
12151 DATA 0	<5>	20042 DATA 234, 176, 235, 235, 176, 235, 235, 17	<212>
12152 DATA 0, 0, 0, 0, 0, 0, 0, 28, 0	<254>	2, 235	<6>
12153 DATA 0, 95, 0, 0, 223, 128, 1, 227, 192	<220>	20043 DATA 235, 220, 219, 215, 215, 215, 215, 55	<211>
12154 DATA 3, 160, 224, 7, 16, 112, 7, 16, 192	<14>	215	<159>
12155 DATA 14, 11, 56, 14, 28, 56, 14, 104, 56	<190>	20044 DATA 215, 53, 87, 215, 13, 87, 215, 13, 87	<250>
12156 DATA 1, 132, 112, 7, 4, 112, 3, 130, 224	<197>	20045 DATA 215, 3, 87, 215, 3, 87, 215, 0, 215	<85>
12157 DATA 1, 227, 192, 0, 253, 128, 0, 125, 0	<156>	20046 DATA 215, 0, 215, 215, 0, 215, 255, 0, 255	<124>
12158 DATA 0, 28, 0, 0, 0, 0, 0, 0, 0	<73>	0	<35>
12160 DATA 0, 0, 0, 0, 0, 0, 0, 28, 0	<220>	20050 DATA 243, 224, 0, 146, 24, 0, 146, 4, 0	<189>
12161 DATA 0, 63, 0, 0, 191, 128, 1, 163, 192	<14>	20051 DATA 147, 194, 0, 144, 34, 0, 144, 17, 0	<210>
12162 DATA 3, 192, 224, 7, 32, 128, 7, 17, 112	<190>	20052 DATA 144, 17, 0, 144, 34, 0, 147, 194, 0	<177>
12163 DATA 14, 10, 56, 14, 28, 56, 14, 40, 56	<197>	20053 DATA 146, 4, 0, 146, 24, 0, 147, 224, 0	<116>
12164 DATA 7, 68, 112, 0, 130, 112, 3, 129, 224	<156>	20054 DATA 144, 0, 0, 144, 0, 0, 144, 0, 0	<205>
12165 DATA 1, 226, 192, 0, 254, 128, 0, 126, 0	<227>	20055 DATA 144, 0, 0, 144, 0, 0, 240, 0, 0	<21>
12166 DATA 0, 28, 0, 0, 0, 0, 0, 0, 0	<65>	20056 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<55>
12168 DATA 0, 0, 0, 0, 0, 0, 0, 28, 0	<228>	20060 DATA 240, 3, 192, 144, 2, 64, 144, 2, 64	<28>
12169 DATA 0, 127, 0, 0, 255, 128, 0, 227, 0	<22>	20061 DATA 72, 4, 128, 72, 4, 128, 72, 4, 128	<49>
12170 DATA 2, 128, 224, 7, 65, 112, 7, 34, 112	<67>	20062 DATA 36, 9, 0, 36, 9, 0, 36, 9, 0	<15>
12171 DATA 14, 20, 56, 14, 8, 56, 14, 20, 56	<63>	20063 DATA 18, 18, 0, 18, 18, 0, 18, 18, 0	<120>
12172 DATA 7, 34, 112, 7, 65, 112, 3, 128, 160	<216>	20064 DATA 9, 36, 0, 9, 36, 0, 9, 36, 0	<215>
12173 DATA 0, 99, 128, 0, 255, 128, 0, 127, 0	<77>	20065 DATA 6, 72, 0, 6, 72, 0, 6, 112, 0	<117>
12174 DATA 0, 28, 0, 0, 0, 0, 0, 0, 0	<14>	20066 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	<55>
12176 DATA 0, 0, 0, 0, 0, 0, 0, 28, 0	<236>	20070 DATA 0, 63, 128, 7, 32, 128, 25, 32, 128	<71>
12177 DATA 0, 126, 0, 0, 254, 128, 1, 226, 192	<30>	20071 DATA 33, 63, 128, 71, 0, 0, 72, 0, 0	<196>
12178 DATA 3, 129, 224, 0, 130, 112, 7, 68, 112	<79>	20072 DATA 72, 0, 0, 144, 0, 0, 144, 0, 0	<67>
12179 DATA 14, 40, 56, 14, 28, 56, 14, 10, 56	<180>	20073 DATA 144, 0, 0, 144, 0, 0, 144, 0, 0	<62>
12180 DATA 7, 17, 112, 7, 32, 128, 3, 192, 224	<96>	20074 DATA 144, 0, 0, 144, 0, 0, 72, 0, 0	<244>
12181 DATA 1, 163, 192, 0, 191, 128, 0, 63, 0	<188>	20075 DATA 72, 0, 0, 71, 0, 0, 33, 63, 128	<225>
12182 DATA 0, 28, 0, 0, 0, 0, 0, 0, 0	<93>	20076 DATA 25, 32, 128, 7, 32, 128, 0, 63, 128, 0	
12184 DATA 0, 0, 0, 0, 0, 0, 0, 28, 0	<244>	20080 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	
12185 DATA 0, 125, 0, 0, 253, 128, 1, 227, 192	<226>	20081 DATA 255, 255, 255, 250, 170, 175, 238, 17	
12186 DATA 3, 130, 224, 7, 4, 112, 1, 132, 112	<182>	0, 187	
12187 DATA 14, 104, 56, 14, 28, 56, 14, 11, 56	<189>	20082 DATA 235, 170, 235, 234, 235, 171, 234, 25	
12188 DATA 7, 16, 192, 7, 16, 112, 3, 160, 224	<104>	5, 171	
12189 DATA 1, 227, 192, 0, 223, 128, 0, 95, 0	<182>	20083 DATA 235, 190, 235, 238, 170, 187, 250, 17	
12190 DATA 0, 28, 0, 0, 0, 0, 0, 0, 0	<189>	0, 175	
12192 DATA 0, 0, 0, 0, 0, 0, 0, 24, 0	<104>	20084 DATA 255, 255, 255, 0, 0, 0, 0, 0, 0	
12193 DATA 0, 123, 0, 0, 251, 128, 1, 231, 192	<182>	20085 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	
12194 DATA 3, 132, 224, 7, 4, 112, 7, 4, 112	<248>	20086 DATA 0, 0, 0, 0, 0, 0, 0, 0, 0	
12195 DATA 1, 232, 56, 14, 28, 56, 14, 11, 192	<218>	20090 DATA 0, 0, 0, 0, 255, 255, 3, 170, 170	
12196 DATA 7, 16, 112, 7, 16, 112, 3, 144, 224	<49>	20091 DATA 14, 170, 170, 14, 86, 86, 254, 154, 1	
12197 DATA 1, 243, 192, 0, 239, 128, 0, 111, 0	<164>	02	
12198 DATA 0, 12, 0, 0, 0, 0, 0, 0, 0	<202>	20092 DATA 254, 154, 86, 14, 154, 90, 14, 154, 1	
20000 DATA 255, 252, 0, 234, 171, 192, 234, 170,	<223>	02	
176	<36>	20093 DATA 14, 170, 170, 14, 102, 86, 14, 102, 1	
20001 DATA 234, 170, 172, 235, 254, 172, 235, 3,	<183>	06	
171	<55>	20094 DATA 14, 102, 90, 14, 102, 106, 254, 154,	
20002 DATA 235, 3, 171, 235, 3, 171, 235, 254, 1	<253>	86	
71	<158>	20095 DATA 254, 170, 170, 14, 85, 85, 14, 170, 1	
20003 DATA 234, 170, 172, 234, 90, 172, 213, 86,	<106>	70	
176	<201>	20096 DATA 3, 170, 170, 0, 255, 255, 0, 0, 0, 0	
20004 DATA 215, 215, 192, 215, 215, 0, 215, 215,	<237>	20100 DATA 0, 0, 0, 255, 255, 170, 170, 170	
0		20101 DATA 170, 170, 170, 154, 90, 101, 102, 102	
20005 DATA 215, 53, 192, 215, 53, 192, 215, 53,		102	
192		20102 DATA 102, 102, 102, 102, 102, 102, 154, 10	
20006 DATA 215, 13, 112, 215, 13, 112, 255, 15,		2, 101	
240, 0		20103 DATA 170, 170, 170, 86, 106, 154, 102, 106	
20010 DATA 255, 0, 63, 235, 0, 59, 235, 0, 59		102	
20011 DATA 58, 192, 235, 58, 192, 235, 14, 179,		20104 DATA 86, 106, 86, 90, 106, 102, 102, 86, 1	
172		02	
20012 DATA 14, 179, 172, 3, 174, 176, 3, 170, 17		20105 DATA 170, 170, 170, 85, 85, 85, 170, 170,	
6			
20013 DATA 0, 234, 192, 0, 234, 192, 0, 229, 192			
20014 DATA 3, 87, 0, 3, 87, 0, 13, 92, 0			


```

170
20106 DATA 170, 170, 170, 255, 255, 255, 0, 0, 0, 0
20110 DATA 0, 0, 0, 255, 192, 0, 170, 176, 0
20111 DATA 170, 172, 0, 106, 172, 0, 170, 172, 0
20112 DATA 165, 172, 0, 170, 172, 0, 106, 172, 0
20113 DATA 170, 172, 0, 86, 172, 0, 106, 172, 0
20114 DATA 86, 172, 0, 102, 172, 0, 86, 172, 0
20115 DATA 170, 172, 0, 85, 108, 0, 170, 172, 0
20116 DATA 170, 176, 0, 255, 192, 0, 0, 0, 0, 0
31998 FOR X=12288TD 13823:READ DT:POKE X,DT:NEXT
31999 FOR X=12288TD12288+1024:POKE X+1024,255-PEE
K(X):NEXT X
32000 DATA219,219,24,255,255,24,219,219,0,44,102,
110,102,102,102,0,108,102,102
32001 DATA108,102,102,108,0,44,102,96,96,96,102,
44,0,104,108,102,102,102,108
32002 DATA104,0,110,96,96,104,96,96,110,0,110,96,
96,104,96,96,96,0,44,102,96
32003 DATA110,102,102,44,0,102,102,102,110,102,1
02,102,0,44,8,8,8,8,4,0,14
32004 DATA12,12,12,12,108,40,0,102,108,104,96,10
4,108,102,0,96,96,96,96,96,96
32005 DATA110,0,98,102,110,106,99,99,99,0,102,10
2,110,110,110,102,102,0,44,102
32006 DATA102,102,102,102,44,0,108,102,102,108,9
6,96,96,0,44,102,102,102,102
32007 DATA44,14,0,108,102,102,108,104,108,102,0,
44,102,96,44,6,102,44,0,110,8
32008 DATA8,8,8,8,0,102,102,102,102,102,102,44,
0,102,102,102,102,102,44,8,0
32009 DATA98,98,98,106,111,103,99,0,102,102,44,8,
44,102,102,0,102,102,102,44
32010 DATA8,8,8,0,110,6,12,8,32,96,110,0,44,32,3
2,32,32,44,0,12,2,32,108,32
32011 DATA98,236,0,44,12,12,12,12,44,0,0,8,44,
110,8,8,8,0,0,32,110,111,32
32012 DATA0,0,0,0,0,0,0,0,0,8,8,8,8,0,0,8,0,10
2,102,102,0,0,0,0,255,4,4,4
32013 DATA255,32,32,32,128,192,224,240,248,252,2
54,255,24,56,120,248,248,120,56
32014 DATA24,165,36,251,0,0,231,36,165,6,12,8,0,
0,0,0,0,12,8,32,32,32,8,12,0,32
32015 DATA8,12,12,12,8,32,0,0,102,44,238,44,102,
0,0,0,8,8,110,8,8,0,0,0,0,0,0
32016 DATA0,8,8,32,0,0,0,110,0,0,0,0,0,0,0,0,0,8
,8,0,0,2,6,12,8,32,96,0,44,102
32017 DATA110,102,102,102,44,0,8,8,40,8,8,8,110,
0,44,102,6,12,32,96,110,0,44
32018 DATA102,6,12,6,102,44,0,6,14,14,102,111,6,
6,0,110,96,108,6,6,102,44,0,44
32019 DATA102,96,108,102,102,44,0,110,102,12,8,8
,8,8,0,44,102,102,44,102,102
32020 DATA44,0,44,102,102,46,6,102,44,0,0,0,8,0,
0,8,0,0,0,0,8,0,0,8,8,32,14,8
32021 DATA32,96,32,8,14,0,0,0,110,0,110,0,0,0,96
,8,12,6,12,8,96,0,44,102,6,12
32022 DATA8,0,8,0,0,0,0,238,239,0,0,0,0,40,108,2
38,111,12,46,0,32,32,32,32,8
32023 DATA8,8,0,0,0,239,239,0,0,0,0,0,239,239,0
,0,0,0,0,239,239,0,0,0,0,0
32024 DATA0,0,0,239,239,0,0,32,32,32,32,32,32,32
,32,12,12,12,12,12,12,12,0
32025 DATA0,0,224,224,0,8,8,8,12,15,7,0,0,0,0,0
,192,224,232,0,0,0,0,192,192
32026 DATA192,192,192,239,239,192,224,96,40,
12,14,7,3,3,7,14,12,40,96,224
32027 DATA192,239,239,192,192,192,192,192,23
9,239,3,3,3,3,3,0,44,110,110
32028 DATA110,110,44,0,0,0,0,0,239,239,0,38,11
1,111,111,46,12,8,0,96,96,96
32029 DATA96,96,96,96,0,0,0,7,15,12,8,8,195,2
31,110,44,44,110,231,195,0,44
32030 DATA110,102,102,110,44,0,8,8,102,102,8,8,4
4,0,6,6,6,6,6,6,6,6,8,12,46,111
32031 DATA46,12,8,0,8,8,8,239,239,8,8,8,192,192,
32,32,192,192,32,32,8,8,8,8
32032 DATA8,8,0,0,3,46,102,38,38,0,239,111,47,
15,15,7,3,1,0,0,0,0,0,0,0,224
32033 DATA224,224,224,224,224,224,0,0,0,0,23
9,239,239,239,239,0,0,0,0,0
32034 DATA0,192,224,225,235,207,206,196,192,97,2
25,225,227,99,98,102,103,199
32035 DATA199,192,96,96,32,32,224,239,239,32,32,
32,32,32,32,174,175,35,32,32
32036 DATA35,47,46,11,139,203,201,201,200,136,9,
1,131,199,239,239,238,238,239
<18>
32037 DATA128,128,128,0,0,0,0,192,192,192,192,
0,64,101,69,103,110,108,108,0
<193>
32038 DATA0,40,40,224,40,8,8,0,0,136,200,32,32,3
2,32,0,0,0,0,39,39,35,34,0,3
<73>
32039 DATA9,41,9,11,139,11,128,131,135,198,239,1
99,128,7,13,0,0,13,0,128,0,0
<119>
<127>
<15>
<175>
<10>
<211>
<252>
32042 DATA236,236,192,195,227,227,97,97,100,46,1
43,140,140,198,198,6,0,0,34,46
<230>
32043 DATA0,0,0,0,0,0,170,238,0,0,0,0,0,0,1,3,
3,2,0,0,0,232,224,224,40,40,40
<92>
32044 DATA12,12,12,0,0,0,0,195,153,145,145,
159,153,195,255,231,195,153
<90>
32045 DATA129,153,153,153,255,131,153,153,131,15
3,153,131,255,96,153,159,159
<139>
32046 DATA159,153,195,255,135,147,153,153,153,14
7,135,255,129,159,135,159
<200>
32047 DATA159,129,255,129,159,159,135,159,159,15
9,255,195,153,159,145,153,153
<199>
32048 DATA195,255,153,153,153,129,153,153,25
5,195,231,231,231,231,231,195
<98>
32049 DATA255,225,243,243,243,243,147,199,255,15
3,147,135,143,136,147,153,255
<234>
32050 DATA159,159,159,159,159,129,255,156,13
6,128,148,156,156,156,255,153
<210>
32051 DATA137,129,129,145,153,153,255,195,153,15
3,153,153,153,195,255,131,153
<233>
32052 DATA153,131,159,159,255,195,153,153,15
3,153,195,241,255,131,153,153
<28>
32053 DATA131,135,147,153,255,195,153,159,195,24
9,153,195,255,129,231,231,231
<98>
32054 DATA231,231,255,153,153,153,153,153,153,15
3,195,255,153,153,153,153,153
<129>
32055 DATA195,231,255,156,156,156,148,128,136,15
6,255,153,153,195,231,195,153
<85>
32056 DATA153,255,153,153,153,195,231,231,231,25
5,129,249,243,231,207,159,129
<165>
32057 DATA255,195,207,207,207,207,195,255,24
3,237,207,131,207,157,3,255,195
<125>
32058 DATA243,243,243,243,195,255,255,231,19
5,129,231,231,231,231,255,239
<156>
32059 DATA207,128,128,207,239,255,207,239,255,56
,16,0,150,57,214,254,16,124,121
<166>
32060 DATA16,249,255,248,254,254,252,16,22,251,2
51,16,215,16,254,222,16,88,244
<151>
32061 DATA85,254,20,255,16,16,121,254,16,254,16,
224,233,0,0,224,251,16,16,253
<200>
32062 DATA16,222,16,56,200,232,8,0,0,128,64,128,
249,254,80,254,82,80,254,255
<249>
32063 DATA215,114,80,16,121,212,84,253,255,244,2
54,92,64,222,121,215,6,246,251
<78>
32064 DATA254,84,215,116,246,206,118,242,24,222,
0,121,190,249,249,249,223,68
<253>
32065 DATA84,84,249,152,252,251,254,84,84,242,22
3,214,161,114,251,246,246,84
<218>
32066 DATA120,128,226,226,0,224,96,128,162,226,0
,226,226,0,221,32,0,252,80,224
<9>
32067 DATA224,32,255,121,64,0,226,226,226,96,226
,226,64,48,0,0,98,66,80,48,98
<182>
32068 DATA96,255,226,226,240,226,226,48,98,48,22
6,80,226,226,226,192,120,224
<103>
32069 DATA226,226,104,226,226,0,0,0,0,71,0,240,8
0,0,32,80,16,80,86,208,16,48
<136>
32070 DATA0,0,80,255,82,0,64,0,64,64,255,16,96
,80,225,112,8,2,56,24,82,48,32
<234>
32071 DATA0,118,160,80,255,80,193,32,32,6,0,242,
80,80,116,84,16,249,196,3,80
<76>
32072 DATA196,202,152
<107>
41999 FOR X=2048 TD 2093:READ DT:POKE X,DT:NEXT
X
<241>
42000 DATA162,8,189,31,49,24,10,144,2,9,1,157,31
,49,202,138,208,240,230,2,165
<14>
42001 DATA2,41,1,208,230,96,162,8,189,31,49,24,1
0,144,2,9,1,157,31,49,202,138
<172>
42002 DATA208,240,96
<173>
43000 A$=CHR$(34):PRINT"(BLACK CLEAR DOWN):LOAD"
A$:"PRG4"A$,B"
<104>
43001 PRINT"(DOWN):RUN"
<102>
43002 PRINT"(HOME)":POKE 631,13:POKE 632,13:POKE
198,2
<167>
ENDE DES LISTINGS
<53>

```


Teil 4

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 REM" (SU SC11)R (SPACE)Y (SPACE)L (SPACE)A (SPACE)N
(SC11 SI
5 REM" (SB) *EX-LMH-SYSTEM (SPACE) WIRD (SPACE) BENDET
IGT!* (SB
6 REM" (SJ SC31 SK
10 GR=4:BS=1:LE=1:LI=5:HT=0
11 FORX=54272T054296:POKEX,0:NEXT
12 BYB49269,15:SYSS0642:POKES3272,PEEK(53272)AND
2410R12
14 IFPEEK(12576)<>128THENSYS2073:GOTO 14
20 PRINT"(CLEAR)":SYS49152,0,0:SYS 49171,4,90,0,
0,255,255,65
30 SYS 49279,4,90,0,0,255,255,65
40 SYS 49348,4,90,0,0,255,255,65
50 FORX=0T02048STEP32:SYS49702,1114,X
70 SYS 49742,1114+10*SIN(X/20),X
80, SYS 49781,1114+20*SIN(X/10),X:NEXTX
100 PRINT"(BLACK RIGHT11)PARAVISION (SPACE)COMPAN
Y"
110 SYS49240,11:GOSUB60000:SYS49240,12:GOSUB6000
0:SYS49240,15:GOSUB60000
140 SYS 49240,7:GOSUB 60000
141 SYS 49171,4,90,8,128,255,255,64
142 SYS 49279,4,90,8,128,255,255,64
143 SYS 49348,4,90,8,128,255,255,64
150 FORX=2048T04096STEP32:SYS49702,1114,X
170 SYS 49742,1114+10*SIN(X/20),X
180 SYS 49781,1114+20*SIN(X/10),X:NEXTX
200 PRINT"(DOWN8)"
470 SYS 50688,31:SYS 50203,153,154,155,156,157,1
58,159,160:POKE 53264,224
500 SYS 50698,255,255
510 SYS 49985,40,90,90,140,90,190,90,240,90,4
0,80,40,140,40,200
520 SYS 50130,6,6,6,6,6,6,7,7:SYS 50655,2,4
540 SYS 50674,1:GOSUB 55000
550 SYS 50674,3:GOSUB 55000
560 SYS 50674,7:GOSUB 55000
570 SYS 50674,15:GOSUB 55000
580 SYS 50674,31:GOSUB 55000
590 SYS 50674,63:GOSUB 55000
600 SYS 50674,127:GOSUB 55000
610 SYS 50674,255:GOSUB 55000
620 PRINT"(DOWN RIGHT7 GREEN)-(SPACE)POSTAL (SPAC
E)SERVICES (SPACE)-"
630 SYS 50334,0,0,0,0,0,0,0,0,0,0,0,253,0,253,0,
253
640 SYS 50479,153,153,154,154,155,155,156,156,15
7,157,158,158,159,159,160,160
650 SYS 50624
670 PRINT"(YELLOW DOWN4 RIGHTS SPACE2)PRESS (SPAC
E)S (SPACE)FOR (SPACE)START"
675 PRINT"(HOME DOWN SPACE4)PARAVISION (SPACE)COM
FANY (SPACE)PRESENTS: (SPACE)"
677 PRINT"(RIGHT SPACE RIGHT SPACE RIGHT SPACE R
IGHT SPACE RIGHT SPACE RIGHT BLUE)ULRICH (SPACE)M
UEHL "S"
680 SYS 50655,2,6:SYS50130,1,12,0,12,1,7,7,7:GOS
UB 60000
690 SYS 50655,10,14:SYS50130,15,11,11,11,15,7,7,
7:GOSUB 60000
700 SYS 50655,7,13:SYS50130,12,0,12,0,12,7,7,7:G
OSUB 60000
710 SYS 50655,13,5:SYS50130,11,11,15,11,11,7,7,7
:GOSUB 60000
720 SYS 50655,5,6:SYS50130,0,12,1,12,0,7,7,7:GOS
UB 60000
725 SYS 50655,14,13:SYS50130,15,15,15,12,11,7,7,
7:GOSUB 60000
730 GET A$:IF A$="" THEN 680
735 IF A$<>"S" THEN 690
740 SYS 50655,12,2:GOSUB 60000
750 SYS 50655,11,9:GOSUB 60000
760 SYS 50655,0,0:GOSUB 60000
770 SYS 50130,6,6,6,6,6,6,15,15,15:GOSUB 60000
780 SYS 50130,6,6,6,6,6,6,12,12,12:GOSUB 60000
790 SYS 50130,6,6,6,6,6,6,11,11,11:GOSUB 60000
800 SYS 50130,6,6,6,6,6,6,0,0,0:GOSUB 60000
810 SYS 50130,13,13,13,13,13,0,0,0:GOSUB 60000
820 SYS 50130,14,14,14,14,14,0,0,0:GOSUB 60000
830 SYS 50130,15,15,15,15,15,0,0,0:GOSUB 60000
840 SYS 50130,12,12,12,12,12,0,0,0:GOSUB 60000
850 SYS 50130,11,11,11,11,11,0,0,0:GOSUB 60000
860 SYS 50130,0,0,0,0,0,0,0,0,0:GOSUB 60000
870 SYS 49240,14:GOSUB 60000
880 SYS 49240,12:GOSUB 60000
890 SYS 49240,11:GOSUB 60000
900 SYS 49240,0:GOSUB 60000
910 PRINT"(CLEAR)":SYS 50674,0:SYS 50642
930 PRINT"(GREEN DOWN7)WOLLEN (SPACE)SIE (SPACE)SP
IELPARAMETER (SPACE)AENDERN (SPACE)J/N (SPACE)?"
940 SYS 51287,0,0,0,0,11,11,11,11:SYS 51387,233
950 SYS 51360:SYS 49152,0,0:GET A$:IF A$<>"J" AN
D A$<>"N" THEN 950
960 IF A$="N" THEN 1090
980 PRINT"(CLEAR RIGHT13 YELLOW)PARAMETERMENUE"
990 PRINT"(RIGHT13 RED)-----"
1000 PRINT"(DOWN2 RIGHT4 L10.GREEN) (1) (SPACE)BES
CHWINDIGKEIT (SPACE)RYLAN (SPACE3)";GR
1010 PRINT"(DOWN2 RIGHT4 L10.BLUE) (2) (SPACE2)BEE
INFLUSSUNG (SPACE)SATELLIT (SPACE)";BS
1020 PRINT"(CYAN DOWN3 RIGHT11)WELCHEN (SPACE)1/2
(SPACE)?"
1025 PRINT"(CYAN DOWN RIGHT11) 'S' (SPACE)FUER (SPA
CE)START"
1030 SYS 51287,0,0,0,0,11,12,15,1:SYS 51387,233
1040 SYS 51360:SYS 49152,0,0:GET A$:IF A$<>"1" A
ND A$<>"2" AND A$<>"S" THEN 1040
1041 IF A$="S" THEN 1090
1045 IF A$="2" THEN 1070
1050 IFA$="1" THEN PRINT"(UP11)":INPUT"(RIGHT31)"
;BR
1060 GOTO 980
1070 PRINT"(UP8)":INPUT"(RIGHT31)":BS
1080 GOTO 980
1090 SYS 51462,GR,GR:SYS 51481,BS,BS
1100 PRINT"(CLEAR DOWN10 RIGHT14 GREEN)GET (SPACE
)READY"
1110 FORX=0T0100:SYS49171,INT(255*RND(1)),0,0,0,
239,0,17:NEXT X
1120 SYS 49171,0,0,0,0,0,0,0
1130 ON LE GOTO 5000,5100,5200,5300,5400,5500,56
00,5700,5800,5900
1140 POKE 53278,0:SYS 49269,3:SYS 49171,34,207,0
,0,10,205,17
1145 SYS 49279,34,200,0,0,10,205,17:SYS 49348,8,
180,0,0,10,205,129
1147 GOSUB 2000:SYS 50624
1160 FORR=0T01500:SYS2048:SYS51500:IF (PEEK(53278
)AND1)<>0 THEN HT=1:R=1500
1170 NEXT R
1180 IF HT=0 THEN 1500
1190 HT=0:SYS 49152,0,11:GOSUB 60000
1200 SYS 49152,0,12:GOSUB 60000
1210 SYS 49152,0,15:SYS 50130,1,12,12,12,12,1,2,12:GOSUB 60000
1220 SYS 49152,0,1:SYS 50130,15,11,11,11,11,1,1,11:GOSUB 60000
1230 SYS 49152,0,15:SYS 50130,12,0,0,0,0,0,0,0:SYS
49240,4:GOSUB 60000
1240 SYS 49152,0,12:SYS 50130,11,0,0,0,0,0,0,0:SYS
49240,8:GOSUB 60000
1250 SYS 49152,0,11:SYS 50130,10,0,0,0,0,0,0,0:SYS
49240,11:GOSUB 60000
1260 SYS 49152,0,0:SYS 50130,0,0,0,0,0,0,0,0:SYS
49240,0:GOSUB 60000
1270 FORX=0T085:SYS49171,INT(255*RND(1)),0,0,0,0,
205,16:SYS49669,150-X,0,128
1280 SYS49279,INT(255*RND(1)),0,0,0,0,205,16:NEX
T X
1290 SYS 50642:SYS 50674,0:LI=LI-1
1300 PRINT"(CLEAR DOWN12 RIGHT6)THERE (SPACE)ARE (
SPACE)";LI;"LIVES (SPACE)LEFT (SPACE)!"
1310 SYS 49240,11:GOSUB 60000
1320 SYS 49240,12:GOSUB 60000
1330 SYS 49240,15:GOSUB 60000
1370 SYS 49240,7:GOSUB 60000
1380 SYS 49240,15:GOSUB 60000
1390 SYS 49240,12:GOSUB 60000
1400 SYS 49240,15:GOSUB 60000
1410 SYS 49240,1:GOSUB 60000
1420 FOR WT=0 TO 500:NEXT WT
1430 IF LI>0 THEN 1090
1440 PRINT"(CLEAR BLACK DOWN10 RIGHT15)GAME (SPAC
E)OVER!"
1450 SYS 49240,11:GOSUB 60000

```



```

1455 SYS 49240,12:G0SUB 60000
1460 SYS 49240,15:G0SUB 60000
1465 SYS 49240,7:G0SUB 60000
1470 SYS 49240,1:G0SUB 60000
1475 FORX=100TOSTEP-1:SYS49171,17,103,0,0,170,7
0,21:SYS49279,X,0,0,0,170,70,21
1480 SYS49348,INT(255*RND(1)),0,0,0,170,70,129:N
EXTX
1490 FOR WT=0 TO 250:NEXT WT:RUN
1500 FORX=PEEK(53248) TO255:SYS2048:SYS49171,X,X,
0,0,170,170,21:POKE53248,X:NEXT
1510 SYS 50674,0:PRINT"BLACK CLEAR DOWN RIGHT
Z>GREAT(SPACE)!" :POKE 54290,0
1515 SYS 49240,5:G0SUB 60000
1520 SYS 49240,3:G0SUB 60000
1525 SYS 49240,13:G0SUB 60000
1530 SYS 49240,7:G0SUB 60000
1540 PRINT"DOWN2 RIGHT10 GREEN>YOU(SPACE)MADE(S
PACE)LEVEL(SPACE)";LE;"(SPACE)!!"
1550 FORX=0 TO50:SYS49171,X,X,0,X,253,205,81:SYS4
9279,X,X+100,0,0,170,205,17
1560 NEXTX:SYS49603,0,0,0:SYS49636,0,0,0:SYS 496
69,0,0,0
1570 FOR WT=0 TO 1000:NEXT WT:LE=LE+1
1580 IF LE<11 THEN 1090
1590 PRINT"DOWN2 RIGHT3 LIG.BLUE>YOU(SPACE)HAVE
(SPACE)ALB(SPACE)COMPLETED(SPACE)THIS(SPACE)GAM
E(SPACE)!!!"
1600 PRINT"DOWN2 RIGHT6 LIG.GREEN>PLEASE(SPACE)
CONTACT(SPACE)US(SPACE)FOR(SPACE)NEW(SPACE)JOBS"
1610 FORMH=0 TO10:FORX=150 TO250STEP15:SYS49171,X,
X,0,0,170,205,20
1620 SYS49279,X,X-100,0,0,170,205,16:NEXTX:NEXTW
H
1630 FOR WT=0 TO 1000:NEXT WT:RUN
1999 END
2000 PRINT"HOME DOWN":PRINTTAB(33)"(SPACE6)"
2010 PRINTTAB(33)"(YELLOW SPACE)PARA(SPACE)"
2020 PRINTTAB(33)"VISION"
2030 PRINTTAB(33)"(SPACE6)"
2040 PRINTTAB(33)"(WHITE SPACE)RRRR(SPACE)"
2050 PRINTTAB(33)"(SPACE6)"
2060 PRINTTAB(33)"(YELLOW SPACE)YYYY(SPACE)"
2070 PRINTTAB(33)"(SPACE6)"
2080 PRINTTAB(33)"(HGREY SPACE)LLLL(SPACE)"
2090 PRINTTAB(33)"(SPACE6)"
2100 PRINTTAB(33)"(HGREY SPACE)AAAA(SPACE)"
2110 PRINTTAB(33)"(SPACE6)"
2120 PRINTTAB(33)"(DGREY SPACE)NNNN(SPACE)"
2130 PRINTTAB(33)"(SPACE6)"
2140 PRINTTAB(33)"(SPACE6)"
2150 PRINTTAB(33)"(SPACE6)"
2160 PRINTTAB(33)"(GREEN)LEVEL:"
2170 PRINTTAB(33)"(LIG.GREEN SPACE6)";:PRINT"(LE
FTS)";LE
2180 PRINTTAB(33)"(SPACE6)"
2190 PRINTTAB(33)"(SPACE6)"
2200 PRINTTAB(33)"(SPACE6)"
2240 RETURN
5000 REM LEVEL 1
5010 PRINT"(RED CLEAR)":SYS 49444,36
5020 SYS 50479,136,136,128,135,128,135,128,135,1
28,135,128,135,128,135,128,135
5030 SYS 50130,7,14,14,14,14,13,2,2
5040 SYS 50203,136,128,128,128,128,128,128,128
5050 SYS 50698,0,0:SYS 50688,0
5060 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5070 SYS 50334,0,0,254,1,254,255,253,0,254,1,254
,255,254,0,253,0
5080 SYS 50674,31:SYS 51699,0
5090 GOTO 1140
5100 REM LEVEL 2
5110 PRINT"(BLUE CLEAR)":SYS 49444,36
5120 SYS 50479,136,136,128,135,128,135,128,135,1
28,135,128,135,128,135,128,135
5130 SYS 50130,7,7,7,7,7,2,2,2
5140 SYS 50203,136,128,128,128,128,128,128,128
5150 SYS 50698,254,254:SYS 50688,0
5160 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5170 SYS 50334,0,0,254,1,254,255,253,0,254,1,254
,255,254,0,253,0
5180 SYS 50674,63:SYS 51699,0
5190 GOTO 1140
5200 REM LEVEL 3
<138>
<170>
<69>
<26>
<82>
<153>
<188>
<30>
<119>
<103>
<92>
<217>
<134>
<2>
<227>
<156>
<82>
<30>
<110>
<125>
<36>
<157>
<37>
<86>
<200>
<120>
<246>
<172>
<239>
<193>
<89>
<213>
<54>
<233>
<103>
<253>
<150>
<17>
<27>
<37>
<165>
<67>
<67>
<77>
<87>
<86>
<97>
<255>
<219>
<65>
<186>
<111>
<199>
<79>
<89>
<51>
<204>
<108>
<64>
<117>
<31>
<186>
<44>
<180>
<232>
<152>
<56>
5210 PRINT"(GREEN CLEAR)":SYS 49444,36
5220 SYS 50479,136,136,137,144,137,144,137,144,1
28,135,128,135,128,135,128,135
5230 SYS 50130,7,7,1,7,7,1,7,7
5240 SYS 50203,136,137,137,137,128,128,128,128
5250 SYS 50698,240,240:SYS 50688,0
5260 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5270 SYS 50334,0,0,253,1,253,255,254,0,254,1,252
,254,254,0,253,0
5280 SYS 50674,127:SYS 51699,0
5290 GOTO 1140
5300 REM LEVEL 4
5310 PRINT"(DGREY CLEAR)":SYS 49444,36
5320 SYS 50479,136,136,137,144,137,144,137,144,1
37,144,128,135,128,135,128,135
5330 SYS 50130,7,13,13,13,13,1,13,1
5340 SYS 50203,136,137,137,137,137,128,128,128
5350 SYS 50698,224,224:SYS 50688,0
5360 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5370 SYS 50334,0,0,253,2,253,255,253,0,254,1,252
,254,253,255,253,0
5380 SYS 50674,127:SYS 51699,0
5390 GOTO 1140
5400 REM LEVEL 5
5410 PRINT"(ORANGE CLEAR)":SYS 49444,36
5420 SYS 50479,136,136,145,152,137,144,137,144,1
28,135,145,152,128,135,128,135
5430 SYS 50130,7,1,1,7,7,1,1,1
5440 SYS 50203,136,145,137,137,128,145,128,128
5450 SYS 50698,0,0:SYS 50688,0
5460 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5470 SYS 50334,0,0,253,1,253,255,254,1,254,2,252
,254,252,0,253,0
5480 SYS 50674,127:SYS 51699,0
5490 GOTO 1140
5500 REM LEVEL 6
5510 PRINT"(CYAN CLEAR)":SYS 49444,36
5520 SYS 50479,136,136,145,152,137,144,137,144,1
28,135,145,152,128,135,128,135
5530 SYS 50130,7,1,1,1,1,1,1,1
5540 SYS 50203,136,145,137,137,128,145,128,128
5550 SYS 50698,0,0:SYS 50688,0
5560 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5570 SYS 50334,0,0,254,2,254,3,254,254,254,255,2
53,0,253,0,254,1
5580 SYS 50674,255:SYS 51699,0
5590 GOTO 1140
5600 REM LEVEL 7
5610 PRINT"(ORANGE CLEAR)":SYS 49444,36
5620 SYS 50479,136,136,145,152,137,144,137,144,1
28,135,145,152,128,135,128,135
5630 SYS 50130,7,7,7,7,7,1,1,1
5640 SYS 50203,136,145,137,137,128,145,128,128
5650 SYS 50698,0,0:SYS 50688,0
5660 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5670 SYS 50334,0,0,253,255,253,255,253,0,253,0,2
52,0,252,1,253,1
5680 SYS 50674,255:SYS 51699,0
5690 GOTO 1140
5700 REM LEVEL 8
5710 PRINT"(LIG.GREEN CLEAR)":SYS 49444,36
5720 SYS 50479,136,136,145,152,145,152,145,152,1
45,152,145,152,145,152,145,152
5730 SYS 50130,7,6,6,6,6,6,6,6
5740 SYS 50203,136,145,145,145,145,145,145,145
5750 SYS 50698,0,0:SYS 50688,0
5760 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
5770 SYS 50334,0,0,253,1,253,1,253,1,253,1,252,1
,252,1,253,1
5780 SYS 50674,255:SYS 51699,0
5790 GOTO 1140
5800 REM LEVEL 9
5810 PRINT"(PURPLE CLEAR)":SYS 49444,36
5820 SYS 50479,136,136,137,144,137,144,137,144,1
37,144,137,144,137,144,137,144
5830 SYS 50130,7,7,7,7,7,7,7,7
5840 SYS 50203,136,137,137,137,137,137,137,137
5850 SYS 50698,0,0:SYS 50688,0
5860 SYS 49985,50,140,0,20,0,50,0,80,0,110,0,140
,0,170,0,200
<206>
<158>
<81>
<128>
<168>
<144>
<150>
<131>
<252>
<163>
<157>
<0>
<5>
<227>
<64>
<244>
<185>
<232>
<97>
<15>
<192>
<209>
<226>
<26>
<157>
<89>
<65>
<76>
<197>
<122>
<126>
<53>
<146>
<126>
<1>
<189>
<133>
<191>
<41>
<229>
<136>
<153>
<46>
<227>
<102>
<34>
<218>
<36>
<142>
<81>
<53>
<22>
<139>
<181>
<202>
<134>
<192>
<136>
<242>
<188>
<162>
<240>
<95>
<214>
<46>
<234>

```


Zeitschriften und Programme von 1986

Heft 1/86

Commodore

Bestell-Nr. COM CK 1/2 16,- DM
 Bestell-Nr. COM CD 1/2 20,- DM
 Checksummer 1.0 G, Astronaut Battle (C-64), Hueycobra (C-16/116), Hardcopy-routine (C-16/116), Helmut und die Zeitmaschine (C-16/116), Locos (VC-20), Rock'me (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, Basis-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 (VC20), Falcons Lander (VC20), Frogger (VC20), The Games (C64), Cave of Death (C64) Sprites and Sound mit dem UMH-System (C64), Diggy (C16/116), Tapedirectory (C16/116), Reversi (C16/116)

Schneider

Bestell-Nr. COM SK 9/2 16,- DM
 Bestell-Nr. COM SD 9/2 39,- DM
 Creepy, Briefeschreiben, Frutty Man, Mirror Battle

Aus diesem Heft:

Commodore

Bestell-Nr. COM CK 1/3 16,- DM
 Bestell-Nr. COM CD 1/3 20,- DM

Schneider

Bestell-Nr. COM SK 1/3 16,- DM
 Bestell-Nr. COM SD 1/3 39,- DM

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
 Disksort, Mini-Farth, Starcommand

Heft 10/86

Commodore

Bestell-Nr. COM CK 10/2 16,- DM
 Bestell-Nr. COM CD 10/2 20,- DM
 Laufschrift (VC20), Checksummer (VC20), Treasure Robber (VC20), Scramble (VC20), Refusal-Basiclader (C-16), Hardy-Copy-Blitz (C-16), Autostart (C-16), Fill-Utility (C-16), Railway (C-16), Domino (C-16), Sprite Turn/Character-Switch (C-64), Checksummer (C-64), Horse Games (C-64), Patrouille (C-64)

Schneider

Bestell-Nr. COM SK 10/2 16,- DM
 Bestell-Nr. COM SD 10/2 39,- DM
 Checksummer, Panic-Car, Scurfy Joe, Drawgenerator, Airlighter

Alle hier aufgeführten Kassetten und Disketten können jederzeit nachbestellt werden.

Bei den Zeitschriften sind die Ausgaben 5, 9, 12 (85) und 5, 6, 7, 8, 9, 10, 11 (86) noch vorhanden.

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), Starship (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-Neil (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
 Paker, Cave-Walker, Diskettenarchiv

Heft 11/86

Commodore

Bestell-Nr. COM CK 11/2 16,- DM
 Bestell-Nr. COM CD 11/2 20,- DM
 Cooky (VC20), Gambol (VC20), Wostok (VC20), Topball (C16), Zauberpuzzle (C16), Schieberoutine (C16), Atlantis (C64), Astronomie (C64), Deutsche Fehlermeldung (C64), Rem-zellenkiller (C64)

Schneider

Bestell-Nr. COM SK 11/2 16,- DM
 Bestell-Nr. COM SD 11/2 39,- DM
 Diskettenhilfe, Horror Driver, Disc of Tron, Space Hawk, Morsedecoder

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 Höhlen (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), Platinenlayout (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

Heft 12/86

Commodore

Bestell-Nr. COM CK 12/2 16,- DM
 Bestell-Nr. COM CD 12/2 20,- DM
 Ballon Hopper (VC20), Fight Night (VC20), King of Kings (C16/116), Galaxis (C16/116), Sprites (C16/116), Atom (C64), Extended-UMH (C64), IRQ-Directory (C64)

Schneider

Bestell-Nr. COM SK 12/2 16,- DM
 Bestell-Nr. COM SD 12/2 39,- DM
 Alien Attack, Lucky Luck, Mouse, Checksummer, Tracer

Superpakete!

Beachten Sie bitte unsere neu geschnürten Programmpakete. ... und das zu einem Superpreis!

Neu: Superpakete zum Superpreis!

C=64

Superpaket 1 Spiele

Monster Attack, Projekt, High Noon, Brieftaube, Cadillon, Turbo Jet, Sea Attack, The Maze Graze Game, Galaxi, Space Comets, Sheet, Vampir, Memory

SP-C-K-1 Kassette 20,-
SP-C-D-1 Diskette 20,-

Superpaket 2 Spiele

Tank Duell, Astronaut, Jungle 1, Jungle 2, Sufo, Fire-Legion, Spiders, Star Battle, Super Breakout, Mission Rescue, Roadpainter, Moonsweeper, Scotti

SP-C-K-2 Kassette 20,-
SP-C-D-2 Diskette 20,-

Superpaket 3 Anwender

The Basic, Basicreset, Data-Generator, Print-Generator, Supermon-64, Restore N, Disky 10, Fill-Loader, Funktionstasten-Belegung, Autostart, Multibasic, Bildeditor, Spritehilfe + Demo, Autonumber, Micro-Tools + Mloader

SP-C-K-3 Kassette 20,-
SP-C-D-3 Diskette 20,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Tron/Mercurious	24,50/29,50	C 13
Interceptor		
Base/Schotter	24,50/29,50	C 23
Jump Man!		
The Way/Space Taxi	24,50/29,50	C 33
Fight Night/Monsterjagd/Colossus Mühle	24,50/29,50	C 43
Mr. Postman/Moon-tunnel/Moonrallyes!		
Steinschlag	24,50/29,50	C 53
Taxi Driver!		
Deadly Mission	24,50/29,50	C 63

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

C-16

Superpaket 1

Horror Castle, Fantasy County, Senso, Jumping Joe, Schatztaucher, Shaolin, City Runner

SP-O-K-1 Kassette 20,-
SP-O-D-1 Diskette 20,-

Superpaket 2 Genesis-Super-Assembler für C-16/116&plus 4

Komplett mit Anleitung und Demo-Programm

SP-O-K-2 Kassette 15,-
SP-O-D-2 Diskette 15,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Goldrausch/ Moon Fighter	18,-/23,-	O 33
Frogger/Humor	18,-/23,-	O 43
Cosmic Terror/ Schwimmen/Break In	18,-/23,-	O 53
Panic/Phantom/ Meteorsturm	18,-/23,-	O 63

VIC-20

Superpaket 1

Flowers, Tanker, Jupiter Rescue, Einbrecher, Hospital, Alien, Flipperball, Sub-Zone, Dr. Corp, Geo-Zayxon, Froggies Night Mare, Punktefresser, Starwars, Salvage Crew

SP-V-1 nur Kassette 20,-

Programm	Preis Kass.	Bestell-Nr.
Brückenbau/Jango Minor, der Fassaden- streicher/Inkaschatz	16,-	V 13
Bergshooting/ Bobby in Action	16,-	V 23
Josef in den Katakomben	8,50	V 43
Hot Food/ Men-Rescue/ Soundprogramm	14,50	V 53
Castle Dracula/ UFO-Destroyer	14,50	V 63

Schneider

Superpaket 1 Spiele

Alien Hunter, Catch, Interceptor 3 D, City Lander, Conan's Castle, Copter Patrol, CPC Bert, Crash, Demon Attack, Duell, Etis, Labyrinth, Miner, Tower, Snider's Maze, Geister Schloß, Fallschirmspringer, Secret Valley, Aladin, Hounted Castle, Cave runner, Cobra, Erwin, Kruehl, Q-Bert

SP-SR-K-1 Kassette 20,-
SP-SR-D-1 Diskette 27,-

Superpaket 2 Anwender

Back up, Data gen, Datenverwaltung, Discopy, Monitor, Painter, Remkiller, SU-File, Tastaturbelegung

SP-SR-K-2 Kassette 20,-
SP-SR-D-2 Diskette 27,-

Programm	Preis Kass./Disk.	Bestell-Nr.
Schneider Panik/ Killer Ship	18,50/28,50	SR 13
Midnight/Horror Caves	18,50/28,50	SR 23
Spider Maze	18,50/28,50	SR 33
Triton/Traumland	18,50/28,50	SR 43
Starpatrol/ Helicopter Pilot	18,50/28,50	SR 53
Red Alert/ Speedy-Frogs/Aurion	18,50/28,50	SR 63

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

0 56 51/3 00 11

Leser- & Meckerecke

```
5870 SYS 50334,0,0,253,255,253,255,253,255,252,0
,252,1,252,1,251,1
5880 SYS 50674,255:SYS 51699,0
5890 GOTO 1140
5900 REM LEVEL 10
5910 PRINT" (BLACK CLEAR)":SYS 49444,36
5920 SYS 50479,136,136,145,152,137,144,145,152,1
28,135,137,144,128,135,137,144
5930 SYS 50130,7,7,7,7,7,7,7,7
5940 SYS 50203,136,145,137,145,128,137,128,137
5950 SYS 50698,0,0:SYS 50688,0
5960 SYS 49985,50,140,50,0,80,0,110,0,140,0,170,
0,200,0,230,0
5970 SYS 50334,0,0,1,3,1,3,0,252,0,4,252,3,252,2
53,251,1
5980 SYS 50674,255:SYS 51699,0
```

```
<105>
<236>
<87>
<105>
<227>
<229>
<196>
<23>
<147>
<243>
<51>
<81>
5990 GOTO 1140
55000 FOR X=100 TO 0 STEP -10
55010 SYS 49171,X,X,0,0,10,0,21
55020 SYS 49279,X,X,0,0,10,0,21
55030 SYS 49348,X,X,0,0,10,0,21
55040 NEXT X:RETURN
60000 FOR WT=0 TO 40:NEXT WT:RETURN
61000 FORWT=0TO1250:NEXTWT:PRINT"CHOSE DONNS":P
RINTTAB(20)"(SPACE18)"
61010 PRINTTAB(20)"(SPACE19)"
61020 PRINTTAB(20)"(SPACE12)"
61030 PRINTTAB(20)"(SPACE19)"
61040 PRINTTAB(20)"(SPACE19)"
61050 RETURN
ENDE DES LISTINGS
<187>
<191>
<146>
<208>
<201>
<239>
<114>
<226>
<130>
<140>
<150>
<160>
<246>
```

Leser- & Meckerecke

Betrifft: "Compute mit" 12/86

Sehr geehrte Damen und Herren!

Ich beziehe mich auf das in der Ausgabe 12/86 Ihrer Zeitschrift "Compute mit" erschienene Editorial. Darin bezeichnet Ihr Chefredakteur Uwe Knierim die Erscheinung des Schneider PC 1512 als das Mißverständnis des Jahres. dieser Meinung kann ich ganz und gar nicht zustimmen. Ihre Redakteure sollten nicht nur einen flüchtigen Blick auf neuerschienenene Computer werfen, sondern auch "das was dahinter steckt" testen.

Schneider bringt mit diesem IBM-kompatiblen ein wahres Meisterstück heraus, was in dieser Preisklasse und Qualität noch nicht erschienen ist.....

Der PC 15123 verfügt außerdem noch über eine Hardware-Uhr, die durch zwei Batterien gespeist wird. Im Inneren des Pc's befinden sich Steckplätze für RAM-Erweiterungen, einen 8087 Prozessor und drei IBM-kompatible Slots. Die Stromversorgung sowie alle anderen Anschlüsse im Monitor unterzubringen, ist meiner Meinung nach eine vortreffliche Lösung, da der bekannte Kabelsalat somit entfällt.

Alle Vorzüge des neuen Schneider PC's hier aufzuzählen, würde viel Zeit in Anspruch nehmen, bedarf aber auch einer genauen Betrachtung des Computers im Bereich Hard- und Software. Ich möchte hier auf die Zeitschrift "CPC international" 10/11/86 hinweisen, in der ein ausführlicher Test über diesen Computer steht.

Da Sie diesen Brief wahrscheinlich doch nicht abdrucken, bitte ich Sie zumindest, daß Sie sich in

Zukunft besser über Computer informieren und erst dann einen Bericht über das betreffende Gerät schreiben.

Alexander Zöhler, Rednitzhembach

Redaktion:

Doch, wir drucken ihn ab, wenn auch in gekürzter Form.

Da das Editorial in der letzten "Compute Mit" offensichtlich auch auf gegenteilige Meinungen gestoßen ist, möchte ich hier noch einmal kurz auf Ihren Leserbrief eingehen.

Sie erwähnen in Ihrem Schreiben, daß es Ihnen bei dem PC 1512 um die Qualität geht. Ich verstehe auch nach dem tausendsten Blick nicht, was diesen Rechner qualitativ von anderen Kompatiblen, welche zum Teil noch günstiger angeboten werden, unterscheidet. Ich arbeite schon seit über einem Jahr mit einem anderen IBM-Kompatiblen, ohne jemals Schwierigkeiten mit irgendeiner Software zu haben. Auch habe ich echte IBM-Steckplätze, kann jede IBM-Tastatur, jeden beliebigen Monitor anschließen, was Sie sicherlich bald vermissen werden.

Ist es vielleicht der Name Schneider? Bedenken Sie in diesem Zusammenhang, daß auch der Schneider, wie die meisten IBM-Kompatiblen, im fernen Osten produziert wird, in diesem Fall in Korea. Auch die Leistungen des Monitors können hier wohl nicht überzeugen, fast keine Farbe zeigt sich hier ohne Farbbränder, welches sich insbesondere in der GEM-Betriebsart störend bemerkbar macht. Ein Austausch

des Monitors ist leider wegen des eingebauten Netztesiles nur sehr umständlich möglich. Übrigens ich kenne keinen PC, welcher durch Kabelsalat diese Verbindung schafft. Ganz im Gegenteil, spätestens wenn Zusatzkarten mit größerem Strombedarf eingesteckt werden sollen, dürfte beim Schneider PC ein zweites Netzgerät fällig sein.

Es gibt noch zahlreiche weitere Punkte welche davon zeugen, daß der Schneider PC wohl kaum besser ist als andere Kompatible (z.B. sind die Steckplätze in keiner Weise IBM-kompatible, schon die Größe beschränkt die Kompatibilität). Das Argument, daß im Gerätepreis auch reichlich Software enthalten ist, ist sicherlich richtig, darf jedoch in der Bewertung eines Computers keine Rolle spielen. Die richtige Software für seine persönlichen Anforderungen hat noch niemand auf Anhieb gefunden.

Außerdem: sollte die Schneider-Software wirklich so genial sein, zweifel ich nicht daran, daß sie auch separat erhältlich sein wird.

Übrigens: auch andere Zeitungen sind nach ausführlichem Test zu ähnlichen Ergebnissen gekommen. Daß "Schneider International" dies anders sieht, ist Ihr gutes Recht, aber leider auch nur zu gut zu verstehen.

Frank Brall/ Uwe Knierim

"Compute mit" 11/86

An die Redaktion,

im August '86 habe ich meinem Sohn einen CPC 464 zu seinem

14. Geburtstag geschenkt, seitdem beschäftigt er sich sehr intensiv damit. Er tippt auch fleißig Ihre Listings ab, welche aber leider nicht funktionieren.

1. Space-Hawk
Nach mehrmaligem Überprüfen des eingetippten Listings erscheint immer wieder nach dem Start TYPE MISMATCH in Zeile 2740

2. Disc of Tron
eine kurze Erklärung zum Einladen der 3 Listings hätte mit Sicherheit keiner bedauert.

3. Horror Driver
ebenfalls nach Überprüfen des abgetippten Listings bricht das Spiel nach dem Start zusammen.

Bernhard Klein, Heusenstamm

Redaktion

Zu Punkt 1. Sie haben ganz sicher in Ihren Datenzeilen einen unzulässigen Buchstaben eingemogelt. TYPE MISMATCH bedeutet nämlich, daß eine falsche Variablenform vorgefunden wurde. Schauen Sie mal nach, ob Sie nicht zum Beispiel statt einer "Null" irgendwo einen Buchstaben "O" eingemogelt haben.

Punkt 2. Sie haben Recht. Tippen Sie zuerst Teil 1 ein Und speichern ihn. Dann geben Sie Teil 2 ein und starten ihn. Das Programm erzeugt einen File, der hinter Teil 1 abgespeichert werden muß. Teil 3 eintippen und speichern und schließlich Teil 1 laden und starten.

Punkt 3: Sicher haben Sie noch Fehler in den Datenzeilen.

UK

Ice-Racing

Vor den Spaß haben die Götter das Abtippen gestellt. Zunächst müssen die Programmteile "ICE DATA" und "ICE LOADER" abgetippt und auf jeden Fall abgespeichert werden.

Danach startet man den Teil "ICE DATA", wodurch nach ein paar Minuten der Teil "DATA ICE" auf Disk oder Kassette geschrieben wird.

Von nun an muß nur noch der Teil "LOADER ICE" gestartet werden, da dieser den Rest nachlädt. Allerdings sollte, bevor das Programm geladen wird, zuerst der Computer kurz aus- und wieder eingeschaltet werden (RESET reicht nicht aus!), da sonst die Sprites nicht korrekt aussehen. Nachdem das Programm gestartet wurde, erscheint nach einiger Zeit das Titelbild. Sie können jetzt wählen, ob Sie zu zweit (F7) oder gegen den Computer (F1-leicht, F3, F5-schwer) spielen wollen. Die Eislaufschnellbahn wird auf dem Bildschirm zweimal abgebildet, und zwar jeweils aus der Sicht der beiden Spieler. Dabei ist der Einzelspieler immer oben und steuert Joystick 2.

Durch den direkten Vergleich der beiden Läufer sind besonders spannende Rennen möglich. Um Ihren Läufer zu beschleunigen, müssen Sie den Joystick immer in gleiche Richtung drücken wie das vorwärtsbewegende Bein Ihres Läufers. Ob Sie mit Ihrem Läufer schnell oder langsam sind, können Sie anhand des Geschwindigkeitsbalken unterhalb Ihres Bildausschnittes sehen.

Sobald beide Läufer im Ziel sind, werden die Zeiten angezeigt und Sie können ein neues Spiel beginnen.



U-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 GOTO10
1 *****
2 *
3 * ICE RACING (C) BY *
4 *
5 * RADDISH-SOFT GERMANY *
6 *
7 * AUTOR : CHRISTIAN ROUCH *
8 *
9 *****
10 IFA=OTHENA=1:LOAD"DATA(SPACE)ICE",8,1:REM BEI
    KASSETTE ,1,1
11 *****
12 PRINT" (GREY CLEAR DOWN3 RIGHT3)PLEASE(SPACE)
    WAIT(SPACE)1(SPACE)MINUTE(SPACE)"
13 *****
14 PRINT" (DOWN2 RIGHT3)ICE(SPACE)RACING(SPACE)BY
    (SPACE)RADDISH(SPACE)SOFT(SPACE)GERMANY":FORI=1T
    O4000:NEXT
15 DIMW(1519):FORI=0TO1515:POKE50176+I,PEEK(2698
    4+I):NEXT
16 POKE56576,PEEK(56576)AND252:POKE53272,8:POKE
    648,192:PRINT" (BROWN)"
17 POKE53270,216
18 FORX=832TO865:READQ:POKEX,Q:NEXT
19 DATA120,169,51,133,1,169,0,133,95,133,90,133
    ,88,169,208,133,96,169,240,133
20 DATA89,169,224,133,91,32,191,163,169,55,133,
    1,88,96
21 SYSB32
22 BA=57344
23 READZE:IFZE=-1THEN1000
24 FORI=0TO7:READQ:POKEBA+ZE*8+I,Q:NEXT
25 GOTO230
500 DATA33,,63,234,234,234,234,234,63,34,,252,3,

```

```

3,3,3,3,252,35,,255,,,,,255
510 DATA36,,255,128,128,128,128,255,37,,255,
160,160,160,160,160,255
520 DATA38,,255,168,168,168,168,168,255,39,,255,
170,170,170,170,170,255
600 DATA40,,,,,20,85,41,170,40,,,,,42,,,,,1
,17,85,43,,,,,64,80,84
610 DATA44,170,170,34,,,,,45,168,160,128,,,,,4
6,,,,,84,47,168,,,,,
620 DATA58,,,,,21,85,85,59,,,,,64,80,84,60,170
,42,10,,,,,61,168,168,160,,,,
630 DATA62,,,,,1,5,21,85,63,,,,,64,80,85,85,64,,
,,,,,64,65,170,42,10,,,,,66
640 DATA17,170,168,,,,,67,128,,,,,68,,,,,1,5,
21,21,85,69,170,42,10,2,,,,
650 DATA70,,,,,1,7,21,23,87,71,,,,,85,223,221,221,
223,72,170,170,170,170,,,,
660 DATA73,,,,,85,223,221,93,223,74,,,,,80,212,213
,213,213,75,,,,,247,215,247,215
670 DATA247,76,,,,,119,247,247,247,119,77,,,,,208,
116,117,117,213
700 DATA78,,,,,2,79,,,,,42,170,170,80,,,,,1
70,170,170,81,,,,,168,170,170
710 DATAB2,,,,,128,83,2,10,10,10,42,42,42,42,
84,170,170,170,170,154,86,86,86
720 DATAB5,170,170,170,170,170,170,170,86,17
0,170,170,170,170,150,85,85
730 DATAB7,170,170,170,170,170,90,86,86,88,128,1
60,160,160,168,168,168,168
740 DATAB9,170,170,170,170,170,169,169,169,90,86
,86,86,86,86,90,90,90
750 DATA91,170,170,170,165,149,149,86,90,92,170,
170,170,106,90,90,154,170
760 DATA93,170,170,170,165,149,150,90,90,94,170,
170,170,106,90,90,154,154

```


programme

```

770 DATA95,170,170,170,170,170,170,170,165,96,17
0,170,170,170,170,170,86
780 DATA97,170,170,170,170,170,169,169,169,98,85
,86,86,85,85,85,85,85
790 DATA99,149,165,165,149,86,86,90,170,100,170,
170,170,169,165,149,150,154
800 DATA101,170,170,170,90,86,86,150,150,102,170
,170,170,169,165,165,149,150
810 DATA103,170,170,170,90,86,86,166,170,104,166
,149,149,170,166,149,149,149
820 DATA105,170,170,170,169,165,165,165,165,106,
170,170,170,150,85,85,101,165
830 DATA107,170,170,170,170,169,169,165,165,108,
170,170,170,89,85,85,85,165
840 DATA109,169,169,169,169,169,41,42,42,110,90,
90,90,90,90,106,170
850 DATA111,90,90,90,86,149,149,165,170,112,170,
170,154,90,90,106,170,170
860 DATA113,85,90,90,86,149,149,165,170,114,106,
170,154,90,90,106,170,170
870 DATA115,149,149,165,170,170,170,170,170,116,
85,85,86,170,170,170,170
880 DATA117,169,169,169,169,169,169,169,170,170,118,
85,89,89,90,90,90,106,170
890 DATA119,106,106,90,90,150,150,170,120,169
,165,149,150,150,149,165,170
900 DATA121,86,86,150,150,86,86,150,170,122,150,
150,150,149,165,165,169,170
910 DATA123,170,170,166,150,86,90,106,170,124,14
9,149,149,149,149,149,166,170
920 DATA125,165,165,165,165,165,165,166,170,126,
165,165,165,165,165,166,170
930 DATA127,165,165,165,165,169,169,170,170,128,
165,165,165,85,85,85,85,165
940 DATA129,170,170,170,170,170,168,168,168,130,
42,10,10,10,2,2,,
950 DATA131,170,170,170,170,170,170,170,42,132,1
66,165,165,169,169,170,170,170
960 DATA133,165,149,85,86,86,90,170,168,134,168,
160,160,160,128,128,0,0
999 DATA-1
1000 FOR T=1 TO 1519:READ W(T):NEXT
1003 SI=54272:L=SI+24
1006 W1=SI+4:W2=SI+11:W3=SI+18
1009 A1=SI+5:A2=SI+12:A3=SI+19
1012 H1=SI+6:H2=SI+13:H3=SI+20
1015 T3=SI+16:P3=SI+17
1018 POKE L,15
1021 POKE A1,74 :POKE A2,95 :POKE A3,95
1024 POKE H1,60 :POKE H2,191:POKE H3,191
1027 POKE T3,15 :POKE P3,15:SYB 30000
1030 FOR T=1 TO 1519
1033 POKE 54273,W(T):T=T+1:POKE 54272,W(T):T=T+1
1036 POKE 54280,W(T):T=T+1:POKE 54279,W(T):T=T+1
1039 POKE 54287,W(T):T=T+1:POKE 54286,W(T):T=T+1
1042 POKE W1,33:POKE W2,33:POKE W3,65
1045 FOR I=0 TO W(T)-10:NEXT
1048 POKE W1,32:POKE W2,32:POKE W3,32
1051 NEXT T
1054 GOTO 1030
1057 REM BEGIN 1:E-DUR:TON E;-----
1060 DATA 5,123,0,0,0,0,50
1063 DATA 5,123,0,0,0,0,50
1066 DATA 5,123,0,0,0,0,50
1069 DATA 5,123,0,0,0,0,50
1072 DATA 5,123,0,0,0,0,50
1075 DATA 5,123,0,0,0,0,50
1078 DATA 5,123,0,0,0,0,50
1081 DATA 5,123,0,0,0,0,50
1084 DATA 5,123,0,0,0,0,50
1087 DATA 5,123,0,0,0,0,50
1090 DATA 5,123,0,0,0,0,50
1093 DATA 5,123,0,0,0,0,50
1096 DATA 5,123,0,0,0,0,50
1099 DATA 5,123,0,0,0,0,50
1102 DATA 5,123,0,0,0,0,50
1105 DATA 5,123,0,0,0,0,50
1108 DATA 5,123,0,0,5,123,50
1111 REM TON E-E FOLGE:VERZ.-----
1114 DATA 10,247,0,0,10,247,50
1117 DATA 5,123,0,0,5,123,50
1120 DATA 10,247,0,0,10,247,50
1123 DATA 5,123,0,0,5,123,50
1126 DATA 10,247,0,0,10,247,50
1129 DATA 5,123,0,0,5,123,50
1132 DATA 10,247,0,0,10,247,50

```

```

<222> 1135 DATA 5,123,0,0,5,123,50 <146>
1138 DATA 10,247,0,0,10,247,50 <177>
1141 DATA 5,123,0,0,5,123,50 <152>
<32> 1144 DATA 10,247,0,0,10,247,50 <183>
1147 DATA 5,123,0,0,5,123,50 <158>
<180> 1150 DATA 10,247,0,0,10,247,50 <189>
1153 DATA 5,123,0,0,5,123,50 <164>
<0> 1156 DATA 10,247,0,0,10,247,50 <195>
1159 REM TON E-E FOLGE:VERZ.----- <20>
<163> 1162 DATA 5,123,5,123,52,39,50 <88>
1165 DATA 10,247,10,247,49,58,50 <194>
<40> 1168 DATA 5,123,5,123,43,219,50 <138>
1171 DATA 10,247,10,247,32,219,50 <186>
<151> 1174 DATA 5,123,5,123,5,123,50 <221>
1177 DATA 10,247,10,247,10,247,50 <144>
<65> 1180 DATA 5,123,5,123,5,123,50 <227>
1183 DATA 10,247,10,247,10,247,50 <150>
<236> 1186 DATA 5,123,5,123,5,123,50 <233>
1189 DATA 10,247,10,247,10,247,50 <156>
<66> 1192 DATA 5,123,5,123,5,123,50 <239>
1195 DATA 10,247,10,247,10,247,50 <162>
<195> 1198 DATA 5,123,5,123,5,123,50 <245>
1201 DATA 10,247,10,247,10,247,50 <168>
1204 DATA 5,123,5,123,5,123,50 <251>
1207 DATA 10,247,10,247,10,247,50 <174>
<193> 1210 REM ANF.1:ST1:BGL.ST2,3----- <27>
1213 DATA 5,123,87,182,52,39,50 <75>
<76> 1216 DATA 10,247,0,0,49,58,50 <169>
1219 DATA 5,123,0,0,43,219,50 <41>
<148> 1222 DATA 10,247,0,0,32,219,50 <44>
1225 DATA 5,123,87,182,52,39,50 <87>
<140> 1228 DATA 10,247,0,0,49,58,50 <181>
1231 DATA 4,226,0,0,43,219,50 <73>
<136> 1234 DATA 9,196,0,0,32,219,50 <92>
1237 DATA 4,90,69,157,52,39,50 <68>
<150> 1240 DATA 8,180,0,0,49,58,50 <225>
1243 DATA 4,90,0,0,43,219,50 <127>
1246 DATA 8,180,0,0,34,207,50 <37>
1249 DATA 4,90,69,157,52,39,50 <80>
<165> 1252 DATA 8,180,0,0,49,58,50 <237>
<90> 1255 DATA 4,90,0,0,43,219,50 <139>
<178> 1258 DATA 8,180,0,0,34,207,50 <49>
<185> 1261 REM ----- <202>
<66> 1264 DATA 4,226,0,0,39,18,50 <179>
<13> 1267 DATA 9,196,0,0,39,18,50 <223>
<34> 1270 DATA 4,226,0,0,39,18,50 <185>
<27> 1273 DATA 9,196,0,0,39,18,50 <229>
<125> 1276 DATA 4,226,0,0,58,138,50 <195>
<98> 1279 DATA 9,196,0,0,52,39,50 <214>
<88> 1282 DATA 4,226,0,0,49,58,50 <14>
<61> 1285 DATA 5,123,43,219,0,0,50 <8>
<99> 1288 DATA 10,247,10,247,43,219,50 <81>
<187> 1291 DATA 5,123,5,123,43,219,50 <6>
<89> 1294 DATA 10,247,10,247,43,219,50 <87>
<85> 1297 DATA 5,123,5,123,0,0,50 <249>
<21> 1300 DATA 10,247,10,247,0,0,50 <4>
<128> 1303 DATA 5,123,5,123,0,0,50 <255>
<147> 1306 DATA 10,247,10,247,0,0,50 <10>
<73> 1309 DATA 5,123,5,123,0,0,50 <5>
<89> 1312 DATA 10,247,10,247,0,0,50 <16>
<95> 1315 DATA 5,123,5,123,0,0,50 <11>
<189> 1318 DATA 10,247,10,247,0,0,50 <22>
<192> 1321 DATA 5,123,5,123,0,0,50 <17>
<195> 1324 DATA 10,247,10,247,0,0,50 <28>
<198> 1327 DATA 5,123,5,123,0,0,50 <23>
<201> 1330 DATA 10,247,10,247,0,0,50 <34>
<204> 1333 REM TEIL 2;W-BEGIN 1:TON E;----- <149>
<207> 1336 DATA 5,123,5,123,52,39,50 <7>
<210> 1339 DATA 10,247,10,247,49,58,50 <113>
<213> 1342 DATA 5,123,5,123,43,219,50 <57>
<216> 1345 DATA 10,247,10,247,32,219,50 <105>
<219> 1348 DATA 5,123,5,123,5,123,50 <140>
<222> 1351 DATA 10,247,10,247,10,247,50 <63>
<225> 1354 DATA 5,123,5,123,5,123,50 <146>
<228> 1357 DATA 10,247,10,247,10,247,50 <69>
<231> 1360 DATA 5,123,5,123,5,123,50 <152>
<234> 1363 DATA 10,247,10,247,10,247,50 <75>
<119> 1366 DATA 5,123,5,123,5,123,50 <158>
<241> 1369 DATA 10,247,10,247,10,247,50 <81>
<153> 1372 DATA 5,123,5,123,5,123,50 <164>
<128> 1375 DATA 10,247,10,247,10,247,50 <87>
<159> 1378 DATA 5,123,5,123,5,123,50 <170>
<134> 1381 DATA 10,247,10,247,10,247,50 <93>
<165> 1384 DATA 5,123,87,182,52,39,50 <247>
<140> 1387 DATA 10,247,0,0,49,58,50 <85>
<171> 1390 DATA 5,123,0,0,43,219,50 <213>

```


1103 DATA F7ADF27FC9D290034C8B76CE7B7F3BA919EDB4
7F0A0A08DF07F18ADF17F6D, 4187 <135>

1104 DATA F078BDF17F9003CE7B7FAD7B7FC9D090034C31
76A9D78D7B7FEF27FAEF27F, 4626 <186>

1105 DATA A028B03B73991FC3BD35749947C3CABBDOF0EA
4CB76ADCA7F8DFCC3ADCB7F, 4710 <50>

1106 DATA 8DFDC3ADCC7F8DFEC3ADCC7F8DFCC3A602E003
D06520647720E377A011B9CD, 4679 <84>

1107 DATA 7F99FFCF8BDOF7ADF57FC9D290034C8B76A906
8D25D0CE7D7F3BA919EDB07F, 4575 <188>

1108 DATA 0A0A08DF37F18ADF47F6DF37F8DF47F9003CE
7D7FAD7D7FC9D090034C8B76, 4064 <177>

1109 DATA A9D78D7D7FEF57FAEF57FA028BD3B739967C1
BD3574998FC1CABBDOF0A602, 4746 <152>

1110 DATA BD7A7F8D16D0BD27F8D12D08AF0066B8B68AA
684020F076202A77205F7920, 3567 <126>

1111 DATA 467B20AE7B20167C20617C20F77C20AC7C2019
7D4C31EAADF57FC9D2900160, 3422 <64>

1112 DATA CE7E7FF00160AC7F7F89B07F8D7E7FB98A7F8D
CA7FB9947F8DCB7F899E7F8D, 4518 <96>

1113 DATA CC7FB987F8DCD7FC8C00AD002A0008C7F7F60
ADF27FC9D2900160CEB27FF0, 4491 <142>

1114 DATA 0160ACB37FB9847F8DB27FB98A7F8DBE7FB994
7FBDBF7FB99E7F8DC07FB98, 4623 <62>

1115 DATA 7FBDC17FCBC00AD002A0008C37F60A9008DF6
7FBDF77F8DF97F8DFA7F8D5, 4553 <180>

1116 DATA 7FBDF67FA00B18ADF67F6DF57FBDF67FADF77F
69008DF77F8800EB38A9D7ED, 4797 <64>

1117 DATA 7D7F8DF87F18ADF67F6DF87F8DF67FADF77F69
008DF77FADF27F8DF97FA008, 4730 <125>

1118 DATA 18ADF97F6DF27FBDF97FADF7F69008DFA7F8B
D0EB38A9D7ED7B7F8DF87F18, 4790 <211>

1119 DATA ADF97F6DF87F8DF97FADF7F69008DFA7F603B
ADF67FEDF97F8DFC7FB003EE, 4974 <50>

1120 DATA FA7F38ADF77FEDFA7F8DFD7FADF7FC9FFF005
C900F06460ADF7FC9979042, 5120 <31>

1121 DATA C9AF901D38A900EDFC7F6DD67FBDC7F8DD27F
6918DD07F8DD47FA900BDEE, 4409 <55>

1122 DATA 7F60A90ABDD7F38A9AEEDFC7F8DD07F8DD47F
38A900EDFC7F6DD67FBDC7F, 4628 <142>

1123 DATA 8DD27F60A90F8DD7F38A996EDFC7F8DCE7F8D
D27F6917BDD07F8DD47F60AD, 4544 <87>

1124 DATA FC7FC9BF90034CBF78ADD67FEDFC7F8DCE7F8D
D27F6917BDD07F8DD47FA900, 4544 <218>

1125 DATA 8DDE7F60A9008DDE7F8DD07F8DD47F8DD27F8D
CE7F6038ADF97FEDF67F8DFC, 4857 <121>

1126 DATA 7FB003EEF77F38ADF7FEDF77F8DFD7FADF7F
C9FFF005C900F06460ADF7C7F, 5099 <242>

1127 DATA C9979042C9AF901D38A900EDFC7F6DDF7F8DE7
7FBDEB7F6918BDE97F8DE07F, 4548 <158>

1128 DATA A9008DEF7F60A9008DEF7F38A9AEEDFC7F8DE9
7FBDE7F38A900EDFC7F6DDF, 4802 <255>

1129 DATA 7FBDE77F8DEB7F60A90F08DEF7F38A996EDFC7F
8DE77FBDEB7F6917BDE97F8D, 4904 <202>

1130 DATA ED7F60ADF7FC9BF90034C4D79ADD7FEDFC7F
8DE77FBDEB7F6917BDE97F8D, 4688 <217>

1131 DATA ED7FA9008DEF7F60A9008DEF7F8DE97F8DE07F
8DEB7F8DE77F60AD807FC914, 4565 <33>

1132 DATA B016C911B037C90EB058C90B079C90B80034C
357A4C107AA90C8DB17F8DB4, 3339 <152>

1133 DATA 7FBDB67F8DB97FA9068DB27F8DB37F8DB77F8D
8B7FAD807F8DB57F4C577AA9, 3996 <61>

1134 DATA 0A8DB17F8DB47F8DB67F8DB97FA9058DB27F8D
B37F8DB77F8DB87FAD807F8D, 3961 <129>

1135 DATA 857F4C577AA9088DB17F8DB47F8DB67F8DB97F
A9048DB27F8DB37F8DB77F8D, 3840 <225>

1136 DATA 8B7FAD807F8DB57F4C577AA9068DB17F8DB47F
8DB67F8DB97FA9038DB27F8D, 3867 <213>

1137 DATA 837F8DB77F8DB87FAD807F8DB57F4C577AA905
8DB17F8DB47F8DB67F8DB97F, 3957 <58>

1138 DATA A9028DB27F8DB37F8DB77F8DB87FAD807F8DB5
7F4C577AA9048DB17F8DB47F, 3859 <126>

1139 DATA 8D867F8DB97FA9028DB27F8DB37F8DB77F8DB8
7FAD807F8DB57FAD847FC914, 4112 <87>

1140 DATA B016C911B035C90EB054C90B073C90B80034C
237B4C007BA90C8DB57F8DB8, 3399 <70>

1141 DATA 7FBDBA7F8DBD7FA9068DB67F8DB77F8DB87F8D
BC7FAD847F8DB97F60A90ABD, 4374 <18>

1142 DATA B57F8DB87F8DBA7F8DBD7FA9058DB67F8DB77F
8DBB7F8DB87FAD847F8DB97F, 4590 <45>

1143 DATA 60A908DB57F8DB87F8DBA7F8DBD7FA9048DB6
7F8DB77F8DB87FAD847FAD84, 4423 <230>

1144 DATA 7FBDB97F60A9068DB57F8DB87F8DBA7F8DBD7F
A9038DB67F8DB77F8DB87F8D, 4332 <237>

1145 DATA BC7FAD847F8DB97F60A9058DB57F8DB87F8DBA
7FBDBD7FA9028DB67F8DB77F, 4402 <38>

1146 DATA 8DBB7F8DB87FAD847F8DB97F60A9048DB57F8D
B87F8DBA7F8DBD7FA9028DB6, 4419 <252>

1147 DATA 7FBDB77F8DB87F8DB97FAD847F8DB97F60ADF5
7FC9D2900160AD01DC2904F0, 4394 <142>

1148 DATA 0AAD01DC2908F00D4C7E7BAD7F7FC905B00D4C
7E7BAD7F7FC90590034C7E7B, 3283 <61>

1149 DATA EEF7E7ADF7FC914F00F60CEFE7CFE7E7ADF7E
7FC902901260A9028DFE7FAD, 4800 <4>

1150 DATA B07FC905B00160CEB07F60A913BDFE7FAD807F
C918900160E0EB7F60ADF27F, 4026 <45>

1151 DATA C9D2900160AD00DC2904F00AAD00DC2908F00D
4CE67BADB37FC905B00D4CE6, 3644 <163>

1152 DATA 7BADB37FC90590034CE67BEEFF7FADFF7FC914
F00F60CEFF7CEFF7FADFF7F, 4862 <108>

1153 DATA C902901260A9028DF7FAD847FC905B00160CE
B47F60A913BDF7FAD847FC9, 4115 <234>

1154 DATA 18900160EEB47F60A923BDF0C1A000AEB07FEB
E01EB01618B9F0C1690199F0, 4098 <202>

1155 DATA C1C927D0EECB9A92399FC14C207CA9238DABC3
A000AEB47FE8E01EB01618B9, 4343 <204>

1156 DATA ABC3690199ABC3C927D0EECB9A92399ABC34C45
7C60ADF57FC9D2900160EE62, 4489 <26>

1157 DATA C0EE62C0AD62C0C93AB001603BAD62C0E90ABD
62C0EE61C0AD61C0C93AB001, 4333 <207>

1158 DATA 603BAD61C0E90ABD61C0EE5FC0AD5FC0C93AB0
01603BAD5FC0E90ABD5FC0EE, 4229 <110>

1159 DATA 5EC060ADF27FC9D2900160EE42C2EE42C2AD42
C2C93AB001603BAD42C2E90A, 4269 <242>

1160 DATA 8D42C2EE41C2AD41C2C93AB001603BAD41C2E9
0ABD41C2EE3FC2AD3FC2C93A, 4177 <165>

1161 DATA B001603BAD3FC2E90ABD3FC2EE3EC260AD06B0
C901F009C902F00BC903F00D, 3659 <77>

1162 DATA 60A9148DB07F60A908DB07F60A908DB07F60
ADF57FC9D2800160ADF27FC9, 4092 <65>

1163 DATA D2B00160EAAD5EC08D0A80AD5FC08D0B80AD61
C08D0C80AD62C08D0B80AD3E, 3918 <93>

1164 DATA C28D0E80AD3FC28D0F80AD41C28D10B0AD42C2
8D11B020167EA98C8D01D08D, 3607 <9>

1165 DATA 03D0A9A18D05D08D07D0A97F8D09D08D08D0A9
B48D0D08D0F0A96E8D08D0, 4008 <224>

1166 DATA 8D0C0A98B8D0AD08D0E0A9108DF8C3A9118D
F7C3A9128DFAC3A9138DF8C3, 4474 <20>

1167 DATA A9208DFCC3A9218DFDC3A9228DFEC3A9238DFF
C3A9098D25D0A90E8D20D0AD, 4469 <117>

1168 DATA 0E808DB7C3AD0F808DB8C3AD10808DBAC3AD11
B08DB8C3AD0A808D67C3AD08, 4111 <5>

1169 DATA 80BD68C3AD0C808D6AC3AD0808D6BC34CBA7E
A0FAA92099FFBF79F9C099F3, 4668 <145>

1170 DATA C199EDC2A90B99FFD799F9D899F3D999EDDAA9
208BD0E160A0FAB9536F99FF, 5482 <42>

1171 DATA BF894D7099F9C0B9477199F3C1B9417299EDC2
A90B99FFD799F9D899F3D9A9, 5268 <114>

1172 DATA 0099EDDAB8D0D560A930B3EC28D3FC28D41C2
8D42C28D5EC08D5FC08D61C0, 4359 <230>

1173 DATA 8D62C0A9218DFEFC18DA7C3A9238DABC38DA9C3
8DAAC38DABC38DACC38DADC3, 4952 <21>

1174 DATA 8DF0C18DF0C18DF1C18DF2C18DF3C18DF4C18D
F5C1A9228DF7C18DAFC3A923, 5532 <31>

1175 DATA 8DF6C18DAEC3A9008D62D88D61D88D5F88D5E
D88D42D88D41D88D3F88D3E, 4545 <35>

1176 DATA DA6078A9CE8D1403A97EBD1503A9008D12D085
028860AD19D08D19D02901D0, 3318 <7>

1177 DATA 034CFA7EC6021004A9028502A602BD03808D21
D08D07B08D23D08D00808D12, 3035 <172>

1178 DATA D0BAF006688B8A684020067F4C31EAA5C8C9
04F00DC905F011C906F015C9, 3788 <29>

1179 DATA 03F01960A9018D06804C397FA9028D06B04C39
7FA9038D06804C397FA9008D, 2792 <73>

1180 DATA 06804C397FA9288DF57F8DF27FA9178D807F8D
B47F20F27D20447EA9068D25, 3640 <201>

1181 DATA D0A908BD23D0A9008D20D078A998BD1403A975
BD1503A9008D12D085028860, 3233 <23>

1182 DATA 3EE698BE00050007D0D0D000101060C06060C
180C06060C1014181C202420, 1722 <185>

1183 DATA C18141115191D2125211D191512161A1E2226
221E1A161317181F2327231F, 862 <122>

1184 DATA 1B1701010A0C06060C180C06060C0101010101
010101010101010101010176, 324 <12>

1185 DATA 69AE69967EAE7E9655AE55966AAE6A0096C1AE
C176D6AED696AEDA96C2AE, 4640 <46>

1186 DATA C20000002000000000000000000000000101B4
7000020007000D08000000, 954 <239>

1187 DATA 0000000000FF, 255 <91>

ENDE DES LISTINGS

ASSEMBLERKURS

Teil 20

Eines der interessantesten Programmiergebiete dürften beim Commodore 64 eigene Basic-Befehlsweiterungen darstellen. Da der C64 von "Natur" aus mit einem recht mageren Befehlssatz ausgestattet ist, insbesondere im Bereich Grafik und der Sprite-Animation, bietet sich eine Befehlsweiterung geradezu an. Die grundlegenden Eigenschaften, vor allem die Parameterübergabe, wurden im letzten Teil unseres Kurses schon erläutert. Wir wollen uns nun etwas näher mit den Grafikeigenschaften insbesondere mit dem VIDEOCHIP des C64 auseinandersetzen um nachher eine kleine SPRITE-Befehlsweiterung programmieren zu können.

Der VIDEOCONTROLLER VIC 6567 des Commodore 64

Im Gegensatz zu früheren Computer-Systemen, besitzt der Commodore 64 zur Unterstützung der Bildschirm-Verwaltung einen speziellen Prozessor, den sogenannten Videocontroller. Dieser Videocontroller, kurz VIC genannt, sorgt für die Aufbereitung aller Videosignale, wie auch für die flexiblen Grafikeigenschaften des Commodore 64. In der Praxis bedeutet dies, daß die Darstellung von Sprites vollkommen vom VIC übernommen wird. Flackern oder Geschwindigkeitprobleme wie sie auf früheren Computersystemen üblich waren, fallen beim C64 ganz weg.

Wie läßt sich der VIC beeinflussen ?

Um Sprites bewegen, oder Zeichen definieren zu können, muß der VIC-Chip beeinflusst werden. Um dies zu erreichen, belegt der VIC-Chip einen bestimmten Adressbereich des Rechners. In einem unserer letzten Teile haben wir schon mehrmals den I/O Eingabe/Ausgabebereich erwähnt. Ein kleiner Teil dieses Bereiches (47Byte) wird vom VIC genutzt. Man nennt die 47 Byte, welche den VIC beeinflussen bzw. steuern auch Register. Numeriert man die Register der Reihe nach durch, so liegt Register 0 auf der Speicherstelle 53248 (\$D000) und Register 46 auf Speicherstelle 53294 (\$D02E). Um diese Register zu lesen bzw. beschreiben zu können, muß natürlich der I/O Bereich aktiviert sein, der Zeichengenerator oder das darunterliegende RAM kann also nicht gleichzeitig manipuliert werden.

Die Register des VIC

Die nachfolgende Tabelle zeigt die Adresse sowie die Bedeutung jeder einzelnen Register auf. Übrigens ist in der Standard Konfiguration nach dem Einschalten des Computers der I/O Bereich aktiviert. Dadurch können die VIC-Register auch von Basic mittels POKE erreicht werden.

Register Adresse Funktion

Register	Adresse	Funktion
0	53248	Sprite 0 X-Koordinate (0-255) LOW Byte
1	53249	Sprite 0 Y-Koordinate (0-255)
2	53250	Sprite 1 X-Koordinate (0-255) LOW Byte
3	53251	Sprite 1 Y-Koordinate (0-255)
4	53252	Sprite 2 X-Koordinate (0-255) LOW Byte
5	53253	Sprite 2 Y-Koordinate (0-255)

6	53254	Sprite 3 X-Koordinate (0-255) LOW Byte
7	53255	Sprite 3 Y-Koordinate (0-255)
8	53256	Sprite 4 X-Koordinate (0-255) LOW Byte
9	53257	Sprite 4 Y-Koordinate (0-255)
10	53258	Sprite 5 X-Koordinate (0-255) LOW Byte
11	53259	Sprite 5 Y-Koordinate (0-255)
12	53260	Sprite 6 X-Koordinate (0-255) LOW Byte
13	53261	Sprite 6 Y-Koordinate (0-255)
14	53262	Sprite 7 X-Koordinate (0-255) LOW Byte
15	53263	Sprite 7 Y-Koordinate (0-255)
16	53264	X-Koordinate High Bit:

Bit 0 = Sprite 0

Bit 1 = Sprite 1

Bit 2 = Sprite 2

Bit 3 = Sprite 3

Bit 4 = Sprite 4

Bit 5 = Sprite 5

Bit 6 = Sprite 6

Bit 7 = Sprite 7

17	53265	Verschiedene Funktionen: Bit 0-2 Y-Position des Screen (für SOFT-SCROLL) Bit 3 Umschaltung zwischen 24 u. 25 Zeilen (1) Bit 4 Bildschirm EIN (1) Bit 5 HI-RES Grafik (1) Bit 6 Hintergrund Mehrfarbig (1) Bit 7 Rasterwertregister High Byte
----	-------	---

18	53266	Rasterwertregister (Bits 0-7)
----	-------	-------------------------------

19	53267	Light-Pen X-Koordinate
----	-------	------------------------

20	53268	Light-Pen Y-Koordinate
----	-------	------------------------

21	53269	Sprite Ein/AUS:
----	-------	-----------------

Bit 0 Sprite 0 ein

Bit 1 Sprite 1 ein

Bit 2 Sprite 2 ein

Bit 3 Sprite 3 ein

Bit 4 Sprite 4 ein

Bit 5 Sprite 5 ein

Bit 6 Sprite 6 ein

Bit 7 Sprite 7 ein

22	53270	Verschiedene Funktionen: Bit 0-2 X-Position Screen (SOFT-SCROLL) Bit 3 Umschaltung zwischen 40 und 38 Spalten Bit 4 Mehrfarbenmodus Bit 5-7 nicht benutzt
----	-------	---

23	53271	Sprite 0-7 Y-Vergrößerung
----	-------	---------------------------

24	53272	Bit 0 nicht benutzt
----	-------	---------------------

Bit 1-3 Adresse des Zeichengenerators bzw.

Bitmap

Bit 4-7 Adresse des Bildschirmspeichers

25	53273	Unterbrechungs-Flag-Register: Bit 0 Rasterinterrupt Bit 1 Kollision Sprite-Hintergrund Bit 2 Kollision Sprite-Sprite Bit 3 Interrupt Light Pen Bit 4-6 nicht benutzt Bit 7 eines der Bits 0-3 gesetzt
26	53274	Unterbrechungs-Maskenregister: Bit 0 Rastervergleich ein Bit 1 Kollision Sprite-Hintergrund Bit 2 Kollision Sprite-Sprite Bit 3 Unterbrechung durch Light-Pen Bit 4-7 nicht benutzt
27	53275	Sprite 0-7 Hintergrund Priorität
28	53276	Sprite 0-7 Mehrfarbig
29	53277	Sprite 0-7 X-Vergrößerung
30	53278	Sprite 0-7 Sprite-Sprite Kollision anzeigen
31	53279	Sprite 0-7 Sprite-Hintergrund Kollision anzeigen
32	53280	Randfarbe (0-15)
33	53281	Hintergrundfarbe 1 (0-15)
34	53282	Hintergrundfarbe 2 (0-15)
35	53283	Hintergrundfarbe 3 (0-15)
36	53284	Hintergrundfarbe 4 (0-15)
37	53285	Sprite-Vielfarbenregister 0 (0-15)
38	53286	Sprite-Vielfarbenregister 1 (0-15)
39	53287	Sprite 0 Farbbregister (0-15)
40	53288	Sprite 1 Farbbregister (0-15)
41	53289	Sprite 2 Farbbregister (0-15)
42	53290	Sprite 3 Farbbregister (0-15)
43	53291	Sprite 4 Farbbregister (0-15)
44	53292	Sprite 5 Farbbregister (0-15)
45	53293	Sprite 6 Farbbregister (0-15)
46	53294	Sprite 7 Farbbregister (0-15)

Wie werden Sprites im Speicher abgelegt ?

Wie schon aus der obigen Tabelle zu entnehmen ist, wurde bei der Entwicklung des VIC besonders viel Wert auf die Sprites (kleine Figuren) gelegt. Diese Sprites lassen sich vorzüglich für Action-Spiele verwenden, da sie sehr schnell arbeiten und automatisch auf Kollisionen reagieren. Dies gilt jedenfalls für Maschinenprogramme, in Basic dagegen stehen dem C64 Programmierer nur die lästigen POKE-Befehle zur Spritesteuerung zur Verfügung. Um nun in unserem Assemblerkurs gleich zwei Fliegen mit einer Klappe zu schlagen, entwickeln wir einige Maschinenroutinen, welche vom Basic aus die Handhabung der Sprites etwas erleichtern.

Um eine Spriteerweiterung schreiben zu können, benötigen wir genaue Kenntnisse über deren Aufbau. Da die Informationen des Commodore-Handbuchs teilweise etwas unklar sowie auch unvollständig sind, möchte ich an dieser Stelle noch einmal etwas genauer auf dieses Thema eingehen.

Sprites können beim C64 in zwei verschiedenen Auflösungen dargestellt werden - HIRES oder Multicolor. Hiressprites erlauben beim C64 eine Auflösung von 24 * 21 Pixel, allerdings nur in einer Farbe. Multicolorsprites dagegen können in 4 verschiedenen Farben (incl. Hintergrundfarbe) dargestellt werden, haben jedoch nur halbe Auflösung.

Wir wollen uns an dieser Stelle nicht mit dem Aufbau des Bitmapping beschäftigen, da dieses mittlerweile in jedem zweiten Artikel über den C64 wiederholt wird. Mittlerweile gibt es auch zahlreiche Programme, sogenannte Spritedesigner, welche das Definieren von Multicolor- oder Hiressprites stark vereinfachen. Wer noch keinen Spritedesigner besitzt, kann übrigens das Basic-Listing am Ende dieses Teiles eingeben. Dieses kurze Programm erlaubt das einfache Erstellen von Hiressprites. Die Spritedaten werden automatisch in Data-Zeilen gewandelt und im Speicher abgelegt.

Egal ob wir ein Multicolor- oder ein Hiressprite definiert haben, in beiden Fällen belegen die Daten (Bitmap) genau 64 Bytes. Nun, wie teile ich dem Videocollimator mit, wo die Daten von welchem Sprite liegen?

Diese Frage ist berechtigt, denn bisher haben wir im Videocontroller noch kein Register gefunden, welches diese Aufgabe übernehmen könnte. Hierfür sind 8 Speicherstellen zuständig, welche sich 17 Bytes hinter dem Videoscreenende beginnen. Da der Videoscreen nach dem Einschalten im Bereich 1024 bis 2023 liegt, beginnen die 8 Spritzezeiger ab der Adresse 2040. Verlegt man den Videocontroller oder nur den Videoscreen so werden auch die Spritzezeiger verlegt. Da ein Spritpointer nur aus einem Byte besteht, kann er verständlicherweise nicht die vollständige Spriteadresse errechnen, sich indem man diesen Block mit der Zahl 64 multipliziert. Auf diese Weise können Sprites beliebig in einem 16kByte Block verteilt plaziert werden. Die nachfolgende Tabelle verdeutlicht diesen Zusammenhang noch einmal:

Spritezeiger

2040	Spriteblock von Sprite 0	Adresse=Spriteblock * 64
2041	Spriteblock von Sprite 1	Adresse=Spriteblock * 64
2042	Spriteblock von Sprite 2	Adresse=Spriteblock * 64
2043	Spriteblock von Sprite 3	Adresse=Spriteblock * 64
2044	Spriteblock von Sprite 4	Adresse=Spriteblock * 64
2045	Spriteblock von Sprite 5	Adresse=Spriteblock * 64
2046	Spriteblock von Sprite 6	Adresse=Spriteblock * 64
2047	Spriteblock von Sprite 7	Adresse=Spriteblock * 64
2048	Spriteblock von Sprite 8	Adresse=Spriteblock * 64

An dieser Stelle möchte ich meine Einweisung in die Welt der Sprites beenden. Im nächsten Teil werden wir mit Hilfe unserer bisherigen Kenntnisse einige nützliche Sprite Experimente in Maschinensprache durchführen.

Frank Brall

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

C-64 LISTING + CHECKSUMMEN (C V1.0)

```

0 DATA 60000:REM ERSTE SPRITE-DATAZEILE <44>
1 REM *** SPRITE-DESIGNER *** <10>
2 REM ASSEMBLERKURS TEIL 20 <26>
3 REM (C) 1983 BY C.LIPPE <174>
4 REM <147>
5 REM <148>
6 REM <149>
7 PRINT "CLEAR DOWN3 RIGHT2"***<SPACE>SPRITE-DAT
A<SPACE>*** <149>
8 PRINT "DOWN2 SOLLEN<SPACE>SPRITE-DATAS<SPACE>I
N<SPACE>PROGRAMM<SPACE>EINGE-<SPACE2>FUEGT<SPACE
>WERDEN?(J/N)" <175>
9 POKE198,0:WAIT198,1:GETA$:IFA$="J"THENPOKE198,
0:GOTOB7 <87>
10 PRINT "CLEAR":POKE650,128:FORI=1TO21:FORJ=1
TO24:PRINT ".":NEXT:PRINT:NEXT <178>
11 I=24:PRINT "HOME"TAB(I)"<DOWN RIGHT2>SPRITE-
DATA":PRINTTAB(I)"<RIGHT2>=====" <111>
12 PRINTTAB(I)"<DOWN RIGHT>CRSR-TABTEN<SPACE>=>"
:PRINTTAB(I)"<RIGHT3>STEUERUNG" <250>
13 PRINTTAB(I)"<DOWN RIGHT>I*3<SPACE>=>":PRINTTA
B(I)"<RIGHT3>PUNKT<SPACE>BETZEN" <205>
14 PRINTTAB(I)"<DOWN RIGHT><SPACE><SPACE>=>":PRI
NTTAB(I)"<RIGHT3>P.<SPACE>LIESCHEN" <55>
15 PRINTTAB(I)"<DOWN RIGHT>F1<SPACE>=>":PRINTTAB
(I)"<RIGHT3>SPRITE<SPACE>VER-":PRINTTAB(I)"<RI

```



```

73)ARBEITEN"
16 PRINTTAB(1)"(DOWN RIGHT) [HOME] (SPACE) =>": PRIN
TTAB(1)"(RIGHT) CURSOR (SPACE) HOME"
17 PRINTTAB(1)"(DOWN RIGHT) [CLR] (SPACE) =>": PRINT
TAB(1)"(RIGHT) BILDSCHIRM": PRINTTAB(1)"(RIGHT) L
DESCHEN"
19 PRINT"(DOWN RIGHT) (SPACE) (C) (SPACE) 1983 (SPACE
) BY (SPACE) LIPPE (SPACE) ENTERPRISES (HOME)": X=0: Y=0
: VR=1024
20 Z=PEEK(VR+X+Y*40)
21 POKEVR+X+40*Y, 160
22 GETA#: POKEVR+X+Y*40, Z: IFA#="" THEN 21
23 IFA#="(F1)" THEN 50
25 IFA#="(DOWN)" THEN Y=Y+1
30 IFA#="(UP)" THEN Y=Y-1
33 IFA#="(HOME)" THEN X=0: Y=0
35 IFA#="(CLEAR)" THEN GOTO 10
37 IFA#="(SPACE)" THEN POKEVR+X+40*Y, 46: X=X+1
40 IFA#="(RIGHT)" THEN X=X+1
42 IFA#="(LEFT)" THEN X=X-1
43 IFA#="" THEN POKEVR+X+Y*40, 42: X=X+1
44 IF X>23 THEN X=0: Y=Y+1
45 IF Y<0 THEN Y=0
46 IF X<0 THEN X=0
47 IF Y>20 THEN Y=20
48 GOTO 20
50 DIM W(64): X=0: Y=0: PRINT"(HOME DOWN 22 RIGHT 6) BI
TTE (SPACE) WARTEN!"
55 FOR I=0 TO 20: FOR J=0 TO 23
60 IF PEEK(VR+I*40+J)=42 THEN W(I*3+INT(J/8))=W(I*3
+INT(J/8)) OR 2^(7-(J AND 7))
70 NEXT J: NEXT I: READ Z: INPUT"(CLEAR DOWN 4) NAME (SPACE
) DES (SPACE) SPRI TES: ": C$
71 PRINT"(CLEAR DOWN 2) Z" REM (SPACE) C$
72 PRINT Z+10"(LEFT) DATA": FOR I=0 TO 15: PRINT RIGHT$(
STR$(W(I)), LEN(STR$(W(I)))-1),";
73 NEXT I: PRINT"(LEFT) SPACE"
74 PRINT Z+20"(LEFT) DATA": FOR I=16 TO 31: PRINT RIGHT
$(STR$(W(I)), LEN(STR$(W(I)))-1),";
75 NEXT I: PRINT"(LEFT) SPACE"
76 PRINT Z+30"(LEFT) DATA": FOR I=32 TO 48: PRINT RIGHT
$(STR$(W(I)), LEN(STR$(W(I)))-1),";
77 NEXT I: PRINT"(LEFT) SPACE"
78 PRINT Z+40"(LEFT) DATA": FOR I=49 TO 62: PRINT RIGHT
$(STR$(W(I)), LEN(STR$(W(I)))-1),";
79 NEXT I: PRINT"(LEFT) SPACE"
80 PRINT"ODATA Z+50
81 PRINT"GOTO 83"
82 FOR X=63 TO 640: POKE X, 13: NEXT X: POKE 198, 10: PRINT
(HOME)": : END
83 PRINT"(CLEAR DOWN 5) NOCH (SPACE) EINMAL? (J/N)"
84 GETA#: IFA#="J" THEN RUN 10
85 IFA#="N" THEN Z=0: GOTO 98
86 GOTO 84
87 INPUT"(DOWN 2) PROGRAMMNAME: ": B$
88 PRINT"(DOWN RYSON) C (RVSOFF) ASSETTE (SPACE) ODER
(SPACE RYSON) D (RVSOFF) ISKETTE?": WAIT 198, 1: GETA#:
IFA#="C" OR A#="D" THEN 87
89 PRINT"(DOWN) TASTE (SPACE) DRUECKEN, WENN": IFA#=""
D" THEN PRINT"RICHTIGE (SPACE) DISKETTE (SPACE) EINGEL
EHT."
90 IFA#="D" THEN PRINT"CASSETTE (SPACE) AUF (SPACE) AN
FANG (SPACE) PROGRAMM (SPACE) GESPULT (SPACE 4) WURDE."
91 POKE 198, 0: WAIT 198, 1: PRINT"(DOWN 2 RIGHT 5) BITTE
(SPACE) WARTEN!"
92 FOR X=2048 TO 1015: IF PEEK(X) OR PEEK(X+1) OR PEEK(X+2
) THEN NEXT
93 X=X+1: POKE 43, X AND 255: POKE 44, X/256
94 PRINT"(CLEAR DOWN 2) LOAD"CHR$(34) B#CHR$(34): : I
FA#="D" THEN PRINT", B"
95 PRINT: PRINT"(DOWN 10) POKE 43, 1: POKE 44, 8: RUN 10":
FOR X=63 TO 640: POKE X, 13: NEXT
96 POKE 198, 10: PRINT"(HOME)": : END
98 PRINT"(CLEAR DOWN 2) Z: PRINT"Z="Z+1": GOTO 98 (HO
ME)": : POKE 631, 13: POKE 632, 13: POKE 198, 2: END
ENDE DES LISTINGS

```

Interruptprogrammierung auf dem Commodore 64

Artikelserie von Daniel Durstewitz und Waldemar Raaz

Teil 2

Ergänzung zum Artikelabschnitt »RESET«

Eigentlich sollte in der letzten Folge dieses Kurses im Anschluß an den Artikelabschnitt »RESET« ein kurzes Maschinenprogramm folgen, das die Initialisierung des RAMs, der Vektoren usw. übernimmt. Wie Sie sicher schon bemerkt haben, wurde dieses Programm leider vergessen (so etwas kommt in den besten Verlagshäusern vor). Dies sollte aber kein Anlaß zu Trübsinn sein, da wir den Abdruck an dieser Stelle nachholen:

STARTROUTINE FUER EIGENE PROGRAMME
=====

```

JSR #FDA3 ; INTERRUPT VORBEREITEN
JSR #FD50 ; ARBEITSPEICHER INITIALISIEREN
JSR #FD15 ; HARDWARE- UND I/O-VEKTOREN SETZEN
JSR #FF5B ; VIDEO-RESET
CLI ;
JSR #E453 ; BASIC-VEKTOREN SETZEN
JSR #E3BF ; RAM INITIALISIEREN
JSR #A644 ; NEW-BEFEHL AUSFUEHREN
JMP #E386 ; NMI-EINSPRUNG ZUM WARMSTART

```

3. NMI (Not Maskable Interrupt)

Während die Anwendungsmöglichkeiten des Reset und des Breaks relativ gering sind, wie die Ausführungen gezeigt haben, kommen wir bei den folgenden beiden Interruptarten auf den eigentlichen Kern des Artikels, der sinnvollen Interruptprogrammierung. Zunächst einmal der NMI (=Not Maskable Interrupt, Nicht Maskierbarer Interrupt): dieser nicht maskierbare, d.h. nicht durch Software abschaltbare Interrupt wird ausgeführt, wenn Pin 4 des 6510 kurzzeitig auf Masse gelegt wird. Beim C64 kann dies auf drei verschiedene Arten geschehen. Die erste Möglichkeit ist ähnlich der Resetauslösung, nämlich über eine Schnittstelle des C64. Dazu liegt am Expansionport an Pin D das -NMI-Signal des Prozessors

an. Folglich wird ein NMI ausgelöst, sobald dieser Pin mit dem Massepin am Expansionport verbunden wird. Diese NMI-Quelle ist für externe Baugruppen oder Schnittstellen gedacht, die am Expansionport des C64 betrieben werden. Daher ist diese Möglichkeit für uns weniger interessant, es ist eher ein Gebiet, auf dem die Hardware-Freaks zum Zuge kommen.

Die zweite Möglichkeit, einen NMI auszulösen, dürfte selbst dem blutigsten Einsteiger bekannt sein, da man von dieser Möglichkeit eigentlich ständig Gebrauch macht: der Druck auf die RESTORE-Taste. Auch das löst einen Interrupt aus. Der Computer unterbricht das laufende Programm, das er gerade bearbeitet, rettet Programmzähler und Prozessorstatus, holt aus den Speicherzellen 65530/65531 (hex.

\$FFFA/FFFB) die Startadresse der NMI-Routine und verzweigt dorthin. In dieser Routine prüft er unter anderem auch, ob die RUN/STOP-Taste gedrückt ist.

Ist dies nicht der Fall, so holt der Rechner Programmzähler und Prozessorstatus wieder vom Stack und fährt mit der Bearbeitung des laufenden Programmes fort, als sei er gar nicht unterbrochen worden. Ist die RUN/STOP-Taste hingegen gedrückt, bricht er die Programmbearbeitung endgültig ab und führt einen Basic-Warmstart durch - der Computer meldet sich mit einem »READY.« und blinkendem Cursor in der linken, oberen Bildschirmcke. An dieser Stelle zwei nützliche Tips:

- Wenn man verhindern möchte, daß ein Anwender ein selbstgeschriebenes Programm mit STOP bzw. STOP/RESTORE anhalten kann, so verbiegt man einfach den STOP-Vektor mit POKE808,251.
- Bei Maschinenprogrammen bietet es sich an, den NMI-Vektor auf den Beginn des eigenen Programms zu verbiegen. Der Anwender kann das Programm dann nicht mehr mit der Tastenkombination STOP/RESTORE anhalten, da bei Betätigung der RESTORE-Taste lediglich das Programm neu gestartet wird. Dazu legt man in der Speicherzelle 792 (hex. \$0318) das Low-Byte der Startadresse des eigenen Programms ab und in Speicherzelle 793 (hex. \$0319) das High-Byte. Bedenken Sie jedoch, daß in dieser Konstellation bei einem NMI nicht mehr die entsprechende Betriebssystemroutine durchlaufen wird. Dies schadet nicht weiter, doch sollten Sie nicht vergessen, mit einem CLI zu Beginn Ihres Programms den Systeminterrupt wieder freizugeben, den der Rechner bei erfolgtem NMI sperrt, da sich der Rechner sonst meist »aufhängt«.

Der durch die CIA #2 ausgelöste Interrupt

Die dritte Möglichkeit, einen NMI auszulösen, besteht darin, das ICR (Interrupt Control Register) in der CIA #2 entsprechend zu programmieren. Dieser Baustein schiebt nämlich ein Low-Signal an Pin 4 des Prozessors, wenn im ICR bestimmte Voraussetzungen erfüllt sind, und löst damit einen NMI aus. Beim ICR

(CIA-Register 13) muß man zwei Betriebsarten unterscheiden: bei einem Lesezugriff über einen Ladefehler auf das Register werden die Daten aus dem INT DATA-Register ausgelesen und zugleich alle (!) Bits des Registers gelöscht. Bei einem Schreibzugriff auf das ICR per Speicherbefehl schreibt man ins INT-MASK-Register; dabei dient Bit 7 dazu, zwischen einer Art Bit-Setz- und Bit-Lösch-Modus umzuschalten: bei gesetztem Bit 7 setzt jedes 1er-Bit unter den übrigen sieben Bit das entsprechende INT MASK-Bit, während ein 0er-Bit das entsprechende Bit unbeeinflusst läßt. Ist Bit 7 dagegen nicht gesetzt, löscht ein 1er-Bit das zugehörige INT MASK-Bit und 0er-Bits sind wieder ohne Einfluß. Die Bits im ICR haben folgende Bedeutung:

- Bit 0: 1=Unterlauf von Timer A (Register4/5)
- 1: 1=Unterlauf von Timer B (Register6/7)
- 2: 1=Uhrzeit und vorgewählte Alarmzeit stimmen überein (Register8-11)
- 3: 1=SDR (Serial Date Register) ist voll oder leer, je nach Betriebsart (Register12)
- 4: 1=negative Flanke am -Flag-Pin
- 5: nicht belegt
- 6: nicht belegt
- 7: INT DATA : 1=mindestens ein Bit im INT MASK- und im INT DATA-Register stimmen überein
INT MASK : Schaltbild Bit setzen / Bit löschen (s. Text)

Die CIA #2 löst einen NMI aus, sobald im INT DATA- sowie im INT MASK-Register ein und dasselbe Bit auf eins gesetzt sind. Dazu muß das entsprechende Bit im INT MASK-Register vom Anwender gesetzt werden, um die betreffende Interruptquelle quasi freizugeben. Trifft das zugehörige Ereignis dann ein, so setzt der Rechner auch das entsprechende Bit im INT DATA-Register hoch, und es erfolgt ein NMI, da auch das siebte Bit im INT DATA-Register auf eins gesetzt wird.

Die Interruptquellen

Sehen wir uns die Ereignisse, die einen NMI der CIA #2 auslösen können etwas näher an: da sind zunächst einmal die beiden Timer A und B in der CIA. Es handelt

sich dabei um zwei 16-Bit Zählerregister, die in Low-/High-Byte-Darstellung die Register 4/5 (TimerA) und 6/7 (TimerB) in der CIA belegen. Schreibt man in diese Register einen Startwert hinein, so wird dieser im Systemtakt um jeweils eine Einheit herabgezählt, bis der Wert Null erreicht ist. Dann wird das entsprechende Bit im INTER DATA Register gesetzt, und abhängig vom Zustand des INT MASK-Register, das bestimmt, ob diese Interruptquelle freigegeben ist, erfolgt ein NMI - oder er wird unterdrückt. Die nächste Interruptquelle ist die Echtzeituhr in der CIA. Diese Uhr ist nämlich mit der Alarmfunktion versehen: man kann eine bestimmte Alarmzeit vorwählen, und wenn die Uhrzeit erreicht, so wird Bit 2 im INT DATA-Register gesetzt, und in Abhängigkeit vom INT MASK-Register kann wiederum ein NMI erfolgen. Diese Interruptquelle wird weiter unten noch einmal aufgegriffen und anhand eines Demoprogrammes näher erläutert. Bit 3 des ICR wird durch Register 12 der CIA, das SDR beeinflusst. Das SDR ist ein serielles Schieberregister und gehört eigentlich mehr in den Bereich der Hardware-Bastler, doch soviel sei hier gesagt: mit Hilfe des SDR ist es möglich, ein Byte bitweise - also seriell - über den Pin SP am Userport auszugeben oder einzulesen. Dazu wird Bit für Bit entweder aus dem SDR (Register 12) auf den Pin SP hinausgeschoben (mit dem höchstmöglichen Bit voran), oder aber Bit für Bit wird von SP aus ins SDR geschoben. Ob sich das SDR im Eingabe- oder im Ausgabe-Modus befindet, darüber entscheidet Bit 6 des Registers 15 in der CIA. Ist es gesetzt, so ist das SDR Eingaberegister - ansonsten Ausgaberegister. Unabhängig davon ist das SDR nach acht Schiebeimpulsen entweder voll oder leer, so daß neue Daten benötigt werden. Dies anzuzeigen, ist die Aufgabe des dritten Bit im INT DATA-Register; nach jeweils acht Schiebeimpulsen wird es nämlich auf eins gesetzt und kann somit signalisieren, wenn Daten am SDR benötigt werden oder vorhanden sind. Mit diesem Register lassen sich sehr einfach und komfortabel serielle Schnittstellen programmieren. Demjenigen, der sich damit weiter beschäftigen will, ist aber auf jeden Fall ein genaues Studium der Datenblätter

zum CIA6526 zu empfehlen, da sich auch noch einige Timing-Probleme ergeben, die hier zu erläutern zu weit führen würde. Ebenfalls in den Bereich der Hardware gehört die letzte Interruptquelle der CIA: Bit 4 des INT DATA-Registers wird nämlich immer dann auf eins gesetzt, wenn am Pin-Flag am Userport (Pin B) eine negative Flanke, also ein Wechsel von logisch high auf logisch low auftritt. Diese Interruptquelle eignet sich folglich hervorragend zur Messung und Erfassung von Impulsen. Denkbar wäre zum Beispiel eine kleine Schaltung aus zwei Lichtschranken, die mit dem -Flag-Eingang gekoppelt sind. Mit der ersten Lichtschranke könnte man die Echtzeituhr in der CIA starten und entsprechend mit der zweiten stoppen. So wäre eine äußerst exakte Zeitmessung zu realisieren. Dies nur als Beispiel und Denkanstoß, der Phantasie des Lesers sind in dieser Hinsicht keine Grenzen gesetzt.

Auswertung der Interruptimpulse

Wir wissen jetzt genau, wie und wodurch ein NMI ausgelöst wird, doch zunächst nützt uns dieses Wissen herzlich wenig; denn die bestehende NMI-Routine, die einen erfolgten NMI behandelt und auswertet, steht im ROM und kann von uns nicht beeinflusst werden, so daß sie einen NMI auch nicht nach unseren Vorstellungen auswerten und entsprechend reagieren kann. Einen Eingriff in diese ROM-Routine macht erst die Vektoren-Technik möglich, derer sich der C64 reichlich bedient. Gleich zu Beginn der NMI-Routine nämlich verzweigt der Rechner über einen Vektor, der im RAM (!) liegt, auf den nächsten Befehl der Routine. Dies erscheint auf den ersten Blick völlig unnötig, sogar unsinnig, doch erst durch den RAM-Vektor, den man ja beliebig verändern kann, wird es möglich, die ROM-Routine zu verändern, oder richtiger gesagt, zu erweitern. Man verbiegt einfach den Vektor auf eine eigene Routine, die am Ende zur NMI-Betriebssystemroutine verzweigt oder mit RTI abschließt, in der man dann bei entsprechenden Abfragen programmieren kann, man schiebt also gleichsam einen Keil ins Betriebssystem.

Fortsetzung folgt!

Bewerbung als Programmator

Name: _____ Vorname: _____ Alter: _____

Straße: _____ Wohnort: _____

Titel des Programmes: _____ Computersystem: _____

Erforderlicher Speicherplatz: _____ KBytes

Erforderliche Peripherie: _____

Ich versichere hiermit, daß ich der Autor des oben genannten Programmes bin und alle Rechte besitze. Ich bin damit einverstanden, daß Sie mein Programm in einer Zeitschrift abdrucken und/oder dieses verkaufen. Das einmalige Honorar beträgt 120 DM pro abgedruckter Seite. Sollte das Programm nicht veröffentlicht werden, bitte ich um Rücksendung meiner Unterlagen. Sollte mein Programm in der Zwischenzeit von anderer Seite erworben worden sein, werde ich Sie unverzüglich benachrichtigen. Ich bestätige durch meine Unterschrift, daß ich Kenntnis davon habe, daß der Tronic-Verlag gegen Personen, die Programme einschicken, deren Urheberrechte sie nicht besitzen, rechtliche Schritte einleitet.

Unterschrift des Programmators (bei Minderjährigen auch die eines Erziehungsberechtigten): _____

Folgende Unterlagen liegen diesem Schreiben bei:

Kassette: _____ Diskette: _____ Dokumentation: _____ Listing: _____ Rückporto: _____

Unterschrift des Programmators (bei Minderjährigen auch die eines Erziehungsberechtigten): _____

Einsenden an: Tronic-Verlag Abt. Programmentwicklung, Am Stad, 3440 Eschwege (Bitte zweimal unterschreiben!)

Schneider

Electric Gladiators

Spielbeschreibung

Electric Gladiators ist ein Spiel für zwei Personen, es kann entweder nur mit Joysticks, oder mit Joystick und Tastatur gespielt werden.

Es handelt sich dabei um eine Spielidee aus dem Film Tron. Die Spieler steuern jeweils ein Lichtmotorrad, das eine Lichtmauer auf dem Spielraster hinter sich herzieht. Man darf weder die Lichtmauer des Gegners berühren, noch die eigene oder die Außenwände des Spielfeldes.

Mit steigendem Level, nimmt auch die Anzahl der im Rastergitter vorhandenen Löcher zu.

Sollte einer der Spieler einem tragischen Unfall erliegen, so explodiert er in der Farbe des Gegners, also bei einem roten Motorrad, eine blaue Explosion. Zudem blinkt die gefahrene Strecke des Spielers auf dem Raster auf. Das Spiel endet nach einer vom Spieler vorbestimmten Anzahl von Runden.

Der bessere Spieler hat nun die Möglichkeit, sich im Highscore zu »verewigen«.

Start des Programms:

In das Spielmenü gelangt man durch Drücken der SPACE-Taste.

Nun muß man die Anzahl der zu fahrenden Level (Runden) eingeben. Danach hat man die Auswahl zwischen Joystick-Joystick oder Joystick-Keybord. Bei letzteren steuert Spieler 1 mit dem Joystick und Spieler 2 mit dem Cursortasten. Im Highscore-Menü kann man mit einem Motorrad über das Alphabet fahren, und durch Druck auf den Feuerknopf die Buchstaben anwählen. Aus dem Highscoremenü kommt man, indem man das Ausrufungszeichen (!) auswählt.

Das Programm ist vollständig in BASIC 1.0 geschrieben und läuft auf allen Schneider CPC's. Das Programm hat eine Länge von ca. 7K.

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```
10 REM *****
*** (NJ)
20 REM ***
*** (EM)
30 REM *** ELECTRIC GLADIATORS
*** (PA)
```



```

40 REM *** PROGRAMMIERT IN
***
50 REM *** DER ZEIT VOM
***
60 REM *** 27.-29.4 1986
***
70 REM *** VON MARK HESSBURG
***
80 REM ***
***
90 REM *****
***
100 REM
110 REM *** Sonderzeichen und D
120 REM
130 DIM p(500):DIM o(500)
140 SYMBOL 255,6,6,12,26,255,63
,85,34:SYMBOL 254,0,0,0,24,24,0
,0,0:SYMBOL 253,0,0,60,60,60,60
,0,0:SYMBOL 252,0,126,126,126,1
26,126,126,0:SYMBOL 251,255,255
,255,255,255,255,255,255
150 SPEED INK 5,5:INK 2,6:INK 3
,14:INK 1,0:INK 0,1:BORDER 0:PA
PER 1
160 high=5000:na$="Compute mit"
170 GOTO 1480
180 REM
190 REM *** Farben und Variabel
n ***
200 REM
210 INK 2,6:INK 3,14:INK 1,0:IN
K 0,1:BORDER 0:PAPER 1:GOTO 128
0
220 q=120:w=200:e=520:r=200:x=8
:y=4
230 MODE 1
240 pu=4356
250 REM
260 REM *** Bildschirmaufbau **
*
270 REM
280 ru=ru+1:PAPER 1:PEN 0:CLS
290 LOCATE 17,12:PRINT"Get Read
y"
300 LOCATE 16,14:PRINT"for Leve
l";ru
310 FOR t=1 TO 2000:NEXT
320 CLS
330 FOR t=0 TO 640 STEP 10:MOVE
0,t:DRAW 640,t,0:MOVE t,0:DRAW
t,400,0:NEXT
340 LOCATE 1,1:PRINT"
";LOCATE 1,2:PRINT"
";
350 FOR t=0 TO 10:MOVE t,0:DRAW
t,400,0:NEXT
360 FOR t=0 TO 10:MOVE t,0:DRAW
t,400,0:NEXT
370 FOR t=630 TO 640:MOVE t,0:D
RAW t,400,0:NEXT
380 FOR t=0 TO 10:MOVE 0,t:DRAW
640,t,0:NEXT
390 FOR t=350 TO 367:MOVE 0,t:D
RAW 640,t,0:NEXT
400 PLOT 10,10,2:DRAW 630,10:DR
AW 630,350:DRAW 10,350:DRAW 10,
10:PLOT 8,8,3:DRAW 632,8:DRAW 6
32,352:DRAW 8,352:DRAW 8,8
410 LOCATE 2,1:PEN 3:PRINT"Play
er 1";sc1:LOCATE 21,1:PEN 2:P
RINT"Player 2";sc2:LOCATE 2,2
:PEN 3:PRINT"LEVEL: ";ru
420 FOR t=1 TO ru+5:p(ru)=INT(R
ND*36)+3:p(ru)=INT(RND*19)+4:LO
CATE p(ru),o(ru):GOSUB 1680:NEX
T
430 REM
440 REM *** steuerung ***
450 REM
460 IF JOY(0)=1 THEN x=1
470 IF con$="K"THEN IF INKEY(0)
=0 THEN y=1
480 IF con$="J"THEN IF JOY(1)=1
THEN y=1
(EJC) 490 IF JOY(0)=2 THEN x=2
500 IF con$="K"THEN IF INKEY(2)
=0 THEN y=2
(AJB) 510 IF con$="J"THEN IF JOY(1)=2
THEN y=2
(JJP) 520 IF JOY(0)=4 THEN x=4
530 IF con$="K"THEN IF INKEY(8)
=0 THEN y=4
(BJD) 540 IF con$="J"THEN IF JOY(1)=4
THEN y=4
(GJC) 550 IF JOY(0)=8 THEN x=8
560 IF con$="K"THEN IF INKEY(1)
=0 THEN y=8
(FBJ) 570 IF con$="J"THEN IF JOY(1)=8
THEN y=8
(HFJ) 580 REM
(LD) 590 REM *** sound ***
600 REM
610 SOUND 1,0,1,15
620 REM
630 REM *** Bildschirmausgabe *
**
(CDP) 640 REM
650 IF x=1 THEN IF TEST(q,w+10)
=0 THEN MOVE q,w:w=w+10:DRAW q
,w,3 ELSE MOVE q,w:w=w+10:DRAW q
,w,3:FOR t=15 TO 9 STEP-0.5:SOU
ND 1,626,5,t,,,1:NEXT:sp1=1:GOT
O 780
(FA) 660 IF y=1 THEN IF TEST(e,r+10)
=0 THEN MOVE e,r:r=r+10:DRAW e
,r,2 ELSE MOVE e,r:r=r+10:DRAW e
,r,2:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp2=1:GOTO
780
(GBK) 670 IF x=2 THEN IF TEST(q,w-10)
=0 THEN MOVE q,w:w=w-10:DRAW q
,w,3 ELSE MOVE q,w:w=w-10:DRAW q
,w,3:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp1=1:GOTO
780
(PA) 680 IF y=2 THEN IF TEST(e,r-10)
=0 THEN MOVE e,r:r=r-10:DRAW e
,r,2 ELSE MOVE e,r:r=r-10:DRAW e
,r,2:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp2=1:GOTO
780
(FP) 690 IF x=4 THEN IF TEST(q+10,w)
=0 THEN MOVE q,w:w=q+10:DRAW q
,w,3 ELSE MOVE q,w:w=q+10:DRAW q
,w,3:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp1=1:GOTO
780
(AI) 700 IF y=4 THEN IF TEST(e-10,r)
=0 THEN MOVE e,r:r=e-10:DRAW e
,r,2 ELSE MOVE e,r:r=e-10:DRAW e
,r,2:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp2=1:GOTO
780
(CH) 710 IF x=8 THEN IF TEST(q+10,w)
=0 THEN MOVE q,w:w=q+10:DRAW q
,w,3 ELSE MOVE q,w:w=q+10:DRAW q
,w,3:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp1=1:GOTO
780
(DM) 720 IF y=8 THEN IF TEST(e+10,r)
=0 THEN MOVE e,r:r=e+10:DRAW e
,r,2 ELSE MOVE e,r:r=e+10:DRAW e
,r,2:FOR t=15 TO 9 STEP-1:SOUND
1,626,5,t,,,1:NEXT:sp2=1:GOTO
780
(FF) 730 pu=pu-0.5
740 GOTO 460
(BN) 750 REM
760 REM *** Level und Punkte Ab
frage ***
770 REM
780 IF sp1=1 THEN sp1=0:sc2=sc2
+pu:GOSUB 1410:FOR t=1 TO 50:PR
INT:NEXT
790 IF sp2=1 THEN sp2=0:sc1=sc1
+pu:GOSUB 1430:FOR t=1 TO 50:PR
INT:NEXT
800 IF ru=eru THEN CLS:GOTO 850
810 GOTO 220
(CBJ) 820 REM
(GF) 830 REM *** spielende ***
840 REM
850 IF sc1>sc2 THEN a$="Player
1":IF sc1>high THEN high=sc1:ja
=1
860 IF sc2>sc1 THEN a$="Player
2":IF sc2>high THEN high=sc2:ja
=1
870 CLS:PEN 0:PAPER 1:LOCATE 5,
11:PRINT"Congratulations ";a$;"
you are"
880 LOCATE 9,13:PRINT"the winne
r of this match"
890 GOSUB 1800
900 IF ja<>1 THEN GOTO 1480
910 CLS
920 REM
930 REM *** high score ***
940 REM
950 na$=""
960 PLOT 0,0,2:DRAW 0,399:DRAW
639,399:DRAW 639,0:DRAW 0,0
970 PLOT 10,10,2:DRAW 10,389:DR
AW 629,389:DRAW 629,10:DRAW 10,
10
980 z=1:ja=0:WINDOW#1,5,15,3,13
:PAPER#1,2:PEN#1,0:CLS#1:PRINT#
1,a$:PRINT#1,"You have":PRINT#1
,"the high-":PRINT#1,"score!":P
RINT#1:PRINT#1,"Please use":PRI
NT#1,"Joystick #0":PRINT#1,"to
enter":PRINT#1,"your name!"
990 PEN 3:LOCATE 25,3:PRINT"ELE
CTRIC":LOCATE 24,4:PRINT"GLADIA
TORS":PEN 2:LOCATE 21,6:PRINT"H
IGH SCORE TABLE"
1000 PLOT 290,280:DRAW 290,388:
DRAW 610,388,1:DRAW 610,280,2:D
RAW 290,280
1010 PLOT 300,290,2:DRAW 300,38
8:DRAW 600,388:DRAW 600,290:DR
AW 300,290
1020 WINDOW#2,7,34,20,23:PAPER#
2,3:PEN#2,1:CLS#2
1030 LOCATE#2,1,3:PRINT#2,"ABCD
EFGHIJKLMNOPQRSTUVWXYZ !";
1040 WINDOW#3,10,31,17,18:PAPER
#3,0:PEN#3,1:CLS#3:LOCATE#3,1,1
:PRINT#3,"Highscore";high;
1050 a=1
1060 b$=INKEY$:IF b$<>" THEN GO
TO 1060
1070 GOTO 1090
1080 b$=INKEY$:IF b$=""THEN GOT
O 1080
1090 LOCATE#2,a,1:PRINT#2," ";
1100 IF JOY(0)=4 THEN a=a-1:IF
a=0 THEN a=1
1110 IF JOY(0)=8 THEN a=a+1:IF
a=29 THEN a=28
1120 LOCATE#2,a,1:PRINT#2,CHR$(
255);
1130 IF JOY(0)<>16 THEN GOTO 10
80
1140 IF a<27 THEN GOTO 1180
1150 IF a=27 THEN b$=""
1160 IF a=28 THEN b$=""
1170 GOTO 1200
1180 m=0:FOR t=65 TO 90:m=m+1:I
F m=a THEN b$=CHR$(t)
1190 NEXT
1200 c=c+1:IF c=23 THEN GOTO 14
80
1210 IF a<>28 THEN na$=na$+b$
1220 LOCATE#3,1,2:PRINT#3,na$;
1230 IF ASC(b$)=33 THEN GOTO 14
80
1240 GOTO 1060
1250 REM
1260 REM *** titelbild ***
1270 REM
1280 MODE 1:CLS:PEN 3:PRINT"
ELECTRIC GLADIATORS
";PRINT:PRINT"-----
";

```


programme

```

PEN 2:PRINT"WRITTEN 1986 B
V MARK HESSBURG":PRINT:PRINT"FI
RST NATIONAL":PRINT"SOFTWARE
COMPANY":PRINT:PRINT"$ TRONIC
VERLAG
1290 ru=0:PRINT:PRINT"HIGHSCORE
":high:PRINT:PRINT na$
1300 LOCATE 1,17:INPUT"HOW MANY
LEVELS ? ",eru
1310 IF ERU<1 OR ERU>300 THEN G
OTO 1300
1320 LOCATE 1,20:INPUT"JOYSTICK
/KEYBOARD OR JOYSTICK/JOYSTICK
CONTROL (K/J) ";con$
1330 sci=0:sc2=0
1340 IF con$="K"OR con$="J"THEN
GOTO 220
1350 IF con$="k"THEN con$="K":G
OTO 220
1370 GOTO 1320
1380 REM
1390 REM *** EXPLOSION ***
1400 REM
1410 INK 3,14,0:PLOT q,w,3:FOR
t=1 TO 400 STEP 10:MOVE q,w:PLD
TR t,t:MOVE q,w:PLDTR-t,t:MOVE
q,w:PLDTR-t,t:MOVE q,w:PLDTR t
,t:MOVE q,w:PLDTR 0,t:MOVE q,w
:PLDTR 0,-t:MOVE q,w:PLDTR-t,0:
MOVE q,w:PLDTR t,0:GOUND 1,500,
1,15:GOUND 1,500,1,3:NEXT
1420 INK 3,14:RETURN
1430 INK 2,6,0:PLOT e,r,2:FOR t
=1 TO 400 STEP 10:MOVE e,r:PLD
R t,t:MOVE e,r:PLDTR-t,t:MOVE
e,r:PLDTR-t,t:MOVE e,r:PLDTR t,
-t:MOVE e,r:PLDTR 0,t:MOVE

```

```

PLOT 0,-t:MOVE e,r:PLDTR-t,0:M
OVE e,r:PLDTR t,0:GOUND 1,500,1
,15:GOUND 1,500,1,3:NEXT
1440 INK 2,6:RETURN
1450 REM
1460 REM *** Titelgrafik ***
1470 REM
1480 MODE 0:FOR t=0 TO 10:INK t
,0:NEXT:BORDER 0
1490 INK 15,26:GOTO 1590
1500 FOR f=26 TO 1 STEP-1
1510 FOR q=1 TO 3
1520 FOR t=10 TO 0 STEP-1
1530 a$=INKEY$:IF a$<" "THEN GO
TO 1640
1540 INK t,f:INK t+1,0
1550 CALL &BD19
1560 NEXT
1570 NEXT
1580 NEXT:GOTO 1500
1590 RESTORE 1630:FOR t=1 TO 10
:READ a$:READ b$:FOR q=a TO b STE
P 2:PLOT 0,400-q:DRAW 640,400-q
,t:PLD 0,q:DRAW 640,q,t:NEXT:N
EXT
1600 PAPER 2:PEN 15:LOCATE 1,5:
PRINT" M. Hessburg presents:
1610 PAPER 2:PEN 15:LOCATE 1,22
:PRINT"ELECTRIC GLADIATORS":
1620 GOTO 1500
1630 DATA 1,50,50,90,90,125,125
,150,150,170,170,185,185,189,18
9,194,194,199,199,200
1640 GOTO 180
1650 REM
1660 REM *** Loecher sprengen *
**
1670 REM
1680 SOUND 1,10,5,15,0,0,7:GOUN

```

```

D 1,100,8,15,0,0,7
1690 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(254)
1700 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(253)
1710 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(252)
1720 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(251)
1730 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(252)
1740 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(253)
1750 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(254)
1760 CALL &BD19:LOCATE p(ru),o(
ru):PRINT CHR$(32)
1770 REM
1780 RETURN
1790 REM *** High-Score Music *
**
1800 FOR a=1 TO 2:RESTORE 1820:
FOR t=1 TO 26:READ na1:READ na2
:GOUND 1,na1,20,7,0,0:GOUND 2,n
a2,20,7,0,0:NEXT:NEXT
1810 REM
1820 DATA 1911,478,1703,426,151
7,379,1432,358,1911,478,0,0,143
2,358,1517,379,1432,358,1276,31
9,1703,426,0,0,1703,426,0,0,170
3,426,1517,379,1432,358,1136,28
4,1276,319,1276,319,1432,358,14
32,358,1517,379,1703,426,1517,3
79,1911,478
1830 FOR t=1 TO 1200:NEXT
1840 RETURN

```

ENDE DES LISTINGS

Bildschirm-Toolbox:

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```


1 *****
2 *      Bildschirm Toolbox
3 *      (c) 1986
4 *      OTTFRIED SCHMIDT
5 *****
6 *
7 *
1000 DATA 11,00,40,21,00,C0,01,
00
1010 DATA 40,ED,B0,C9,21,00,C0,
06
1020 DATA 03,11,03,00,ES,CD,19,
BD
1030 DATA CD,20,92,E1,23,10,F5,
C9
1040 DATA CB,BC,7E,CB,FC,77,CD,
2D
1050 DATA 92,19,DB,1B,F3,3E,0A,
3D
1060 DATA 20,FD,C9,21,00,C0,06,
03
1070 DATA 11,03,00,ES,CD,19,BD,
CD
1080 DATA 47,92,E1,23,10,F5,C9,
AF
1090 DATA 77,CD,2D,92,19,DB,1B,
F7
1100 *

```

```

1110 MEMORY &3FFF
1120 FOR i=&9200 TO &9200+79
1130 READ a$:POKE i,VAL("&"a$)
1140 NEXT i
1150 *
1160 MODE 1:PRINT"BILDSCHIRM-TO
OLBOX"
1170 PRINT"CALL &9200 = ABSPEIC
HERN EINES BILDES"
1180 PRINT"CALL &920C = LADEN E
INES BILDES"
1190 PRINT"CALL &9233 = LOESCHE
N EINES BILDES"
1200 PRINT"          V I E L S P
A S S "
1210 PRINT:PRINT"BITTE TASTE FU
ER DEMO DRUECKEN"
1220 CALL &BB18
1230 CLS:FOR i=400 TO 0 STEP-2
1240 MOVE 0,i:DRAW 640,i,1
1250 NEXT i
1260 FOR i=350 TO 50 STEP-2
1270 MOVE 50,i:DRAW 590,i,2
1280 NEXT i
1290 FOR i=300 TO 100 STEP-4
1300 MOVE 150,i:DRAW 490,i,3
1310 NEXT i
1320 CALL &9200
1330 FOR i=1 TO 500:NEXT i
1340 CALL &9233
1350 FOR i=1 TO 500:NEXT i
1360 CALL &920C
1370 FOR i=1 TO 500:NEXT i
1380 CALL &9233
1390 END
ENDE DES LISTINGS

```



Dieses Programm ermöglicht verschiedene Bildschirmeffekte.

Mit CALL&9200 speichert man den aktuellen Bildschirm im Bereich ab & 4000 ab.

Mit CALL&920C bringt man diesen Bereich wieder auf den Bildschirm. Dies geschieht in 3 Durchgängen, wodurch sich ein interessanter Effekt ergibt.

Mit CALL&9233 wird der Bildschirm in 3 Durchgängen gelöscht.

Spiel, Utility & Erweiterung

SUPRA-BASIC

erweitert den Schneider Befehlsatz um 32 neue Befehle, hier in alphabetischer Reihenfolge:

BANK, BIGOFF, BIGON, BYE, CALM, COPY, CRUNCH, DISPLAY, DOKE, ESY, EX, FAST, FIELD, HELP, HIDE, INV, LFIN, LFOUT, LTURN, NI, NOISE, PEY, PICO, QUAD, RTURN, SHOW, SPRITE, STRETCH, SYM, TRAP, WOWN, WUP.
Allgemein: ****SUPRA-BASIC**** wurde für den CPC 464 geschrieben. Die Befehlsweiterung ist so angelegt, daß sie auch auf dem CPC 464 mit angeschlossenem Diskettenlaufwerk voll funktionsfähig ist, wobei man bei den Kassettenbefehlen **FAST, LFIN** und **LFOUT** natürlich Einschränkungen machen muß: **FAST** erhöht zwar die Schreibgeschwindigkeit auf 3 500 Baud, zur Beschleunigung des Diskettenlaufwerks ist der Befehl jedoch ungeeignet, ebenso wie sich **LFIN** und **LFOUT** nur auf Kassettenfiles beziehen.

****SUPRA-BASIC**** belegt den Speicher von & A 000 bis & A 520. Da es jedoch möglich ist, Sprites und Pictures zu definieren, kann der Speicherplatz nochmals um 4 KByte verkleinert werden, weil die **Picture-DATAs** von & 9000 bis & 97FF und die **Sprite-DATAs** von & 9800 bis & 9FFF abgelegt werden. Es ist also das Kommando **MEMORY & 8FFF** erforderlich, um die **DATAs** nicht zu überschreiten; arbeitet man jedoch ohne Sprites und Pictures, genügt natürlich **MEMORY & 9FFF**.

Einige Befehle arbeiten mit Variablen. Diese Variablen müssen vorher schon benutzt worden sein, weil sonst bei (Klammeraffe) z \$ (als Beispiel) ein **"Improper argument"** ausgegeben wird.

Fast alle Befehle haben eine festgelegte Anzahl von Operanden, wird hier ein Fehler gemacht, gibt das Programm ****SUPRA-BASIC**** die Fehlermeldung **"Operand mistake"** aus. Jedem Befehlswort muß der senkrechte Strich ("Shift" + (Klammeraffe)) vorangestellt werden, sonst erreicht man nur einen **Syntax error**.

Es empfiehlt sich, die Befehlsweiterung als binäre Datei abzuspeichern, um schnell darauf zurückgreifen zu können. Hierfür muß man eingeben:

```
POKE & A000,1 : SAVE
"SB.BIN", b, & A000, & 520
```

Man kann ****SUPRA-BASIC**** dann mit **LOAD "SB.BIN", & A000** schnell wieder laden, z. B. vom Programm aus, wie es ja auch bei **MADGOON'S RACE** geschieht. Überhaupt ist dieses Demo-Programm die beste Möglichkeit, die Wirkung der Befehle zu illustrieren. Die Spielidee selbst ist natürlich uralt, sie ist von **"FROGGER"** übernommen, das Männchen darf auf seinem Weg zum oberen Bildrand Nichts berühren, was sich bewegt. Die Musik kann mit der Leertaste an- und ausgeschaltet werden, wonach das Programm ein wenig schneller abläuft, da die **SOUND-Routine** doch ein bisschen Zeit in Anspruch nimmt, das Spielgeschehen aber nicht direkt beeinflusst. Das Programm hat eine eigene **TOP-TEN** Verwaltung, die auf Diskettenbetrieb ausgelegt ist. Die nötige Änderung für die Kassette ist jedoch kein Problem. Man läßt in Zeile 2520 alles bis zur **FOR-NEXT** Schleife weg und setzt in den Zeilen 1080 und 2500 an die Stelle von **"topten.fil"** nur **"ltopten.fil"**. Beim Abspeichern der neuen **TOP-TEN** muß die Kassette natürlich wieder an die rich-

tige Stelle zurückgespult worden sein.

Die Version von **"MAGDON'S RACE"** auf der beigelegten Kassette ist so nicht lauffähig, da auf dem Band weder die Binärform von ****SUPRA-BASIC**** noch die **TOP-TEN-FILE** vorhanden ist. Überspielen Sie das Programm bitte auf Diskette, speichern Sie wie beschrieben ****SUPRA-BASIC**** als Binärdatei ab und **SAVEN** Sie dann die **TOP-TEN** wie folgt:

```
FOR L = 1 TO 10: nam $(1) =
"SUPRA-SOFT": pu (1) = 0: day
$(1) = "01.01.1986": NEXT:
OPENOUT "topten.fil": FOR l
= 1 TO 10: ? # 9, nam $(1): ? #
9, pu (1): ? # 9, day $(1): NEXT:
CLOSEOUT.
```

Erläuterungen der Befehle:

BANK, wert

Dieser Befehl wählt die angegebene Bildschirmbank (1 oder 3). Ist der übergebene Wert 1, wird Bank 1 gewählt (& 4000 - & 7FFF), bei allen anderen Werten Bank 3. Will man beide Bildschirmhalte gleichzeitig benutzen (siehe auch **COPY** und **DISPLAY**), empfiehlt es sich, anfangs bei jeder Bildschirmbank ein **MODE-Kommando** zu setzen, um den eventuell verschobenen Offset auf den Ausgangswert zu bringen.

BIGOFF, BIGON

siehe unter **SPRITE**.

BYE

Nach diesem Kommando ist die Befehlskennung durch **RSX** unmöglich, die Maschinenroutinen sind jedoch noch vorhanden (ab & A104).

CALM

Erzeugt eine Pause beim **SOUND**. Die unterbrochene Melodie kann jedoch mit **NOISE** lückelos fortgesetzt werden.

COPY

Kopiert den Inhalt von Block 3, also dem Bildschirmspeicher von & C000 - & FFFF, in Block 1 (von & 4000 - & 7FFF).

CRUNCH

Dieser Befehl ist an sich nur während des Programmierens nützlich. Sobald die Zeileneingabe

durch **>ENTER<** beendet wird, werden alle überflüssigen Leerzeichen in dieser Zeile entfernt, wenn vorher **CHRUNCH** eingegeben wurde. Das nochmalige Verwenden des Befehls hebt den Effekt wieder auf.

DISPLAY

Das Gegenstück von **COPY**. Der Inhalt von Block 1 wird in den Block 3 kopiert.

DOKE, adresse, zwei-byte-zahl

Poked Zwei-Byte-Zahlen in den Speicher. Das Lowbyte wird an **>adresse<**, das Highbyte an **>adresse + 1<** abgelegt, also gebrauchsfertig für ein **MC-Programm**.

ESY, reihe von picturenummern

Siehe unter **SYM**.

EX, status (Klammeraffe) variable 1, (Klammeraffe) variable 2

Vertauscht die beiden Variablen, d. h. deren Inhalt. Der Status gibt an, um welchen Variablentyp es sich handelt (2 = **INTEGER**, 3 = **STRING**, 5 = **REAL**).

FAST

Dieser Befehl erhöht die Schreibgeschwindigkeit auf die Kassette auf 3500 Band.

FIELD, spalt, zeile, (Klammeraffe) integervariable

Der **ASCII-Code** des Zeichens an der übergebenen Cursorposition wird in der Variablen von **TYP INTEGER** gespeichert. Dieser Befehl gibt dem User also die bisher fehlende Möglichkeit, den Bildschirm abzufragen.

HELP

Ein nützlicher Befehl, wenn der Computer verrückt spielen sollte. **HELP** bewirkt folgendes:
MODE 1: BORDER 1: PAPER 0: PEN 1: CLS,
alle **INKs** werden wieder initialisiert, ebenso die komplette Tastatur. **SPEED INK** und **SPEED KEY** werden auf die Anfangswerte gesetzt, jeder **Break-Schutz**

programme

wird aufgehoben, sowie jeder SOUND endgültig abgewürgt. Ein völliges Reset durch CTRL+SHIFT+ESC ist wieder möglich. Programm und Variablen werden von HELP beeinflusst.

HIDE

Hat man ein Programm im Speicher, das man nicht zerstören will, möchte aber zwischendurch ein anderes bearbeiten (z. B. ein Spiel), gibt man HIDE ein und lädt das zweite Programm. Ist man damit fertig, braucht man, um an das alte Programm zu kommen, nur SHOW einzugeben. Das eben gespielte Spiel ist verschwunden - das alte Programm ist wieder im Speicher!!!

INV

Vertauscht PAPER und PEN. Die gleiche Wirkung hätte z. B. PRINT CHR 4(24).

LFIN, (Klammeraffee) stringvariable

Der Name des letzten Programmes, das von Kassette geladen wurde, wird in der Stringvariablen gespeichert oder auf den Bildschirm ausgegeben, wenn man nur LFIN eintippt.

LFOUT, (Klammeraffee) stringvariable

Wie LFIN, nur daß der Befehl ich auf das letzte auf Band gesavete Programm bezieht.

LTURN, reihe von Zeilennummern

Die angegebenen Zeilen werden um ein Byte nach links gescrollt.

NI, sekundenzahl

Der Interpreter wartet die angegebene Anzahl von Sekunden, ehe er mit dem Programm weitermacht. Während der Wartezeit (max. 255 Sekunden) ist der Rechner nicht beeinflussbar.

NOISE

Siehe unter CALM.

PEY

Dieses Kommando hält den Interpreter so lange an, bis irgendeine Taste gedrückt wird.

PICO, spalte, zeile, pen, picturenummer

Siehe unter SYM.

QUAD, links, rechts, oben, unten, wert

Zeichnet Rechtecke in der übergebenen Farbe (Die Farbtabelle ist im Programm angegeben). Die ersten vier Parameterwerte beziehen sich auf Cursorpositionen. Beispiele: Im Mode 0 ergibt QUAD, 1,20, 1, 2, 240, das die beiden obersten Zeilen mit schwarz gefüllt werden, im Mode 1 ergibt QUAD, 21, 40, 13, 25, 255, daß das Bildschirmviertel rechts unten mit hellrot gefüllt wird.

RTURN, reihe von zeilennummern

Wie LTURN, die Zeilen werden nur nach rechts geschoben.

SHOW

Siehe unter HIDE.

SPRITE, spalte, zeile, spritenummer

"*SUPRA-BASIC*" ermöglicht die Handhabung von Sprites, d. i. Multicolorzeichen im Mode 0. Es ist möglich, mit den Befehlen BIGON und BIGOFF zwischen großen und kleinen Sprites zu wählen. Ein kleiner Sprite belegt ein Cursorfeld, ein großer 2 x 2 Cursorfelder. Die kleinen benötigen 32, die großen demnach viermal so viel, nämlich 128 Farbwerte (d. i. Bytes im Speicher). Der Platz für diese Farbdaten liegt zwischen &9800 und &9FFF, man kann also gleichzeitig bis zu 16 BIGSPRITES (Nummer 0 bis 15) oder maximal 64 (!) kleine Sprites (0-63) verwalten, beides ist natürlich kombinierbar. Beispiel für die Benutzung von SPRITE, BIGON und BIGOFF: Neben den Befehlsroutinen wird auch ein Beispielsprite in den Speicher gepoked. Meldet sich der CPC nach Programmende mit Ready, bitte MODE 0 eingeben und nach der erneuten Ready-Meldung >ENTER< drücken. Nach der Eingabe von SPRITE, 6, 1, 0 wird rechts neben dem READY der SPRITE ausgegeben.

Tippt man nun BIGOFF und anschließend ISPRITE, 13, 3, 2, so erscheint anstelle der 0 nun der linke untere Teil des BIGSPRITES. Nach IBIGNON: sprite, 19,24,0 erscheint ganz rechts unten wieder der große Sprite. Bei großen Sprites bezeichnet die übergebene Cursorposition immer den linken Teil des Sprites, bei kleinen direkt das Ausgabefeld.

STRETCH, spalte, zeile, zeichenzahl

Ab der angegebenen Position wird nach rechts gehend bei der übergebenen Anzahl von Cursorfeldern deren Inhalt auf doppelte Höhe gestreckt. Beispiel: Nach Programmende wie oben MODE 0 und nochmals >ENTER< eingeben, dann Istretch, 2, 1, 3 tippen, so werden bei Ready ab dem e (Position 2, 1) das e selbst, sowie a und d um den Faktor zwei gestreckt (3 Buchstaben). Es ist auch möglich, Sprites zu strecken! Deutlich wird das, wenn man eingibt IBIGEN: Isprite, 16, 1, 0: Istretch, 16, 2, 2. Der untere Teil des Sprites ist nun doppelt so hoch. Einen ganzen Sprite strecken Sie so (z. B.): MODE 0: Isprite, 19, 12, 0: Isprite, 19, 13, 0: Istretch, 19, 12, 2: Istretch, 19, 14, 2. In den beiden anderen Modi ist der Befehl umständlicher zu handhaben, der Effekt stellt sich natürlich auch ein, hauptsächlich wurden die Befehle für die graphischen Tricks allerdings für den MODE 0 konzipiert.

SYM, reihe von ascii-werten, picturenummer

"*SUPRA-BASIC*" erlaubt auch die Verwendung von sog. Pictures, was als Kette von ASCII-Zeichen zu verstehen ist (max. 32 pro Picture). Es können auch Zeichen mit einem ASCII-Code < 32 verwendet werden. Ein Picture wird über SYM definiert, mit PICO ausgegeben und muß mit ESY gelöscht werden, ehe an seiner Stelle ein neues erstellt werden kann. Beispiel: Die folgenden Basic-Ausdrücke:

```
a $ = CHR $ (255) + CHR $ (254) + CHR $ (8) + CHR $ (8) + CHR $ (10) + CHR $ (253) + CHR $ (252); PEN 2: LOCATE 21, 14: PRINT a $  
sehen mit Pictures so aus:  
ISYM, 255, 254, 8, 8, 10, 252, 0:  
IPKO, 21, 14, 2, 0
```

Das erzielt den gleichen Effekt! Als Picturenummer wurde hier 0 verwendet, man kann natürlich andere (0 bis 63) verwenden. Die Werte sind im Speicher von &9000 bis &97FF abgelegt. Soll Picture 0 neu definiert werden, muß zuerst ESY, 0 eingegeben werden, um die ursprünglichen Daten völlig zu löschen, bevor man das Picture neu definiert. ESY löscht auch mehrere Pictures, zum Beispiel ESY, 8, 15, 16, 33, 50.

TRAP, wert

Bei TRAP, 0 wird der Transparentmodus aus-, bei TRAP, 1 eingeschaltet.

WOWN, links, rechts, oben, unten, wert

Parameter wie bei QUAD. Das so definierte Textfenster wird nach unten gescrollt, der Paper der neuen Zeile entspricht dem letzten Wert. WOWN ist unabhängig von Basic-Windows!

WVP, links, rechts, oben, unten, wert

Wie bei WOWN, nur wird hier auch oben gescrollt.

Anleitung für den Spritegenerator

Nach dem Anfangsbild wird der Bildschirm schwarz, ein Rechteck wird gezeichnet, über welchem die Farbanzeige zu sehen ist - um Anfang ein dunkelblaues Rechteck (INK 0). Durch Drücken der Space-Taste kann die INK-Nummer erhöht werden. Wenn man z. B. jetzt drückt, erscheint hellgelb (INK 1), dann helles Blaugrün (INK 2) usw. Nach INK 15 wird wieder bei 0 begonnen. Die Farbe des Rechtecks ist die momentane Zeichenfarbe. Drückt man gleich zu Beginn die COPY-Taste, so wird das Cursorfeld links oben (wo der gelbe Punkt ist) blau. Steuert man den gelben Punkt mit den Cursorstasten auf das Feld rechts daneben, drückt dreimal die Leertaste, so daß die Zeichenfarbe auf hellrot wechselt und drückt dann COPY, wird ein rotes Feld gesetzt. Die beiden Punkte dieses BIGSPRITES wären links oben dann dunkelblau und hellrot. Man kann also einen großen oder vier kleine Sprites in diesem Spritegenerator erstellen. Die Farbwahl kann jedoch auch direkt erfolgen.

Um von hellrot wieder auf dunkelblau zu kommen, müßte man über zehnmal die Leertaste drücken. Man kann jedoch den kleinen gelben Punkt mit den Cursors auf das blaue Feld lenken und "W" drücken. Dunkelblau als Zeichenfarbe wird dann direkt gewählt. Ist der Sprite fertig, drückt man "E", es erscheint das Wort "Calculating". Das Maschinenprogramm berechnet aus den INK-Werten, die mit der Funktion TEST (X,Y) abgefragt und dann in den Speicher gepoked wurden, die 128 codierten Werte in ca. 0,1 Sekunden (die entsprechende Basic-Routine wäre ca. 200mal langsamer). Nach der Codierung wird der Sprite gezeigt. Das Programm fragt dann, ob man weitermachen oder die Arbeit beenden will. Drückt man "R" für RUN, wird man aufgefordert, dem Sprite einen Namen zu geben, unter dem dieser dann abgesaved wird. Das Programm erstellt vier DATA-Zeilen (Nummer 1-4), die als Datei gespeichert und mit dem Befehl MERGE "SPRITENAME.spr" in jedes Programm eingebunden werden können. Jede dieser Zeilen enthält 32 codierte Farbwerte.

Supra-Basic

```

1000 '-----
1010 '---- (c) September 1986
by '----
1020 '---- Alexander Stroh
-----
1030 '-----
1040 MEMORY &OFFF:CALL &BC02:DE
FINT a-z
1050 POKE &BDEE,&C9:BORDER 13:I
NK 0,13
1060 INK 2,0:MODE 1:PAPER 0:PEN
3:CLS
1070 PRINT:PRINT" Befehlserw
eiterung *SUPRA-BASIC*"
1080 PEN 2:PRINT
1090 PEN 1:LOCATE 16,4:PRINT CH
R$(164) 1986 by:"
1100 PEN 2:PRINT
1110 PRINT" ////////////////
////////////////////
1120 PRINT" | SUPRA
-SOFT |
1130 PRINT" |
|
1140 PEN 1:LOCATE 15,10:PRINT"-
-----
1150 LOCATE 15,11:PRINT"BITTE W
ARTEN"
1160 LOCATE 15,12:PRINT"-----
-----
1170 PRINT"
-----
1180 '-----
-----
1190 '----- MC-DATAS: ---
-----
1200 '-----
1210 RESTORE 1240
1220 FOR i=%A000 TO %A4D2:READ
a$
1230 a=VAL("&" + a$):POKE i,a:NEX
T
1240 DATA 01,0f,a0,21,00,a1,cd,
d1,bc,3e
1250 DATA c9,32,00,a0,c9,72,a0,
c3,04,a1
1260 DATA c3,51,a1,c3,63,a1,c3,

```

```

74,a1,c3
1270 DATA 85,a1,c3,b0,a1,c3,be,
a1,c3,d7
1280 DATA a1,c3,e7,a1,c3,f7,a1,
c3,1d,a2
1290 DATA c3,58,a2,c3,72,a2,c3,
a8,a2,c3
1300 DATA 17,a3,c3,37,a3,c3,4f,
a3,c3,73
1310 DATA a3,c3,9d,a3,c3,c0,a3,
c3,ea,a3
1320 DATA c3,14,a4,c3,30,a4,c3,
4c,a4,c3
1330 DATA 74,a4,c3,82,a4,c3,91,
a4,c3,9a
1340 DATA a4,c3,a9,a4,c3,bc,a4,
c3,c5,a4
1350 DATA c3,ce,a4,00
1360 DATA 53,50,52,49,54,c5,43,
52,55,4e
1370 DATA 43,c8,43,4f,50,d9,44,
49,53,50
1380 DATA 4c,41,d9,48,45,4c,d0,
46,41,53
1390 DATA d4,42,41,4e,cb,42,49,
47,4f,ce
1400 DATA 42,49,47,4f,46,c6,52,
54,55,52
1410 DATA ce,4c,54,55,52,ce,51,
55,41,c4
1420 DATA 42,59,c5,53,54,52,45,
54,43,c8
1430 DATA 46,49,45,4c,c4,45,d8,
53,59,cd
1440 DATA 50,49,43,cf,45,53,d9,
4c,46,49
1450 DATA ce,4c,46,4f,55,d4,57,
4f,57,ce
1460 DATA 57,55,d0,48,49,44,c5,
53,48,4f
1470 DATA d7,54,52,41,d0,49,4e,
d6,44,4f
1480 DATA 4b,c5,4e,c9,50,45,d9,
43,41,4c
1490 DATA cd,4e,4f,49,53,c5,0,0,
0,0,0,0
1500 DATA 00,00,00,00,fe,03,c2,
00,a5,eb
1510 DATA 06,07,29,10,fd,eb,21,
00,98,19
1520 DATA eb,dd,66,04,dd,6e,02,
25,2d,c3
1530 DATA 3a,a1,cd,1a,bc,0e,08,
06,03,1a
1540 DATA 77,13,23,10,fa,1a,77,
13,d5,11
1550 DATA fd,07,19,d1,0d,20,ec,
c9,e5,cd
1560 DATA 20,a1,e1,24,e5,cd,20,
a1,e1,2c
1570 DATA 25,e5,cd,20,a1,e1,24,
cd,20,a1
1580 DATA c9,fe,00,c2,00,a5,21,
00,ac,af
1590 DATA be,28,03,36,00,c9,36,
01,c9,fe
1600 DATA 00,c2,00,a5,21,00,c0,
11,00,40
1610 DATA 01,00,40,ed,b0,c9,fe,
00,c2,00
1620 DATA a5,21,00,40,11,00,c0,
01,00,40
1630 DATA ed,b0,c9,fe,00,c2,00,
a5,3e,01
1640 DATA cd,0e,bc,cd,02,bc,af,
cd,96,bb
1650 DATA 3e,01,cd,90,bb,cd,14,
bc,cd,00
1660 DATA bb,cd,03,bb,cd,b6,bc,
cd,45,bb
1670 DATA 3e,c3,32,ee,bd,c9,fe,
00,c2,00
1680 DATA a5,3e,0b,21,5f,00,cd,
68,bc,c9
1690 DATA fe,01,c2,00,a5,3e,01,
bb,28,04

```

```

(HN) 1700 DATA 3e,c0,18,02,3e,40,47,
cd,06,bc (MH)
(BD) 1710 DATA 78,32,cb,b1,c9,fe,00,
c2,00,a5 (BD)
(HD) 1720 DATA 3e,07,32,0b,a1,3e,3a,
32,1e,a1 (OL)
(FK) 1730 DATA c9,fe,00,c2,00,a5,3e,
05,32,0b (DJ)
(BK) 1740 DATA a1,3e,20,32,1e,a1,c9,
fe,00,ca (DI)
(EL) 1750 DATA 00,a5,f5,cd,43,a2,11,
4f,c0,19 (BE)
(ED) 1760 DATA 06,08,c5,e5,d1,2b,01,
4f,00,1a (DC)
(FJ) 1770 DATA ed,b8,12,01,50,08,09,
c1,10,ee (DD)
(JD) 1780 DATA f1,3d,20,e0,c9,fe,00,
ca,00,a5 (IO)
(FH) 1790 DATA f5,cd,43,a2,11,00,c0,
19,06,08 (OM)
(KH) 1800 DATA c5,e5,d1,23,01,4f,00,
1a,ed,b0 (LL)
(OC) 1810 DATA 12,01,b0,07,09,c1,10,
ee,f1,3d (BN)
(JA) 1820 DATA 20,e0,c9,dd,6e,00,2d,
dd,23,dd (AM)
(JI) 1830 DATA 23,af,67,06,04,29,10,
fd,54,5d (ID)
(CD) 1840 DATA 29,29,19,c9,fe,05,c2,
00,a5,7b (PH)
(CD) 1850 DATA dd,5e,02,dd,6e,04,dd,
56,06,dd (FA)
(DN) 1860 DATA 66,08,1d,15,25,2d,cd,
44,bc,c9 (IH)
(IO) 1870 DATA fe,00,c2,00,a5,cd,8a,
a1,cd,88 (FN)
(HJ) 1880 DATA a2,af,06,ff,21,00,a0,
77,23,10 (IP)
(DJ) 1890 DATA fc,c9,21,9c,a2,06,0c,
7e,cd,5a (DJ)
(KE) 1900 DATA bb,23,10,f9,26,01,2e,
03,cd,75 (DP)
(IC) 1910 DATA bb,c9,20,42,41,53,49,
43,20,31 (MM)
(FA) 1920 DATA 2e,30,0d,0a,fe,03,c2,
00,e5,7b (GH)
(OB) 1930 DATA 32,76,a5,dd,6e,02,dd,
66,04,25 (BB)
(OC) 1940 DATA 2d,e5,cd,1a,bc,11,00,
38,19,22 (DH)
(EF) 1950 DATA 72,a5,11,50,00,e5,19,
d1,3e,02 (IN)
(CI) 1960 DATA 32,74,a5,dd,21,74,a5,
0e,04,06 (CF)
(CF) 1970 DATA 04,1a,77,c5,01,00,08,
ed,42,77 (OE)
(OE) 1980 DATA ed,4a,c1,23,13,10,f0,
d5,11,04 (DM)
(DM) 1990 DATA 10,ed,52,d1,eb,c5,01,
00,08,ed (KA)
(KA) 2000 DATA 42,c1,2b,2b,2b,2b,eb,
0d,20,d7 (DC)
(DC) 2010 DATA 2a,72,a5,11,00,20,ed,
52,eb,2a (FK)
(FK) 2020 DATA 72,a5,dd,35,00,20,c4,
e1,24,dd (OH)
(OH) 2030 DATA 35,02,20,a3,c9,fe,03,
c2,00,a5 (OA)
(OA) 2040 DATA cd,78,bb,e5,dd,6e,02,
dd,66,04 (MK)
(MK) 2050 DATA cd,75,bb,cd,60,bb,30,
02,eb,77 (NK)
(NK) 2060 DATA e1,26,00,cd,75,bb,c9,
fe,03,c2 (CM)
(CM) 2070 DATA 00,a5,dd,66,03,dd,6e,
02,dd,46 (MJ)
(MJ) 2080 DATA 04,4e,1a,77,79,12,13,
23,10,f7 (ML)
(ML) 2090 DATA c9,57,3e,20,ba,fa,00,
a5,af,67 (OL)
(OL) 2100 DATA 6b,06,05,29,10,fd,42,
11,1f,90 (PJ)
(PJ) 2110 DATA 19,05,00,dd,7e,02,77,
2b,dd,23 (KH)
(KH) 2120 DATA dd,23,10,f5,af,77,c9,
fe,04,c2 (FB)
(FB)

```


programme

2130 DATA 00,a5,dd,66,06,dd,6e,04,cd,75	(OE)	2580 '-----	rminhalt in Block 1.":PEN 3	(ED)
2140 DATA bb,dd,7e,02,cd,90,bb,af,67,6b	(KN)	2590 RESTORE 2620	3010 PRINT" ICRUNCH":PEN 1	(PA)
2150 DATA 06,05,29,10,fd,11,00,90,19,06	(HK)	2600 FOR i=&9800 TO &987F:READ a	3020 PRINT" Schaltet Komprimiermodus ein/aus.":PRINT:PEN 3	(BK)
2160 DATA 20,7e,cd,5a,bb,23,10,f9,c9,fe	(BA)	2610 POKE i,a:NEXT a	3030 PRINT" IDISPLAY":PEN 1	(FP)
2170 DATA 00,ca,00,a5,f5,af,67,dd,6e,00	(DC)	2620 DATA 0,0,0,0,0,0,17,51,0,0,51,51,0	3040 PRINT" Zeigt Inhalt von Block 1.":PEN 3:PRINT	(DL)
2180 DATA 06,05,29,10,fd,11,00,90,19,06	(CP)	2630 DATA 17,51,51,17,51,204,153,51,51	3050 PRINT" IDOKE,adresse,wert":PEN 1	(IA)
2190 DATA 20,77,23,10,fc,dd,23,dd,23,f1	(ND)	2640 DATA 152,153,51,51,152,153,51,51	3060 PRINT" POKet die 2-Byte-Zahl in die beiden Speicherzellen ab dem ersten Wert.":PEN 3	(JI)
2200 DATA 3d,20,e3,c9,fe,02,f2,00,a5,fe	(PH)	2650 DATA 204,153,0,0,0,0,51,34,0,0,51	3070 PRINT" IESY,picturenummern":PEN 1	(FB)
2210 DATA 01,20,14,3e,10,12,eb,23,5e,23	(PC)	2660 DATA 51,0,0,51,51,34,0,102,204,51	3080 PRINT" Loescht die angegebenen PICTURES."	(NI)
2220 DATA 56,21,07,b8,06,10,7e,12,23,13	(ED)	2670 DATA 34,102,100,51,51,102,100,51	3090 GOSUB 3830	(AO)
2230 DATA 10,fa,c9,21,07,b8,06,10,7e,cd	(BB)	2680 DATA 51,102,204,51,51,51,54,51,51	3100 LOCATE 14,1:PRINT"-----"	(IB)
2240 DATA 5a,bb,23,10,f9,c9,fe,02,f2,00	(IB)	2690 DATA 17,51,60,60,0,51,54,60,0,240	-----":LOCATE 14,2:PRINT"-----"	(IB)
2250 DATA a5,fe,01,20,14,3e,10,12,eb,23	(JE)	2700 DATA 51,51,0,240,0,51,80,160,0,0	TAX (3) --"	(AD)
2260 DATA 5e,23,56,21,4c,b8,06,10,7e,12	(PC)	2710 DATA 80,160,0,0,80,160,0,0,51,51	3110 LOCATE 14,3:PRINT"-----"	(AD)
2270 DATA 23,13,10,fa,c9,21,4c,b8,06,10	(HB)	2720 DATA 57,51,60,60,51,34,60,57,51,0	-----":PRINT:PEN 3	(AD)
2280 DATA 7e,cd,5a,bb,23,10,f9,c9,fe,05	(DC)	2730 DATA 51,51,240,0,51,0,240,0,0,0,80	3120 PRINT" IEX,status,@variable1,@variable2":PEN 1	(EJ)
2290 DATA c2,00,a5,06,00,7b,dd,5e,02,dd	(GL)	2740 DATA 160,0,0,80,160,0,0,80,160	3130 PRINT" Vertauscht beide Variablen.Status: 2=INT,3=STRING,5=REAL":PEN 3:PRINT	(PM)
2300 DATA 6e,04,dd,56,06,dd,66,08,1d,15	(HI)	2750 '-----	3140 PRINT" IFAST":PEN 1	(PB)
2310 DATA 25,2d,cd,50,bc,c9,fe,05,c2,00	(JB)	2760 '----- RSX-EINBINDUNG UND	3150 PRINT" Setzt SAVEspeed auf 3500 Baud.":PEN 3:PRINT	(LP)
2320 DATA a5,06,01,7b,dd,5e,02,dd,6e,04	(HD)	2770 '----- BEFEHLSUEBERSICHT:	3160 PRINT" IFIELD,spalte,zeile,@integervariable":PEN 1	(EF)
2330 DATA dd,56,06,dd,66,08,1d,15,25,2d	(BL)	2780 '-----	3170 PRINT" Der ASCII-Code des Zeichens an der Cursorposition wird in der Variablen gespeichert.":PEN 3:PRINT	(PD)
2340 DATA cd,50,bc,c9,fe,00,c2,00,a5,21	(IH)	2790 CALL &A000	3180 PRINT" IHELP":PEN 1	(MD)
2350 DATA 81,ae,11,83,ae,1a,77,13,23,1a	(JF)	2800 PEN 3:PRINT:PRINT	3190 PRINT" Graphik-,SOUND- und Tastaturreset. In diesen Bereichen wirkt der Befehl wie in CALL 0, ohne jedoch Programm und Variablen zu beeinflussen.":GOSUB 3830	(PD)
2360 DATA 77,ed,5b,81,ae,2a,7b,ae,ed,52	(DM)	2810 PRINT" -----"	3200 LOCATE 14,1:PRINT"-----"	(PD)
2370 DATA eb,2a,7b,ae,af,77,42,4b,e5,d1	(LL)	2820 PRINT" Die Befehle koennen nun gelistet werden. [*SB* Syntax-Erklaerung]"	-----":LOCATE 14,2:PRINT"-----"	(DN)
2380 DATA 1b,ed,b8,c9,fe,00,c2,00,a5,21	(JO)	2830 PRINT" -----":PRINT:GOSUB 3830	TAX (4) --"	(DN)
2390 DATA 81,ae,36,6f,23,36,01,c9,fe,01	(EH)	B30	3210 LOCATE 14,3:PRINT"-----"	(EB)
2400 DATA c2,00,a5,3e,16,cd,5a,bb,7b,cd	(OH)	2840 LOCATE 14,1:PRINT"-----"	-----":PRINT:PEN 3	(NN)
2410 DATA 5a,bb,c9,fe,00,c2,00,a5,cd,9c	(AI)	-----":LOCATE 14,2:PRINT"-----"	3220 PRINT" IHide":PEN 1	(NN)
2420 DATA bb,c9,fe,02,c2,00,a5,dd,66,03	(HL)	TAX (1) --"	3230 PRINT" Macht ein Programm fuer den Interpreter unsichtbar, so dass ein zweites geladen und bearbeitet werden kann, ohne dass das alte geloescht wird.":PEN 3:PRINT	(AB)
2430 DATA dd,6e,02,73,23,72,c9,fe,01,c2	(FB)	2850 LOCATE 14,3:PRINT"-----"	3240 PRINT" IINV":PEN 1	(AJ)
2440 DATA 00,a5,7b,87,47,21,40,fe,2b,7c	(BB)	-----":PRINT:PEN 3	3250 PRINT" Vertauscht PAPER und PEN.":PEN 3:PRINT	(CJ)
2450 DATA b5,20,fb,10,f6,c9,fe,00,c2,00	(BP)	2860 PRINT" IBANK,bildschirmbank":PEN 1	3260 PRINT" ILFIN,@stringvariable":PEN 1	(IJ)
2460 DATA a5,cd,18,bb,c9,fe,00,c2,00,a5	(LI)	2870 PRINT" Wechselt die jeweilige Bildschirmbank. Ist der uebergebene Wert 1, wird Bank 1 gewaehlt, andernfalls Bank 3.":PEN 3:PRINT	3270 PRINT" Weist der Variablen den Namen der letzten LOAD-File zu. Ohne Variable wird er direkt auf den Bildschirm ausgegeben.":PEN 3:PRINT	(MB)
2470 DATA cd,b6,bc,c9,fe,00,c2,00,a5,cd	(LI)	2880 PRINT" IBIGOFF":PEN 1	3280 PEN 3:PRINT	(ED)
2480 DATA b9,bc,c9,00	(FD)	2890 PRINT" Schaltet die Ausgabe von grossen Sprites (2*2 Cursorfelder) ab. Der ausgegebene Sprite belegt ein Cursorfeld.":PEN 3:PRINT	3290 PRINT" ILFOUT,@stringvariable":PEN 1	(KN)
2490 RESTORE 2520	(BF)	2900 PRINT" IBIGON":PEN 1	3300 PRINT" Siehe ILFIN. Beide Befehle gelten nur fuer Cassetten-Files."	(II)
2500 FOR i=&A500 TO &A51E:READ a#	(HJ)	2910 PRINT" Ermoeslicht wieder die Ausgabe von grossen Sprites.":PEN 3:PRINT	3310 GOSUB 3830	(OD)
2510 a=VAL("&"+a#):POKE i,a:NEXT i	(GH)	2920 PRINT" IBYE":PEN 1	3320 LOCATE 14,1:PRINT"-----"	(OD)
2520 DATA 21,0d,a5,7e,fe,00,c8,cd,5a,bb	(ED)	2930 PRINT" *SUPRA-BASIC* wird wieder ausgeklintet."	-----":LOCATE 14,2:PRINT"-----"	(JP)
2530 DATA 23,18,f6,4f,70,65,72,61,6e,64	(NP)	2940 GOSUB 3830	TAX (5) --"	(AF)
2540 DATA 20,6d,69,73,74,61,6b,65,0a,0d	(DB)	2950 LOCATE 14,1:PRINT"-----"	3330 LOCATE 14,3:PRINT"-----"	(AD)
2550 DATA 00	(BP)	-----":LOCATE 14,2:PRINT"-----"	-----":PRINT:PEN 3	(AD)
2560 '-----	(DD)	TAX (2) --"	3340 PRINT" ITURN,zeilennummer a":PEN 1	(GB)
2570 '----- BSprite-DATAs	(EA)	2960 LOCATE 14,3:PRINT"-----"	3350 PRINT" Scrollt die angegebenen Zeilen um ein Byte nach links.":PEN 3:PRINT	(HB)
		-----":PRINT:PEN 3	3360 PRINT" INI,sekunden":PEN 1	(ED)
		2970 PRINT" ICALM":PEN 1	3370 PRINT" Der Interpreter wartet die angegebene Anzahl von Sekunden (bis 255).":PEN 3:PRINT	
		2980 PRINT" Erzeugt eine SOUND-Pause.":PEN 3:PRINT		
		2990 PRINT" ICOPY":PEN 1		
		3000 PRINT" Kopiert den Bidschi		


```

NT
3380 PRINT " INDISE";PEN 1
3390 PRINT " Beendet die durch I
CALM erzeugte Pause.":PEN 3
3400 PRINT " IPEY";PEN 1
3410 PRINT " Wartet auf das Drue
cken einer beliebigen Taste.
":PEN 3:PRINT
3420 PRINT " IPICD,spalte,zeile,
pen,picturenummer":PEN 1
3430 PRINT " Ausgabe des jeweili
gen PICTURES an der Cursorposi
tion mit dem angegebenen PEN."
3440 LOCATE 1,23:GOSUB 3830
3450 LOCATE 14,1:PRINT"-----
":LOCATE 14,2:PRINT"-- SYN
TAX (6) --"
3460 LOCATE 14,3:PRINT"-----
":PRINT:PEN 3
3470 PRINT " IQUAD,links,rechts,
oben,unten,wert":PEN 1
3480 PRINT " Fuehlt die durch di
e Parameter begrenzte Flaechen
(Cursorwerte) mit der Farbe, d
ie dem Parameter wert entspricht.
"
3490 PEN 2:PRINT:PRINT" Dabei g
ilt im MODE 0 (wert/Farbe)":PE
N 1
3500 PRINT " 000/01 003/13 012
/20 015/14 048/26 051/18 06
0/02 063/24,1 192/24 1
95/12 204/06 207/16 240/00
243/22 252/08 255/16,2"
3510 PEN 2:PRINT:PRINT" MODE 1:
":PEN 1
3520 PRINT " 000/01 015/20 240
/24 255/06"
3530 PEN 2:PRINT:PRINT" MODE 2:
":PEN 1:PRINT" 000/01 255/24"
:PEN 3:PRINT
3540 PRINT " IRETURN,zeilennummer
n":PEN 1
3550 PRINT " Scrollt die angegeb
enen Zeilen um ein Byte nach
rechts."
3560 GOSUB 3830
3570 LOCATE 14,1:PRINT"-----
":LOCATE 14,2:PRINT"-- SYN
TAX (7) --"
3580 LOCATE 14,3:PRINT"-----
":PRINT:PEN 3
3590 PRINT " ISHOW":PEN 1
3600 PRINT " Macht ein durch IHI
DE 'verstecktes' Programm f
uer den Interpreter wieder b
earbeitbar."
3610 PEN 3:PRINT:PRINT " ISPRITE
,spalte,zeile,spritenummer":PEN
1
3620 PRINT " Sprite spritenummer
wird an der Cur- position a
usgegeben.Die DATAs sind ab A
dresse &9B00 abgelegt.":PEN 3:P
RINT
3630 PRINT " ISTRETCH,spalte,zei
le,breite":PEN 1
3640 PRINT " Ab der Cursorpositi
on wird die durch breite bes
timte Anzahl 4-Byte-Zeichen (
MODE 0) um den Faktor 2 gestrec
kt.":PEN 3:PRINT
3650 PRINT " ISYM,parametierreihe
,picturenummer":PEN 1
3660 PRINT " Definiert das angeg
ebene PICTURE durch die ASCII-
Codes der Einzelzeichen."
3670 GOSUB 3830
3680 LOCATE 14,1:PRINT"-----
":LOCATE 14,2:PRINT"-- SYN
TAX (8) --"
3690 LOCATE 14,3:PRINT"-----
":PRINT:PEN 3
3700 PRINT " ITRAP,wert":PEN 1
3710 PRINT " Schaltet den Transp
arentmodus bei 1 ein und be

```

```

(BH) i 0 aus."
(HO) 3720 PEN 3:PRINT
3730 PRINT " IOWN,links,rechts,
oben,unten,farbe":PEN 1
(FP) 3740 PRINT" Parameter wie IQUAD
(CCE) .Das WINDOW wird ab- waerts ges
crollt,der PAPER der neuen Z
eile entspricht dem Farbwert. IW
OWN ist unabhaengig von BASIC-W
INDOWs."
(PJ) 3750 PEN 3:PRINT:PRINT" IWUP,li
(IC) nks,rechts,oben,unten,farbe":PE
N 1
(GB) 3760 PRINT " Wie bei IOWN,nur a
(DH) ufwaerts.":PRINT:GOSUB 3830
3770 LOCATE 14,12:PRINT"-----
":LOCATE 14,13:PRINT"Noc

```

```

(PC) hmal? (J/N)"
(HC) 3780 LOCATE 14,14:PRINT"-----
":WHILE INKEY$<>"":WEND
(NH)
(JN) 3790 IF INKEY(45)>-1 THEN PEN 2
:CLS:GOTO 2840
(FO) 3800 IF INKEY(46)>-1 THENIHELP:
(CD) PRINT" BASIC 2.0":PRINT:NEW
(PM) 3810 GOTO 3790
(AO) 3820
(EJ) 3830 PRINT:PEN 2:PRINT" <<<<
<<<<<<<< Taste; >>>>>>>>>>>>"
(GB) 3840 WHILE INKEY$<>"":WEND:IPEY
:CLS:RETURN
(HG)
(DJ) ENDE DES LISTINGS

```

Supra-Basic

Das Utility:

Sprite-Generator

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

1000 '-----
-----
1010 MEMORY &9FFF:MODE 1:PAPER
0:PEN 1:INK 1,1:BORDER 1:CLS
1020 LOCATE 1,2:PRINT STRING$(4
0,"*")
1030 LOCATE 1,3:PRINT"* SUP
E R S P R I T E -- GENERATOR *
1040 LOCATE 1,4:PRINT STRING$(4
0,"*")
1050 LOCATE 1,9
1060 PRINT " Cursor bewegen-----
CURSBORTASTEN"
1070 PRINT:PRINT" Farbe waehle
n-----SPACETASTE"
1080 PRINT:PRINT" Farbe direkt
waehlen--->WK druecken"
1090 PRINT:PRINT" Farbpunkt se
tzen-----COPYTASTE"
1100 PRINT:PRINT" Ende-----
>EK druecken"
1110 LOCATE 1,22:PRINT STRING$(
40,"*")
1120 LOCATE 1,23:PRINT"* (c)
1986 by ALEXANDER STROH
"
1130 LOCATE 1,24:PRINT STRING$(
40,"*")
1140 INK 1,24:DIM d$(4)
1150 FOR I=&A000 TO &A400:POKE
1,0:NEXT
1160 '-----
-----
1170 RESTORE 1200:FOR n=0 TO 36
1180 READ i$:i$="&"+i$:POKE &A1
00+n,VAL(i$)
1190 NEXT
1200 DATA cd,78,bb,25,2d,cd,1a,
bc,14,8
1210 DATA 1,30,a1,a,77,3,1e,3,2
3,a,77,3,1d,c2,12,a1
1220 DATA c5,1,fd,7,9,ci,15,c2,
d,a1,c9
1230 '-----
-----
1240 RESTORE 1280

```

```

1250 READ i$:IF i$="-1"THEN 138
0
1260 i$="&"+i$:POKE &A000+te,VA
L(i$):te=te+1
1270 GOTO 1250
1280 DATA 3a,c0,a0,32,c2,a0,3c,
32,c4,a0,32,c6,a0,3a,c1,a0
1290 DATA 32,c5,a0,3c,32,c3,a0,
32,c7,a0
1300 DATA dd,21,b,a1,2a,c0,a0,d
d,36,0,30,cd,75,bb,cd,0,a1,2a,c
2,a0,dd,36,0,50
1310 DATA cd,75,bb,cd,0,a1,2a,c
4,a0,dd,36,0,70,cd,75,bb,cd,0,a
1,2a,c6,a0
1320 DATA dd,36,0,90,cd,75,bb,c
d,0,a1,c9,0,0,0,0,0,0,0,0,0,0,
0,0
1330 DATA 21,00,a2,11,30,a1,06,
80,cb,7e,ca,71,a0,eb,cb,ce,eb,c
b,76,ca,7a,a0
1340 DATA eb,cb,ee,eb,cb,6e,ca,
83,a0,eb,cb,de,eb,cb,66,ca,8c,a
0,eb,cb,fe,eb
1350 DATA cb,5e,ca,95,a0,eb,cb,
c6,eb,cb,56,ca,9e,a0,eb,cb,e6,e
b,cb,4e,ca,a7
1360 DATA a0,eb,cb,d6,eb,cb,46,
ca,b0,a0,eb,cb,f6,eb,23,13,05,c
2,68,a0,c9
1370 DATA -1
1380 '-----
-----
1390 MODE 0:PAPER 5:BORDER 0
1400 CLS
1410 INK 15,13:SYMBOL 255,255,2
55,255,255,255,255,254
1420 MOVE 60,352:DRAWR 516,0,15
:DRAWR 0,-258,15:DRAWR-516,0,15
:DRAWR 0,258,15
1430 PEN 15:LOCATE 7,2:PRINT"FA
RBE: "
1440 PEN 0:LOCATE 14,2:PRINT CH
R$(143)
1450 tint=0:xp=92:yp=336:xp1=9
2:yp1=336
1460 in$=INKEY$
1470 IF in$="p"THEN yp=yp+16
1480 IF in$="q"THEN yp=yp-16
1490 IF in$="r"THEN xp=xp-32
1500 IF in$="e"THEN xp=xp+32

```


programme

```

1510 IF xp<92 OR xp>572 OR yp>3
36 OR yp<96 THEN xp=xp1:yp=yp1
1520 PLOT xp1,yp1,5:PLOT xp,yp,
1
1530 xp1=xp:yp1=yp
1540 IF in<>" " THEN 1570
1550 tinte=tinte+1:IF tinte=16
THEN tinte=0
1560 PEN tinte:LOCATE 14,2:PRIN
T CHR$(143)
1570 IF LOWER$(in<>" ")="w" THEN 16
00
1580 tinte=TEST(xp,yp+4)
1590 PEN tinte:LOCATE 14,2:PRIN
T CHR$(143)
1600 IF in<>CHR$(224) THEN 1620
1610 LOCATE (xp+4)/32,25-(yp/16)
:PEN tinte:PRINT CHR$(255)
1620 IF INKEY(58)=-1 THEN 1460
1630 PEN 15:LOCATE 5,23:PRINT"C
ALCULATING;"
1640 '-----
1650 RESTORE 1660
1660 DATA 336,224,88,280,336,22
4,344,536,208,96,88,280,208,96,
344,536
1670 FOR null=0 TO 127:POKE &A1
30+null,0:NEXT
1680 FOR lauf=1 TO 4
1690 READ yt,yt1,xt,xt1
1700 FOR y=yt TO yt1 STEP-16
1710 FOR x=xt TO xt1 STEP 64
1720 lb=TEST(x,y):rb=TEST(x+32,
y)
1730 wt=16*lb+rb:POKE &A200+zae
,wt
1740 zae=zae+1:NEXT x,y,lauf
1750 CALL &A060

```

```

(BF)
(KN) 1770 MODE 0:PAPER 5:PEN 15:CLS
(CP) 1780 LOCATE 8,10:PRINT"SPRITE"
(CN) 1790 MOVE 268,216:DRAW 100,0,1
5:DRAW 0,-50,15:DRAW-100,0,15
:DRAW 0,50,15
(FJ) 1800 POKE &A0C0,13:POKE &A0C1,1
0
(PL) 1810 CALL &A000
(JL) 1820 LOCATE 1,17:PRINT"Run [R]
(GP) --- Ende [E]"
(PE) 1830 IF INKEY(58)>-1 THEN:BASIC
(JD) 1840 IF INKEY(50)>-1 THEN 1860
(AD) 1850 GOTO 1830
(JH) 1860 '-----
(GM) 1870 CLS:WHILE INKEY<>"":WEND
1880 INPUT"Filename des Sprites
";nn$;nn$nn$+"$.spr"
1890 bas=41232:OPENOUT nn$
(PD) 1900 FOR l=1 TO 4:d$(l)=STR$(1)
(OE) +" DATA "
1910 FOR m=0 TO 31:dd$=STR$(PEE
K(bas+1*32+m))
1920 d$(l)=d$(l)+RIGHT$(dd$,LEN
(dd$)-1)
(PA) 1930 IF m<>31 THEN d$(l)=d$(l)+
", "
(EK) 1940 NEXT m:PRINT#9,d$(l):PRINT
(ED) "Saving LINE";l:NEXT l:CLOSEOUT
(IA) 1950 LOCATE 1,17:PRINT"Run [R]
--- Ende [E]"
(OJ) 1960 IF INKEY(58)>-1 THEN:BASIC
(KH) 1970 IF INKEY(50)>-1 THEN RUN
(BH) 1980 GOTO 1960
(CK) ENDE DES LISTINGS

```

```

R$(164)+" by" (PL)
1270 PEN 3:LOCATE 16,6:PRINT"SU
PRA-SOFT" (GP)
(ME) 1280 PEN 2:PRINT:PRINT (CC)
(AJ) 1290 PRINT" After these instruc
tions there will be (NA)
1300 PRINT" action very soon. T
he screen shows you (MA)
1310 PRINT" the road and the ri
ver,crossing them (HE)
(CAL) 1320 PRINT" both will be MADGOO
N'S fate. (AF)
(MB) 1330 PEN 3:PRINT (HJ)
(EA) 1340 PRINT" Four men you must p
lace in the four (BM)
(ONK) 1350 PRINT" holes of the fence.
On your way you'll (ID)
1360 PRINT" see crashing cars,m
ighty monsters and (AH)
(OBJ) 1370 PRINT" scratching snakes:
avoid any touch! (PC)
(AA) 1380 PEN 2:PRINT (FB)
(DF) 1390 PRINT" JOYSTICK/CURSORS ar
e the control-keys. (PH)
(CM) 1400 PRINT" SPACE tunes music o
n/off. Good luck!!! (LK)
(IN) 1410 PEN 1:LOCATE 14,23:PRINT"P
ress any key! (FN)
(MB) 1420 INK 1,2:INK 2,18:INK 3,6:W
HILE INKEY<>"":WEND (BF)
(CO) 1430 IPEY:GOSUB 3040:MODE 1:GOS
UB 2780:MODE 0:PAPER 5:CLS (MP)
1440 ' (AC)
1450 ' (KC)
1460 ' B I L D A U F B A U (IN)
1470 ' (HM)
1480 I BANK,1:MODE 0:PAPER 5:CLS
IQUAD,1,20,3,3,48:IQUAD,1,20,1
2,13,195 (MA)
1490 IQUAD,1,20,21,22,195:IQUAD
,1,20,14,20,243:IQUAD,1,20,4,11
,60 (JN)
1500 PAPER 4:PEN 8:LOCATE 1,3:P
RINT STRING$(20,253):GOSUB 3160
:LOCATE 1,23 (NE)
1510 PRINT STRING$(20,253) (BM)
1520 MOVE 0,32:DRAW 640,0,5:MD
VER 0,2:DRAW-640,0 (JB)
1530 PAPER 5:PEN 13:LOCATE 4,1:
PRINT"MADGOON'S RACE":I STRETCH,
4,1,14 (AJ)
1540 LOCATE 1,24:PRINT"SC: 0
HI: 0 LI:5":I STRETCH,1,24,20 (HI)
1550 PAPER 13:PEN 4:LOCATE 1,17
:PRINT STRING$(20,154) (NJ)
1560 FOR l=15 TO 19 STEP 4:FOR
m=2 TO 20 STEP 2:LOCATE m,1:PRI
NT"--":NEXT m,1 (IP)
1570 RESTORE 1590 (MF)
1580 FOR l=1 TO 8:READ m,n:I PIC
0,m,n,RND*8,1 MOD 2:NEXT (KO)
1590 DATA 3,14,8,14,12,14,17,14
,5,18,10,18,13,18,18,18 (KA)
1600 FOR l=1 TO 8:READ m,n:I PIC
0,m,n,RND*8,2+1 MOD 2:NEXT (KJ)
1610 DATA 2,16,8,16,12,16,15,16
,4,20,9,20,13,20,18,20 (LE)
1620 PAPER 9:I PIC0,5,12,0,6:I PI
C0,14,12,0,6:I STRETCH,5,12,12 (CN)
1630 PAPER 6:PEN 4 (MP)
1640 FOR l=1 TO 8:READ m,n,o:LD
DATE m,n:PRINT STRING$(o,254)::
NEXT (EB)
1650 DATA 10,5,3,18,5,2, 3,7,4,
14,7,3, 6,9,2,11,9,4, 2,11,4,15
,11,3 (DL)
1660 FOR l=1 TO 7:READ m,n,o:I P
IC0,m,n,o,4:NEXT (DI)
1670 DATA 3,4,1,8,4,5,17,4,7,1,
8,5,7,8,8,13,8,1,18,8,2 (NB)
1680 FOR l=1 TO 8:READ m,n:I PIC
0,m,n,15,5:NEXT (LE)
1690 DATA 2,6,8,6,13,6,18,6,4,1
0,7,10,15,10,19,10 (IN)
1700 PEN 0:FOR l=1 TO 5:READ m,
n:LOCATE m,n:PRINT CHR$(255):NE

```

Supra-Basic

Das Spiel: MADGOON'S RACE

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```

1000 '-----
1010 '---- (c) August/Septembe
r 1986 by Alexander Stroh ----
1020 '-----
1030 '
1040 '-----
1050 ' T I T E L B I L D
1060 '-----
1070 MEMORY &3FFF:CALL &BC02:PO
KE &BDEE,&C9
1080 LOAD"mb.bin",&A000:CALL &A
000:GOSUB 3210:OPENIN"topten.fi
l"
1090 FOR l=1 TO 10:INPUT#9,nam$(
l):INPUT#9,pu(l):INPUT#9,day$(
l):NEXT l:CLOSEIN
1100 BORDER 0:GOSUB 3040:MODE 0
:PAPER 5:PEN 14:CLS:GOSUB 3060
1110 PEN 12:LOCATE 1,8:PRINT"(c
) SUPRA-SOFT "B6":PEN 14:IBIG
ON
1120 FOR l=1 TO 10:ISPRITE,4,1,
0:SOUND 1,90+1*5,3:ISPRITE,4,1,
2:NEXT
1130 ISPRITE,4,10,1:INI,1:FOR l
=1 TO 6:SOUND 1,150,3

```

```

1140 LOCATE 2+2*1,12:PRINT MID$(
title$,2*1-1,2):I STRETCH,2+2*1
,12,2
1150 ISPRITE,2+2*1,10,2:ISPRITE
,3+2*1,9,0:FOR m=1 TO 100:NEXT:
ISPRITE,3+2*1,9,2
1160 ISPRITE,4+2*1,10,1:FOR m=1
TO 100:NEXT m,1:LOCATE 16,12:P
RINT"CE"
1170 I STRETCH,16,12,2:INI,1:FOR
l=10 TO 1 STEP-1
1180 ISPRITE,16,1,0:SOUND 1,90+
1*5,3:ISPRITE,16,1,2:NEXT:INK 6
,0:CALL &BD19
1190 PLOT 0,270,6:DRAW 0,146:DR
AW 636,146:DRAW 636,270:DRAW 0,
270
1200 MDVER 0,20:DRAW 636,290:DR
AW 320,398:DRAW 0,290:MOVE 0,12
6:DRAW 320,18
1210 DRAW 636,126:DRAW 0,126
1220 PEN 12:LOCATE 1,17:PRINT"P
ress SPACE to run;":CALL &BD19
1230 INK 6,2:WHILE INKEY(47)<0:
ILTURN,17:FOR l=1 TO 150:NEXT:W
END
1240 GOSUB 3040:MODE 1:PAPER 0:
CLS
1250 PEN 1:LOCATE 14,2:PRINT"MA
DGOON'S RACE"
1260 PEN 2:LOCATE 19,4:PRINT CH

```



```

XT
1710 DATA 1,4,12,4,16,6,10,8,3,
10,10,10
1720 PAPER 9: PEN 12: LOCATE 10,2
2: PRINT CHR$(248)
1730 !SPRITE,1,1,0: !SPRITE,19,1
,0
1740 !BANK,3: GOSUB 3060
1750 FOR a=&4000 TO &4780 STEP
&50: FOR b=a TO a+&3800 STEP &80
0
1760 FOR c=b TO b+&4F: POKE c+&B
000,PEEK(c): NEXT c,b,a
1770 '
1780 '-----
1790 ' V O R S P A N N
1800 '
1810 liv=5: sc=0: mm=0: f=0: mu=1: R
ESTORE 2160: !DISPLAY: CALL &BD19
1820 x=10: y=22: hi=pu(1): GOSUB 3
120: PAPER 9
1830 '
1840 '-----
-
1850 ' H A U P T P R O G R A M
1860 '-----
-
1870 x1=x: y1=y
1880 IF crash=1 THEN 2210
1890 IF INKEY(72) XOR INKEY(0) TH
EN y=y-2 ELSE IF INKEY(73) XOR I
NKEY(2) THEN y=y+2
1900 IF INKEY(74) XOR INKEY(8) TH
EN x=x-1 ELSE IF INKEY(75) XOR I
NKEY(1) THEN x=x+1
1910 IF INKEY(47) >-1 THEN mu=1-
mu: !CALM: RESTORE 2160
1920 IF x<1 THEN x=1 ELSE IF x>
20 THEN x=20
1930 IF y=2 THEN y=3 ELSE IF y>
22 THEN y=22
1940 LOCATE x1,y1: PRINT " "
1950 IF RND>0.5 THEN !TURN,12,1
3 ELSE !TURN,12,13
1960 j=j+1: IF j MOD 2=1 THEN 20
00
1970 !TURN,4,8,14,18: !TURN,6,
10,16,20
1980 IF mm>3 THEN !TURN,14,18: !
TURN,10: IF mm>7 THEN !TURN,6,2
0: !TURN,4,8
1990 IF mm>11 THEN !TURN,4,14: !
TURN,10,16
2000 GOSUB 2240: GOSUB 2260: IF c
rash=1 THEN 1880
2010 LOCATE x,y: PRINT CHR$(248)
: CALL &BD19
2020 IF y<y1 THEN SOUND 2,30,20
,15,2: sc=sc+1: GOSUB 3100: GOSUB
2240
2030 IF y=3 THEN 2280
2040 IF mu=1 THEN ON SQ(1) GOSUB
2120
2050 GOTO 1870
2060 '
2070 '-----
-
2080 ' S P I E L R O U T I N E
N
2090 '-----
-
2100 ' SOUND
2110 f=5-f: RESTORE 2160
2120 READ cc,dd: IF dd=-1 THEN G
OTO 2110
2130 SOUND 1,no(cc+f),dd,13,1: j
j MOD 2
2140 !SPRITE,1,1,jj: !SPRITE,19,
1,jj
2150 RETURN
2160 DATA 1,25,1,12,2,12,1,25,1
,12,2,12,1,25,1,12,2,12,1,25,3,
25,2,25
2170 DATA 2,12,3,12,2,25,2,12,3
,12,2,25,2,12,3,12,5,12,4,12,3,
12,2,12
(AF) 2180 DATA 1,25,1,12,2,12,1,25,1
,12,2,12,1,25,1,12,2,12,1,25,3,
25,2,25
(AH)
(JI) 2190 DATA 2,12,3,12,5,12,4,12,3
,12,2,12,1,25,3,25,1,75,-1,-1
2200 ' CRASH
(MD) 2210 SOUND 4,0,120,15,3,0,1: liv
(LD) =liv-1: crash=0: GOSUB 3140
2220 WHILE INKEY<>: WEND: IF 1
iv=0 THEN 2400 ELSE GOTO 1820
(NG) 2230 ' GET PAPER
(GJ) 2240 PAPER TEST((x-1)*32,(25-y)
(PM) *16): PEN 12: RETURN
(EE) 2250 ' GET ASCII-CODE
(GL) 2260 fe=0: !FIELD,x,y,@fe: IF fe<
(GP) >32 THEN crash=1
2270 RETURN
2280 ' PASSED
(LL) 2290 PAPER 4: LOCATE x,y: PRINT C
HR$(248): sc=sc+15: GOSUB 3100
(IL) 2300 CALL &BCA7: RESTORE 2160
(AL) 2310 mm=mm+1: IF mm MOD 4<>0 THE
N p=1 ELSE p=3: sc=sc+50: GOSUB 3
100
(PF) 2320 FOR k=1 TO p: SOUND 7,106,1
5,15,2: SOUND 7,89,15,15,2
(GJ) 2330 SOUND 7,95,15,15,2: SOUND 7
,119,60,15,2: NEXT
(AC) 2340 FOR k=1 TO p: !250: NEXT: IF
(CM) p=3 THEN GOSUB 3160
(CB) 2350 IF mm<>12 THEN IF mm<16 TH
EN 1820 ELSE 2400
(OM) 2360 FOR l=192 TO 224 STEP 2: MD
VE 0,1: DRAW 640,0,9: NEXT: PAPER
9
(OB) 2370 FOR l=2 TO 17 STEP 5: !PICD
,1,12,0,6: NEXT
(DP) 2380 !STRETCH,1,12,20: GOTO 1820
(LF) 2390 ' GAME OVER
(FI) 2400 !INI,3: !CALM
(MM) 2410 FOR l=0 TO 400 STEP 2: MOVE
0,1: DRAW 640,0,5: NEXT: GOSUB 3
040
(LC) 2420 IF sc>pu(10) THEN GOSUB 253
0
(LD) 2430 GOSUB 3040: MODE 0: PAPER 5:
PEN 4: CLS: CALL &BD19
(HN) 2440 IF mm=16 THEN com$="** VI
CTOR! **": ELSE IF sc>pu(10) THE
N com$="ONE OF THE BEST!": ELSE c
om$="FORGET IT! OKAY?"
(GH) 2450 LOCATE 3,9: PRINT "GAME IS O
VER NOW": PEN 12: LOCATE 3,12: PRI
NT com$
(FK) 2460 PEN 6: LOCATE 5,15: PRINT "AG
AIN? (Y/N)": CALL &BD19
(EF) 2470 !STRETCH,3,9,16: !STRETCH,3
,12,16: !STRETCH,5,15,12: GOSUB 3
060
(NC) 2480 IF INKEY(43) >-1 THEN 2490
(CL) ELSE IF INKEY(46) >-1 THEN 2500
ELSE 2480
(CP) 2490 !NI,1: CLS: GOTO 1810
(PI) 2500 OPENOUT "topen.fil": FOR l=
1 TO 10: PRINT#9,nam$(l)
(PJ) 2510 PRINT#9,pu(1): PRINT#9,day$(
1): NEXT: CLOSEOUT
(BB) 2520 r$="topen.bak": !ERA,@r$: F
OR l=1 TO 5000: NEXT: !HELP: NEW
(HH) 2530 ' HALL OF FAME (eingeben)
(DE) 2540 MODE 1: PAPER 0: PEN 3: CLS: p
(DC) u(10)=sc: nam$(10)="" : day$(10)=""
(IA)
(DD) 2550 LOCATE 16,6: PRINT "EXCELLEN
T!"
(JM) 2560 LOCATE 8,8: PRINT "You've re
ached the TOP TEN": LOCATE 8,9
2570 PRINT "-----"
(AD) " : WHILE INKEY<>: WEND: P
(CB) EN 1: LOCATE 16,12
(ND) 2580 PRINT "ENTER NAME": PEN 2: LO
CATE 14,15: PRINT STRING$(14,"-")
(JC) 2590 INK 2,18: INK 1,26: INK 3,6:
v=14: w=14: l=13: k=0: GOSUB 2650
2600 FOR l=0 TO l+nam$(10)=nam
$(10)+CHR$(c(1)): c(1)=32: NEXT (PM)
2610 PEN 1: LOCATE 11,17: PRINT "E
NTER DAY OF SCORING": PEN 2: LOCA
TE 16,20 (CE)
(OD) 2620 PRINT "-----": v=16: w=1
(BF) 9: l=9: k=0: GOSUB 2650 (FA)
2630 FOR l=0 TO 10: day$(l)=day
$(10)+CHR$(c(1)): c(1)=32: NEXT (DN)
2640 !NI,1: GOTO 2720 (GD)
(CH) 2650 WHILE in$="" : in$=INKEY$: WE
(ID) ND: in=ASC(in$) (GO)
2660 IF in$=CHR$(127) THEN k=k-(
1 AND k>0): LOCATE v+k,w: PRINT "
": c(k)=32 (BJ)
(MJ) 2670 IF in$=CHR$(13) THEN in$=""
: RETURN (NH)
(CB) 2680 IF in<32 OR in>122 THEN in
$="" : GOTO 2650 (IM)
(NN) 2690 LOCATE v+k,w: IF k<-1e THEN
PRINT CHR$(in): ELSE PRINT " " (HK)
(MK) 2700 c(k)=in: k=k+(1 AND k<1e+1)
: in$="" : GOTO 2650 (FC)
(JK) 2710 ' HALL OF FAME (sortieren)
(AM) 2720 GOSUB 3040: CLS (ND)
(AB) 2730 FOR h=10 TO 2 STEP -1: IF pu
(h-1)<pu(h) THEN GOSUB 2750 (FJ)
2740 NEXT h: GOTO 2780 (IB)
(LN) 2750 !EX,2,@pu(h),@pu(h-1): !EX,
3,@nam$(h),@nam$(h-1) (EG)
(KP) 2760 !EX,3,@day$(h),@day$(h-1):
RETURN (KD)
(HF) 2770 ' HALL OF FAME (ausdrucken
) (LF)
2780 PEN 3: LOCATE 13,1: PRINT "TH
E HALL OF FAME": FOR l=1 TO 10 (FB)
2790 PEN 2: LOCATE 2,2+1*2: PRINT
nam$(l): LOCATE 29,2+1*2: PRINT
day$(l) (OL)
(OM) 2800 PEN 1: LOCATE 19,2+1*2: PRIN
T pu(1): LOCATE 18,2+2*1: PRINT "*"
: NEXT 1 (EA)
(OB) 2810 PEN 3: LOCATE 14,25: PRINT "P
ress any key!": !INK 1,26: !INK 2,1
8: !INK 3,6 (CI)
(IG) 2820 WHILE INKEY<>: WEND: !PEY
: CLS: PEN 1: LOCATE 13,13 (BB)
(LF) 2830 PRINT "STATISTIC? (Y/N)" (HL)
2840 IF INKEY(46) >-1 THEN GOSUB
3040: RETURN ELSE IF INKEY(43) =
-1 THEN 2840 (CC)
(PJ) 2850 ' MADGOOD STATISTIC (GB)
2860 CLS: GOSUB 3040: PEN 3: LOCAT
E 12,1: PRINT "MADGOOD STATISTIC":
(EJ)
(DD) 2870 FOR l=12 TO 28 STEP 2: !STR
ETCH,1,1,1: NEXT (CF)
(FI) 2880 PEN 2: LOCATE 2,4: PRINT STR
ING$(38,"=") (KI)
2890 PRINT: PRINT "MADGOOD'S RAC
E was programmed with (HP)
(AE) 2900 PEN 1: PRINT " *SUPRA-BASIC*
": PEN 2: PRINT " (RSX-start at &
A000) (AE)
(GN) 2910 PRINT " in August/September
(GJ) 1986 by ": PEN 1: PRINT "A. STROH
(CL)
2920 PEN 2: PRINT: PRINT " Game-re
cords" (HP)
(FA) 2930 PRINT " The highest score w
as": PEN 1: PRINT pu(1): PEN 2 (MF)
(CM) 2940 PRINT "by": PEN 1: PRINT " ": n
am$(1): PEN 2: PRINT " at the " (BK)
(AL) 2950 PEN 1: PRINT day$(1): PEN 2
: PRINT "!": PRINT (GD)
(BH) 2960 PRINT " Bytes used at the m
oment": PEN 1: PRINT HIMEM-370-
FRE(0) (ML)
(OM) 2970 PEN 2: PRINT: PRINT " Minutes
played": PEN 1: PRINT INT((TIM
E-as)/18000) (GM)
(KE) 2980 PEN 2: PRINT: PRINT " Points
reached before": PEN 1: PRINT s
c: PEN 2 (CK)
(CK) 2990 PRINT " Men in the fencehol
es": PEN 1: PRINT mm: PEN 2: PRI
NT " Lives": (DC)
(IG) 3000 PEN 1: PRINT liv: PEN 2: LOCA

```


programme

```
TE 2,21:PRINT STRING$(38,"="):P
EN 3:LOCATE 8,23
3010 PRINT"Press any key to con
tinue;":FOR 1=8 TO 32 STEP 2:IB
TRETCH,1,23,1
3020 NEXT:INK 1,26:INK 2,18:INK
3,6:CALL &BBO3:IFEY:GOSUB 3040
:RETURN
3030 ' INKS AUS
3040 FOR 1=0 TO 15:INK 1,0:NEXT
:RETURN
3050 ' INKS EIN
3060 INK 0,1:INK 1,24:INK 2,15:
INK 3,4:INK 4,26:INK 6,2:INK 7,
8:INK 8,3
3070 INK 9,18:INK 10,14:INK 11,
16:INK 12,6:INK 13,10:INK 14,26
,18:INK 15,12
3080 RETURN
3090 ' SCORE
3100 PAPER 5:PEN 13:LOCATE 4,24
:PRINT USING"###";sc:ISTRETCH,4
,24,3:RETURN
3110 ' HIGHSCORE
3120 PAPER 5:PEN 13:LOCATE 12,2
4:PRINT USING"###";hl:ISTRETCH,
12,24,3:RETURN
3130 ' LIVES
3140 PAPER 5:PEN 13:LOCATE 20,2
4:PRINT USING"#";lv:ISTRETCH,2
0,24,1:RETURN
3150 ' LOECHER IM ZAUN
3160 PAPER 4:FOR 1=3 TO 18 STEP
5:LOCATE 1,3:PRINT" ":NEXT:RET
URN
3170 '
3180 '
-----
3190 ' SYMBOLE UND
VARIABLEN
3200 '
-----
3210 DEFINT c-z:DIM no(10),c(16
),pu(10),ar$(2),al$(2),day$(10)
,nam$(10)
3220 RESTORE 3240:as=TIME:RANDO
MIZE TIME:SPEED INK 5,5
```

```
(DB) 3230 FOR 1=1 TO 10:READ no(1):N
EXT:FOR 1=0 TO 16:c(1)=32:NEXT
3240 DATA 478,426,379,358,319,2
39,213,190,179,159
(ND) 3250 ENV 1,1,15,1,1,0,1,1,0,1,1
2,-1,8,2,-1,20:ENV 2,10,-1,2:EN
V 3,15,-1,10
(LK) 3260 KEY 140,"Mode 2:paper 0:pe
n 1:list"+CHR$(13):title$="MADG
DON'S RA"
(GB) 3270 SYMBOL 240,1,2,4,127,127,2
55,20,8:SYMBOL 241,240,72,68,25
4,255,255,20,8
(CD) 3280 SYMBOL 242,15,18,34,127,25
5,255,40,16
(GG) 3290 SYMBOL 243,128,64,32,254,2
54,255,40,16
(LB) 3300 SYMBOL 244,0,6,12,24,255,1
27,63,31:SYMBOL 245,0,0,0,0,255
,255,254,254
(CB) 3310 SYMBOL 246,0,0,3,198,108:8
YMBOL 247,0,3,133,198,108
(BB) 3320 SYMBOL 248,60,61,25,255,18
8,188,36,102
(CK) 3330 SYMBOL 249,255,255,255,255
,255,255,40,16
(BI) 3340 SYMBOL 250,112,72,72,254,2
55,255,40,16
(OB) 3350 SYMBOL 251,0,0,1,127,255,1
27,20,8:SYMBOL 252,96,144,8,255
,255,254,40,16
(CF) 3360 SYMBOL 253,34,34,255,34,34
,255,34,34
(GF) 3370 SYMBOL 254,0,0,16,41,198,0
,38,80:SYMBOL 255,0,0,129,90,60
,126,126
(FM) 3380 ISYM,240,241,0:ISYM,251,25
2,1
(CA) 3390 ISYM,242,243,2:ISYM,249,25
0,3
(CBA) 3400 ISYM,244,245,4:ISYM,246,24
7,5:ISYM,150,225,156,6
(CB) 3410 RESTORE 3450:FOR 1=0 TO 25
5
(CB) 3420 READ m:POKE &9800+1,m:NEXT
:FOR 1=0 TO 127
(CB) 3430 POKE &9900+1,240:NEXT:RETU
```

```
RN (AC)
(KE) 3440 ' SPRITE 1 (BM)
3450 DATA 240,240,240,0,240,240
,160,0,240,240,0,0,240,240,16,3
2,240,240,16,34 (CM)
3460 DATA 240,240,0,0,240,240,6
4,0,240,240,0,192 (TE)
3470 DATA 0,240,240,240,0,80,24
0,240,0,0,240,240,16,32,240,240
,16,34,240,240,0 (EE)
3480 DATA 0,240,240,0,128,240,2
40,192,0,240,240 (BK)
3490 DATA 240,240,160,0,240,240
,240,0,240,240,177,114,240,240,
51,240,240,177 (LC)
3500 DATA 114,240,240,240,240,2
40,240,240,240,240,240,240,240,
240 (HN)
3510 DATA 0,80,240,240,0,240,24
0,240,177,114,240,240,240,51,24
0,240,240,177,114 (KN)
3520 DATA 240,240,240,240,240,2
40,240,240,240,240,240,240,240
3530 ' SPRITE 2 (AD)
3540 DATA 240,240,240,240,240,2
40,240,240,240,240,240,240,240,
240,240,0,240,240 (BP)
3550 DATA 160,0,240,240,0,0,240
,240,17,32,240,240,16,32 (BI)
3560 DATA 240,240,240,240,240,2
40,240,240,240,240,240,240,0,24
0,240,240,0,80 (AN)
3570 DATA 240,240,0,0,240,240,1
7,32,240,240,16,32,240,240 (HN)
3580 DATA 240,240,0,0,240,240,6
4,0,240,240,0,192,240,240,160,0
,240,240,240,0 (PB)
3590 DATA 240,240,177,114,240,2
40,51,240,240,240,177,114 (NM)
3600 DATA 0,0,240,240,0,128,240
,240,192,0,240,240,0,80,240,240
,0,240,240,240 (BO)
3610 DATA 177,114,240,240,240,5
1,240,240,177,114,240,240 (DD)
(PB) 1,240,240,177,114,240,240 (NP)
(OA) ENDE DES LISTINGS
```



STAR-TREK

Das Spiel gliedert sich in 5 Levels. Der Schwierigkeitsgrad steigt dabei von Level zu Level durch Steigerung der Bewegungsgeschwindigkeit und der Anzahl gegnerischer Schüsse. Aufgabe des Spielers ist es nun, sein Raumschiff ENTERPRICE, rechts im Spielfeld, möglichst durch alle 5 Levels zu bringen. Näheres steht in der Spielanleitung im Programm. Sollte die Enterprice nach einem Leveldurchlauf Treffer eingesteckt haben, so wird sie

von einem Energietankraumer mit max. 4/10 Energie aufge-tankt. Sollten die Klingonen einmal Stärker gewesen sein, so braucht man nicht verzagen. Die Heimatbasis wird einen Bergungs-raumer schicken, der die kampfunfähige Enterprice abholt und zur Reparatur heimholt. Und bevor es dann heißt: Auf ein Neues!, kann man sich noch die Highscore-Tabelle ansehen. Da wird sich dann auch zeigen, wie hoch der Sonderbonus ist.

Das Listing:

SCHNEIDER LISTING MIT CHECKSUMMEN
V 2.0

```
10 ***** (CC)
*****
20 * STAR TREK (NL)
*
30 * ----- (DD)
= *
40 * (C) Guenter Feldmann 06/8
```



```

6 *
50 * Waldemarstr. 2 :Tel.0209/
878761 *
60 * 4650 Gelsenkirchen 1
*
70 * Mit moralischer Unterstue
tzung *
80 * von meinem Bruder Juergen
*
90 *****
*****
100 INK 0,0:INK 1,26:INK 2,15:I
NK 3,21:INK 4,24:INK 5,9:INK 6,
6:INK 7,8:INK 8,16:INK 9,2:INK
10,15,3:INK 11,18:INK 12,11:INK
13,4:INK 14,26,16:INK 15,22:80
RDER 6:PAPER 0:PEN 1
110 RANDOMIZE TIME
120 DIM score(4),name$(4)
130 lev=1:ks=20:sc=0:sh=10
140 MODE 0
150 *****
*****
160 * Schriftzug STAR TREK ze
ichnen *
170 *****
*****
180 FOR x=1 TO 3
190 READ p:IF p THEN A$(x)=A$(x
)+CHR$(p):GOTO 190 ELSE NEXT
200 DATA 150,154,158,158,156,156,15
0,156,146,158,158,156,150,152,1
48,194,0
210 DATA 147,156,149,151,157,15
1,199,32,149,151,199,151,152,15
1,199,0
220 DATA 154,154,154,153,145,14
5,147,153,193,154,153,145,193,1
55,154,153,193,154,154,154,0
230 LOCATE 3,1:PEN 14:PRINT A$(
1)
240 LOCATE 3,2:PRINT A$(2)
250 LOCATE 1,3:PRINT A$(3)
260 LOCATE 1,5:PEN 3:PRINT STRI
NG$(20,233)
270 LOCATE 2,8:PEN 2:PRINT"DIE
ABENTEUER DES":LOCATE 5,10:PRIN
T"RAUMSCHIFFS"
280 LOCATE 2,12:PEN 1:PRINT"U.B
.S. ENTERPRISE"
290 LOCATE 1,13:PEN 3:PRINT STR
ING$(20,140)
300 LOCATE 1,15:PEN 4:PRINT CHR
$(164)+"1986 by G. Feldmann"
310 LOCATE 4,17:PEN 8:PRINT"Wal
demarstr. 2":LOCATE 1,18:PRINT"
4650 Gelsenkirchen 1"
320 LOCATE 3,19:PRINT"Tel. 0209
/878761"
330 LOCATE 1,21:PEN 7:PRINT STR
ING$(20,233)
340 PEN 14:LOCATE 3,23:PRINT"BI
TTE WARTEN ...":PEN 1
350 MEMORY 42000:sprite=42010
360 *****
***
370 * POKE-ROUTINE FUER SPRITE
S *
380 *****
***
390 RESTORE 490:i=0:k=0:READ a
400 WHILE a<256
410 POKE sprite+i,ak=k+ai=i+1
:READ a
420 IF i MOD 8<>0 GOTO 450
430 IF a=k MOD 256 GOTO 440
440 k=0:READ a
450 WEND
460 *****
470 * STOR SPRITE - ROUTINE *
480 *****
490 DATA 253,229,221,229, 42, 4
2,179,237,152
500 DATA 91, 46,179, 25,203,12
4, 32, 27,215
510 DATA 203, 44, 32, 23,203, 2
(PP) 9,235, 33, 34
520 DATA 199, 0,183,237, 82, 2
2, 0, 48, 3
(FB) 530 DATA 15,122,149, 98,106, 8
7, 30,200, 39
(IB) 540 DATA 213, 24, 39,221,225,25
3,225,201,121
(CD) 550 DATA 0,213,125,230, 7,20
3, 39,203,252
(GJ) 560 DATA 39,203, 39, 87, 30,
0,125,230,241
(CK) 570 DATA 248,111,203, 37,203, 2
0, 68, 77,199
580 DATA 203, 37,203, 20,203, 3
7,203, 20,158
(DP) 590 DATA 9, 25,229, 42, 40,17
9,237, 75, 68
(NM) 600 DATA 44,179, 9,203, 44,20
3, 29,203,146
(JH) 610 DATA 44,203, 29, 6, 0,20
3, 44, 40, 57
(AA) 620 DATA 20,203, 29, 36, 32,
8,120,149, 85
(HD) 630 DATA 72, 96,104, 71, 24, 1
5,193,193, 0
(AP) 640 DATA 221,225,253,225,201,20
3, 29, 62,139
(DJ) 650 DATA 80,149, 56,242, 79,20
9, 25,237, 53
(ID) 660 DATA 91,201,177, 25, 17,
0,192, 25,216
(LD) 670 DATA 48, 4, 17, 80,192, 2
5,197,221, 16
(OA) 680 DATA 225,193,237, 91, 48,17
4, 62, 0, 6
(GE) 690 DATA 128, 40, 24, 19, 19, 1
9, 19, 19, 31
(EL) 700 DATA 26,254,140, 32,203, 1
9, 26,214,146
(BB) 710 DATA 32, 56, 6, 19, 26,21
4, 32, 48,177
(JE) 720 DATA 244, 16,232, 65,197,22
9, 19, 19,253
(KD) 730 DATA 19, 19, 19, 26,254,14
0, 32,174,171
(PN) 740 DATA 19, 26,214, 32, 56,11
7,221,229,146
(FL) 750 DATA 193,120,254, 0, 40,
8, 19, 26,148
(JH) 760 DATA 214, 32, 56,103, 16,24
8, 65, 24,246
(BP) 770 DATA 2, 24,217, 19, 26,21
4, 32, 56, 78
(PL) 780 DATA 90,230, 15,203, 71, 4
0, 2,203, 86
(JF) 790 DATA 247,203, 87, 40, 2,20
3,231,230,219
(CA) 800 DATA 90,203, 79, 40, 2,20
3,215,203, 11
(IH) 810 DATA 95, 40, 2,203,199,23
0, 85,203, 33
(EN) 820 DATA 39,174,119, 19, 26,21
4, 32, 56,167
(MD) 830 DATA 50,230, 15,203, 71, 4
0, 2,203, 46
(GI) 840 DATA 247,203, 87, 40, 2,20
3,231,230,219
(MK) 850 DATA 90,203, 79, 40, 2,20
3,215,203, 11
(GE) 860 DATA 95, 40, 2,203,199,23
0, 85,174, 4
(NL) 870 DATA 119, 62,248,164, 79, 3
5, 62, 7, 8
(LB) 880 DATA 164,177,103, 16,166, 1
9, 26,214,117
(KD) 890 DATA 32, 48,250,225, 1,
0, 8, 9, 61
(NK) 900 DATA 48, 8, 1, 80,192,
9, 62,199, 87
(FK) 910 DATA 164,103,193, 16,140,22
1,225,253, 35
(CD) 920 DATA 225,201,256,170
(CF) 930 *****
(IH) *****
(CB) 940 ** H A U P T P R O G R A M
M **
(INF) 950 *****
#### (ED)
960 LOCATE 2,23:PEN 14:PRINT">>
TASTE DRUECKEN<<":PEN 1 (HB)
970 CALL &BB18 (AD)
980 MODE 1:GOSUB 3220 (NP)
990 ***** (AB)
*****
1000 * Schriftzug STAR TREK ze
ichnen * (ON)
1010 ***** (CD)
*****
1020 GOSUB 2010:MODE 0:BORDER 1
5 (KC)
1030 LOCATE 3,1:PRINT A$(1) (PG)
1040 LOCATE 3,2:PRINT A$(2) (FH)
1050 LOCATE 1,3:PRINT A$(3) (IO)
1060 ***** (CD)
*
1070 * Level-u. Punktetabelle (IC)
*
1080 ***** (FJ)
*
1090 LOCATE 1,4:PEN 2:PRINT STR
ING$(20,143) (BF)
1100 LOCATE 1,4:PRINT CHR$(24)+
"K9:"+CHR$(24) (MP)
1110 LOCATE 8,4:PRINT CHR$(24)+
"8:"+CHR$(24) (CB)
1120 LOCATE 15,4:PRINT CHR$(24)
+"T:"+CHR$(24) (DC)
1130 LOCATE 1,21:PRINT CHR$(150
)+STRING$(18,154)+CHR$(156) (OD)
1140 LOCATE 1,22:PRINT CHR$(149
)+"Shields:";LOCATE 20,22:PRIN
T CHR$(149) (KB)
1150 LOCATE 1,23:PRINT CHR$(149
);LOCATE 20,23:PRINT CHR$(149) (EB)
1160 LOCATE 1,24:PRINT CHR$(149
)"Scores:";LOCATE 20,24:PRINT CH
R$(149) (PA)
1170 LOCATE 1,25:PRINT CHR$(147
)+STRING$(18,154)+CHR$(153)+CHR
$(30) (MK)
1180 sh=10:GOSUB 2640 (LI)
1190 LOCATE 4,12:PEN 1:PRINT"L
E V E L ":lev=FOR c=1 TO 3000:
NEXT:LOCATE 4,12:PRINT STRING$(
14,32) (AM)
1200 ***** (FL)
1210 * Landschaft * (PI)
1220 ***** (NJ)
1230 RESTORE 4970:MOVE 1,100:CA
LL sprite (KP)
1240 ***** (BM)
*****
1250 * Anfangs-Positionen fest
legen * (GN)
1260 ***** (GB)
*****
1270 d=0:e=0:ko=0:kl=0:ks=20:GO
SUB 2720 (PK)
1280 x=480:y=208:RESTORE 4620:M
OVE x,y:CALL sprite (GN)
1290 kx=-130:ky=INT(RND*205+128
):RESTORE 4800:MOVE kx,ky:CALL
sprite (HF)
1300 fx=x+20:fy=y-16:fkx=kx+104
:fkky=ky-16 (KA)
1310 IF lev=1 THEN b=20:b1=10:b
2=16:b3=30 (EO)
1320 IF lev=2 THEN b=16:b1=10:b
2=16:b3=30 (FH)
1330 IF lev=3 THEN b=12:b1=12:b
2=20:b3=40 (MH)
1340 IF lev=4 THEN b=12:b1=12:b
2=20:b3=50 (NG)
1350 IF lev=5 THEN b=10:b1=14:b
2=24:b3=60 (BJ)
1360 ***** (PD)
1370 * JOYSTICK-STEUERUNG * (DB)
1380 ***** (OE)
1390 RESTORE 4620:MOVE x,y:a=JO
Y(O) (CM)
1400 IF a=1 AND y<328 THEN y=y+
b1:GOTO 1440 (JN)

```


programme

```

1410 IF a=2 AND y>136 THEN y=y-
bl:GOTO 1440
1420 IF a=16 THEN e=e+1:GOSUB 2
720:GOSUB 1730:GOTO 1440
1430 GOTO 1450
1440 CALL sprite:MOVE x,y:CALL
sprite
1450 MOVE kx,ky:GOSUB 1520
1460 IF sh>3 THEN LOCATE 14,24:
PRINT SPC(6);
1470 IF sh<=3 THEN LOCATE 14,24
:PEN 6:PRINT"DANGER":PEN 1
1480 GOTO 1390
1490 *****
*****
1500 * Klingonenraumschiff ste
uern *
1510 *****
*****
1520 RESTORE 4800:MOVE kx,ky:CA
LL sprite
1530 kx=kx+b2:ky=ky+(INT(RND*b3
)-b3/2)
1540 IF ky>320 THEN ky=320
1550 IF ky<132 THEN ky=132
1560 fky=kx+144:fy=ky-14
1570 RESTORE 4800:MOVE kx,ky:CA
LL sprite
1580 *** Test ob Kollision
***
1590 IF kx+170>=x-4 AND kx<=x+1
25 AND ky<=y+32 AND ky>=y-32 TH
EN GOTO 1670
1600 *** Schiessmoeglichkeit
? ***
1610 IF kx>20 AND kx<x-170 THEN
z=INT((RND*b)/2):IF z=3 THEN G
OSUB 1870 ELSE GOTO 1630
1620 *** Ausserhalb des Bilde
s? ***
1630 IF kx>=636 THEN RESTORE 48
00:MOVE kx,ky:CALL sprite:FOR c
=1 TO 200:NEXT c:ks=ks-1:kl=kl+
1:GOSUB 2720:IF ks=0 THEN GOTO
2460 ELSE GOTO 1690
1640 MOVE x,y
1650 RETURN
1660 *** Kollision
***
1670 GOSUB 2290:RESTORE 4800:MO
VE kx,ky:CALL sprite:sh=sh-1:GO
SUB 2640:ko=ko+1:IF sh=0 THEN G
OSUB 2210:GOTO 3580 ELSE GOTO 1
290
1680 * Kl.-Schiff neu position
ieren *
1690 kx=-130:ky=INT(RND*205+128
):MOVE kx,ky:CALL sprite:GOTO 1
640
1700 *****
1710 ## UNTERPROGRAMME ##
1720 *****
1730 *****
1740 * Schuss Enterprise *
1750 *****
1760 fx=x+20:fy=y-16
1770 IF fy<ky+4 AND fy>=ky-32
AND fx-320<=kx+160 THEN 1780 EL
SE 1790
1780 PLOT fx,fy,1:DRAW kx+160,f
y: SOUND 1,142,26,7,,,1:FOR i=1
TO 100:NEXT:PLOT fx,fy,0:DRAW k
x+160,fy:GOTO 1800
1790 PLOT fx,fy,1:DRAW fx-320,f
y: SOUND 1,142,26,7,,,1:FOR i=1
TO 100:NEXT:PLOT fx,fy,0:DRAW f
x-320,fy:GOTO 1810
1800 GOSUB 2320:sc=sc+INT(RND*2
0+40):GOSUB 2670:d=d+1:GOSUB 27
20
1810 MOVE x,y
1820 RETURN
1830 *****
1840 * Schuss Klingonraumer *
1850 *****
1860 fky=kx+136:fy=ky-16
1870 IF fky<=y+4 AND fky>=y-32
AND fky+270>=x-8 THEN 1880 ELSE
1890
1880 PLOT fky,fy,9:DRAW x-6,fk
y: SOUND 1,320,26,7,,,1:FOR i=1
TO 100:NEXT:PLOT fky,fy,0:DRAW
x-6,fy:GOTO 1970
1890 PLOT fky,fy,9:DRAW fky+27
0,fy: SOUND 1,320,26,7,,,1:FOR
i=1 TO 100:NEXT:PLOT fky,fy,0:
DRAW fky+270,fy
1900 *** Shield-Kontrolle
***
1910 IF sh<>0 THEN GOTO 1930
1920 IF sh=0 THEN GOTO 1960
1930 MOVE kx,ky
1940 RETURN
1950 *** explosions-shield Ent
erp. ***
1960 GOSUB 2210:RESTORE 4800:MO
VE kx,ky:CALL sprite:GOTO 3580
1970 sh=sh-1:RESTORE 5350:MOVE
x-20,y+10:CALL sprite
1980 FOR c=1 TO 50: SOUND 2,426+
c,1,7,,,1:NEXT
1990 CALL sprite
2000 MOVE kx,ky:GOSUB 2610:GOTO
1920
2010 *****
2020 * Spielernamen-Eingabe *
2030 *****
2040 MODE 0: PEN 14
2050 LOCATE 3,1:PRINT A$(1)
2060 LOCATE 3,2:PRINT A$(2)
2070 LOCATE 1,3:PRINT A$(3)
2080 LOCATE 1,5: PEN 2:PRINT STR
ING$(20,143);
2090 LOCATE 1,12: PEN 6:PRINT ST
RING$(20,202);
2100 LOCATE 1,13:PRINT CHR$(202
);:LOCATE 20,13:PRINT CHR$(202
);
2110 LOCATE 1,14:PRINT CHR$(202
);:LOCATE 20,14:PRINT CHR$(202
);
2120 LOCATE 1,15:PRINT CHR$(202
);:LOCATE 20,15:PRINT CHR$(202
);
2130 LOCATE 1,16:PRINT STRING$(
20,202);
2140 LOCATE 2,10: PEN 8:PRINT"Na
me? (max.18 Bst.)"
2150 LOCATE 2,14: PEN 1:INPUT",
spieler$
2160 IF LEN(spieler$)>18 THEN L
OCATE 1,14:PRINT CHR$(18):GOTO
2090
2170 RETURN
2180 *****
2190 * Explosion der ENTERPRIS
E *
2200 *****
2210 RESTORE 5350:MOVE x-20,y+1
0:CALL sprite
2220 FOR c=1 TO 100: SOUND 2,426
,1,7,,,1:NEXT
2230 CALL sprite
2240 RESTORE 4620:MOVE x,y:CALL
sprite
2250 FOR n=7 TO 1 STEP -1
2260 SOUND 1,426,40,n,,,1
2270 NEXT
2280 RETURN
2290 *****
2300 * Explosion KLINGON-raume
r *
2310 *****
2320 RESTORE 5100:MOVE kx+16,ky
+10:CALL sprite
2330 FOR c=1 TO 100: SOUND 2,426
,1,7,,,1:NEXT
2340 CALL sprite
2350 RESTORE 4800:MOVE kx,ky:CA
LL sprite
2360 FOR n=7 TO 1 STEP -1
2370 SOUND 1,426,40,n,,,1
2380 NEXT
2390 kx=-130:ky=INT(RND*225+108
):MOVE kx,ky:CALL sprite:ks=ks-
1:GOSUB 2720
2400 *** Test Level-Ende
***
2410 IF ks=0 THEN d=d+1:GOSUB 2
720:GOTO 2460
2420 RETURN
2430 *****
2440 * Naechstes Level *
2450 *****
2460 ks=20:GOSUB 2720:RESTORE 4
800:MOVE kx,ky:CALL sprite:lev=
lev+1:IF lev=6 THEN GOTO 3680 E
LSE bo=sh*100:sc=sc+40+bo:GOSUB
2640:RESTORE 4620:MOVE x,y:CAL
L sprite
2470 IF lev=1 THEN GOTO 2560
2480 LOCATE 3,8: PEN 2:PRINT"F L
O T T E "lev-1
2490 LOCATE 4,10: PEN 3:PRINT"ve
rnichtet !!!"
2500 LOCATE 3,12: PEN 8:PRINT"Bo
nus: ",bo,"Pkt. ":PEN 1
2510 FOR c=1 TO 3000:NEXT
2520 LOCATE 1,8:PRINT STRING$(2
0,32);:PRINT:PRINT STRING$(20,3
2);
2530 LOCATE 1,12:PRINT STRING$(
20,32);
2540 IF sh<10 THEN x=456:MOVE x
,y:CALL sprite:GOSUB 2790
2550 PEN 1
2560 LOCATE 4,12:PRINT"L E V E
L "lev
2570 LOCATE 5,14: PEN 2:PRINT"P
L A Y E R"
2580 PEN 1:LOCATE 11-INT(LEN(sp
ieler$)/2),16:PRINT spieler$
2590 FOR c=1 TO 3000:NEXT:LOCAT
E 1,12:PRINT STRING$(20,32);
2600 LOCATE 1,14:PRINT STRING$(
20,32);:PRINT:PRINT STRING$(20,
32);:GOTO 1280
2610 *****
*****
2620 * Punktetabelle aktualisi
eren *
2630 *****
2640 IF sh>10 THEN sh=10
2650 IF sh<0 THEN sh=0:GOSUB 22
10:GOTO 3580
2660 PEN 1:LOCATE 10,22:PRINT S
TRING$(10,32);:FOR c=1 TO sh:LO
CATE 9+c,22:PRINT CHR$(233);:NE
XT c
2670 LOCATE 8,24: PEN 1:PRINT US
ING"####";sc
2680 RETURN
2690 *****
*****
2700 * aktuel.St.i.Zeile 4 pri
nten *
2710 *****
2720 PEN 2:LOCATE 4,4:PRINT CHR
$(24);ks:CHR$(24)
2730 LOCATE 10,4:PRINT CHR$(24)
;:CHR$(24)
2740 LOCATE 17,4:PRINT CHR$(24)
;d:CHR$(24):PEN 1
2750 RETURN
2760 *****
*****
2770 * Energie-Tankraumer steu
ern *
2780 *****
2790 PEN 1
2800 IF sh<0 THEN sh=0:GOSUB 22

```



```

10:GOTO 3580
2810 etx=-120:ety=y+8
2820 RESTORE 5640
2830 FOR c=-120 TO 264 STEP 8
2840 MOVE etx+c,ety:CALL sprite
2850 SOUND 2,144,5,5,,,1
2860 FOR cc=1 TO 50:NEXT cc
2870 CALL sprite
2880 NEXT c
2890 FOR c=264 TO 464 STEP 4
2900 MOVE etx+c,ety:CALL sprite
2910 SOUND 2,288,5,5,,,1
2920 FOR cc=1 TO 50:NEXT cc
2930 CALL sprite
2940 NEXT c
2950 MOVE etx+c,ety:CALL sprite
2960 FOR c=1 TO 1000:NEXT
2970 sh=sh+4:IF sh>10 THEN sh=1
0
2980 LOCATE 10,22:PRINT STRING$(
(10,32):
2990 ton=239:tonsteigerung=0
3000 FOR c=1 TO sh
3010 LOCATE 9+c,22:PRINT CHR$(2
33):
3020 SOUND 2,ton-tonsteigerung,
25,5
3030 tonsteigerung=tonsteigerung
g+12
3040 FOR cc=1 TO 300:NEXT cc
3050 NEXT c
3060 FOR c=1 TO 2000:NEXT c
3070 CALL sprite
3080 FOR c=344 TO -120 STEP -8
3090 MOVE c,ety:CALL sprite
3100 SOUND 2,144,5,5,,,1
3110 FOR cc=1 TO 50:NEXT cc
3120 CALL sprite
3130 NEXT c
3140 FOR cc=1 TO 2000:NEXT cc:
SOUND 5,119,2100,4,,,1
3150 RESTORE 4620:MOVE x,y:CALL
sprite:RESTORE 6590
3160 FOR c=4 TO 456 STEP 8
3170 MOVE x-c,y+4:CALL sprite
3180 IF c#2>=700 THEN 3190 ELSE
FOR cc=1 TO 700-2*c:NEXT cc
3190 MOVE x-c,y+4:CALL sprite
3200 NEXT c
3210 RETURN
3220 *****
3230 * Spiel-Anleitung *
3240 *****
3250 BORDER 26:PEN 1
3260 LOCATE 12,2:PRINT"STAR
T R E K"
3270 LOCATE 1,3:PEN 11:PRINT ST
RING$(40,207):
3280 LOCATE 2,6:PEN 1:PRINT"Ste
rzeit 26.1253"
3290 LOCATE 2,8:PRINT"Dies sind
die Abenteuer des Raumschiffs
ENTERPRISE, das mit seiner 500
-Mann starken Besatzung unterwegs
ist um neue Welten zu erforschen,
die nie zuvor ein Mensch betreten
hat...."
3300 LOCATE 2,16:PEN 9:PRINT"Sie
chleupfen Sie in die Rolle des
Kommanders James T. Kirk
und starten Sie im Auftrag
der UNITED FEDERATION OF PLANETS
zur Randwelt FOWLER'S PLANET"
3310 LOCATE 12,24:PEN 1:PRINT"M
usic off: >SPACE<"
3320 GOSUB 4470:LOCATE 12,24:PR
INT STRING$(18,32):LOCATE 14,2
4:PEN 1:PRINT">>> SPACE <<<"
3330 CALL &BB18
3340 CLS
3350 LOCATE 12,2:PRINT"STAR
T R E K"
3360 LOCATE 1,3:PEN 11:PRINT ST
RING$(40,207):
3370 LOCATE 1,6:PEN 1:PRINT"FO
WLER'S PLANET ist eine neuentde
ckte Welt weitab der uebliche
n Handelsrouten und eine
fuerr die UFF wertvolle Welt."
3380 LOCATE 1,10:PEN 11:PRINT"
Aber auch die KLINGONEN interes
sieren sich fuer diese Welt !
"
3390 LOCATE 1,12:PEN 1:PRINT"Si
e sollen nun mit der USS ENTER
PRISE den Planeten vor dem Zu
griff der KLIN-GONEN retten."
3400 LOCATE 1,15:PRINT" Ihr Rau
mschiff ist mit einem energie-
reichen Phaser und einem star
ken Schutzschirm ausgest
attet, dem die KLINGONEN n
ur ihre Flottenstaerke en
tgegenzusetzen haben."
3410 LOCATE 1,20:PRINT" 5 KLING
ONEN-Flotten stehen Ihnen gegen
- ueber.Beginnen Sie den Kampf,
aber unterschuetzen Sie d
en Begner nicht !!"
3420 LOCATE 1,23:PEN 11:PRINT"
V i e l G l u e c k ! !"
3430 LOCATE 12,24:PEN 1:PRINT"M
usic off: >SPACE<"
3440 GOSUB 4470:LOCATE 12,24:PR
INT STRING$(18,32):LOCATE 14,2
4:PEN 1:PRINT">>> SPACE <<<"
3450 CALL &BB18
3460 CLS
3470 LOCATE 12,2:PEN 1:PRINT"S
T A R T R E K"
3480 LOCATE 1,3:PEN 11:PRINT ST
RING$(40,207):
3490 LOCATE 1,7:PEN 1:PRINT" Di
e ENTERPRISE wird mit dem Joyst
ick gesteuert."
3500 LOCATE 1,9:PRINT" Fuer jed
en Abschluss erhalten Sie 40 -
60 Punkte."
3510 LOCATE 1,11:PRINT" Wird di
e ENTERPRISE von einem Klingon-
Raumer getroffen,so wird Ihne
n etwas Schutzschild-Energie
abgezogen."
3520 LOCATE 1,15:PEN 11:PRINT"
Achtung !":LOCATE 1,16:PEN 1:PR
INT" Auch bei Kollisionen Energ
ie-Verlust ! Ausserdem erhalte
n Sie bei Kollisionen keine Pu
unkte."
3530 LOCATE 1,20:PRINT" Nach je
dem Level erhalten Sie 100 Pkt.
fuer jedes zehntel Restenergi
e und einen Sonderbonus am
Ende des Spiels."
3540 LOCATE 12,24:PEN 1:PRINT"M
usic off: >SPACE<"
3550 GOSUB 4470:LOCATE 12,24:PR
INT STRING$(18,32):LOCATE 14,2
4:PEN 1:PRINT">>> SPACE <<<"
3560 CALL &BB18
3570 CLS:RETURN
3580 *****
3590 * Mission nicht erfuehlt
*
3600 *****
3610 GOSUB 3840
3620 LOCATE 3,8:PEN 2:PRINT"Ihr
e Mission war"
3630 LOCATE 5,10:PEN 1:PRINT"ER
FOLGLOS !!"
3640 LOCATE 4,13:PEN 6:PRINT"Ta
belle [J/N]:PEN 1
3650 a#=UPPER$(INKEY$)
3660 IF a#="N" THEN MODE 2:END
3670 IF a#="J" THEN GOSUB 4120:
CLS:RUN ELSE 3650
3680 *****
3690 * Mission erfuehlt *
3700 *****
3710 RESTORE 4620:MOVE x,y:CALL
sprite
3720 sc=sc+sh*100:GOSUB 2640
3730 LOCATE 7,7:PEN 1:PRINT"BRA
VO !!"
3740 LOCATE 5,10:PEN 8:PRINT"Si
e haben es"
3750 LOCATE 5,13:PEN 3:PRINT"GE
SCHAFFT !!"
3760 LOCATE 4,15:PEN 1:PRINT"Bo
nus :":sh=100
3770 LOCATE 4,18:PEN 6:PRINT"Ta
belle [J/N]:PEN 1
3780 a#=UPPER$(INKEY$)
3790 IF a#="N" THEN MODE 2:END
3800 IF a#="J" THEN GOSUB 4120:
RUN ELSE 3780
3810 *****
3820 * Berqung der besch. ENTE
RPRISE *
3830 *****
3840 RESTORE 6080:x=480:y=308:b
x=644:by=152
3850 MOVE x,y:CALL sprite
3860 RESTORE 5790
3870 FOR bx=636 TO 460 STEP -4
3880 MOVE bx,by:CALL sprite
3890 SOUND 2,426,20,5,,,1
3900 FOR c=1 TO 50:NEXT c
3910 CALL sprite
3920 NEXT bx
3930 FOR by=152 TO 300 STEP 4
3940 MOVE bx,by:CALL sprite
3950 SOUND 2,213,20,5,,,1
3960 FOR c=1 TO 100:NEXT c
3970 CALL sprite
3980 NEXT by
3990 bx=460:by=300
4000 MOVE bx,by:CALL sprite
4010 FOR c=1 TO 3000:NEXT c
4020 CALL sprite:RESTORE 6080:M
OVE x,y:CALL sprite
4030 by=308:bx=464
4040 RESTORE 6260
4050 FOR bx=464 TO 636 STEP 4
4060 MOVE bx,by:CALL sprite
4070 SOUND 2,851,20,5,,,1
4080 FOR c=1 TO 100:NEXT c
4090 CALL sprite
4100 NEXT bx
4110 RETURN
4120 *****
4130 * High-Score-Tabelle *
4140 *****
4150 MODE 1
4160 LOCATE 6,2:PEN 1:PRINT"Die
":LOCATE 10,2:PEN 2:PRINT"STAR
TREK ":PEN 1:LOCATE 20,2:PRINT
"Highscore-Tabelle"
4170 LOCATE 1,4:PRINT STRING$(4
0,143):
4180 gnd=50:h=(e-d)*20:i=ko*40:
j=k1*150
4190 LOCATE 2,6:PRINT"Anzahl de
r Treffer ":PRINT USING"####
":d:PRINT" Bonus:":PRINT USI
NG"####":g
4200 LOCATE 2,7:PEN 6:PRINT"Anz
ahl Fehlschuesse:":PRINT USING
"####":e-d:PEN 2:PRINT" Abzu
g":PRINT USING"####":h
4210 LOCATE 2,8:PEN 1:PRINT"Anz
ahl Kollisionen ":PRINT USING
"####":ko:PEN 2:PRINT" Abzug
":PRINT USING"####":i
4220 LOCATE 2,9:PEN 6:PRINT"Ent
kommen sind ":PRINT USING
"####":k1:PEN 2:PRINT" Abzug
":PRINT USING"####":j
4230 LOCATE 2,10:PEN 1:PRINT"So
nderbonus gesamt ":PRINT USIN
G"####":g-h-i-j
4240 LOCATE 2,11:PEN 6:PRINT"DE
INE PUNKTZAHL ":PEN 14:PRI

```


programme

```

NT USING"####":sc:PEN 1
4250 sc=sc+g-h-i-j
4260 t=0:IF sc<score(4) THEN 43
10
4270 IF sc<score(3) THEN score(
4)=sc:name$(4)=spieler$:GOTO 43
10
4280 IF sc<score(2) THEN t=1:GOTO
4310
4290 IF sc<score(1) THEN t=2:GOTO
4310
4300 IF sc<score(0) THEN t=3:EL
SE t=4
4310 FOR i=1 TO t
4320 score(5-i)=score(4-i):name
$(5-i)=name$(4-i)
4330 NEXT i
4340 score(4-t)=sc:name$(4-t)=s
pieler$
4350 LOCATE 1,13:PRINT" PLATZ
PUNKTE NAME"
4360 PRINT STRING$(28,"-"):LOC
ATE 1,15:FOR i=0 TO 4
4370 PRINT" ":i+1:LOCATE 16-IN
T:LOG(score(i)+1)/LOG(10),15+i
:PRINT score(i):LOCATE 22,15+i
:PRINT name$(i)
4380 NEXT i
4390 LOCATE 1,19-t:PRINT">":LOC
ATE 1,24
4400 LOCATE 11,24:PEN 1:PRINT" M
usic off: >SPACE<"
4410 GOSUB 4470:a$="":LOCATE 11
,24:PRINT STRING$(18,32):LOCAT
E 4,22:PEN 6:INPUT"Nach ein Spi
el":a$:PEN 1
4420 IF a$="J" OR a$="j" THEN M
ODE 0:BORDER 15:sc=0:sh=10:ks=2
0:lev=1:GOTO 1020
4430 IF a$="N" OR a$="n" THEN M
ODE 2:END ELSE 4410
4440 *****
4450 * Melodie (Eurovisionstit
el) *
4460 *****
4470 tonh=0
4480 RESTORE 4560:tonh=tonh+1:I
F tonh=5 THEN tonh=1
4490 FOR c=1 TO 36:READ ton,ton
1:tondau=200/ton1
4500 a$=INKEY$:IF a$<>" " THEN 4
540
4510 SOUND 5,0,2,0:SOUND 5,ton/
tonh,tondau,4
4520 NEXT c
4530 SOUND 5,0,100,0:GOTO 4480
4540 SOUND 5,0,1,0:FOR c=1 TO 5
00:NEXT c
4550 RETURN
4560 DATA 758,4,568,4,568,8,506
,8,451,4,568,4,379,2,451,4,451,
4,426,4,379,8,426,8
4570 DATA 451,8,426,8,379,4,506
,8,568,8,506,8,451,8,506,4,758,
4,568,4,568,8,506,8
4580 DATA 451,4,568,4,379,2,451
,4,451,4,426,8,379,8,451,8,426,
8,506,3,568,8,568,1,5
4590 *****
4600 * ENTERPRISE - SPRITE *
4610 *****
4620 DATA 3
4630 DATA 2 444444
4444444444
4640 DATA kkk 454555
555544444
4650 DATA kkkkk 22 444444
44444444
4660 DATA 44444444444444 444
4444
4670 DATA 44444444444444 44
4
4680 DATA kkkkk 444 444
4690 DATA 2 444 444

```


5420 DATA jnjjj		5890 DATA 333333333388		4444 BB kknkkk	(DF)
5430 DATA jjjj	jnj	5900 DATA 33333333338888888888	(NG)	6360 DATA j3	881 44 4
5440 DATA jnnj	jnnj	888888888888kkkkkkk1	(GF)	444444 884kknkkk	(LI)
5450 DATA jnj	jnnj	5910 DATA 77777777777777777777	(DE)	6370 DATA 33	881444444
5460 DATA jnj	jnnj	7777777777kkkkkkk1	(EE)	444444 884kkkkkkk1	(PK)
5470 DATA jjj	jnnj	5920 DATA 77777777722222222222	(CG)	6380 DATA 333333	8 1442244
5480 DATA jnj	jnnj	22222222772kkkkkkk1	(MD)	4 4444 884kkkkkkk1	(PD)
5490 DATA jjjj	jnnj	5930 DATA 777777342222222222	(HF)	6390 DATA 39333333	88 444444
5500 DATA jnnj	jnnj	22222222772knknkk	(NC)	444 884kkkkkkk1	(JO)
5510 DATA jjj	jnnj	5940 DATA 77 3344447777774	(AM)	6400 DATA 333333333388	444
5520 DATA jnnj	jnnj	444777777777kkknkk	(BL)	44 884knnnnkk1	(GO)
5530 DATA jnnj	jnnj	5950 DATA 111111 4444 111	(CN)	6410 DATA 33333333333888888888	888888888888kkkkkkk1
5540 DATA jjjj	jnnj	4444 111 kkkk	(IG)	6420 DATA 77777777777777777777	
5550 DATA jnnj	jnnj	5960 DATA 4444	(GB)	777777777777kkkkkkk1	(KO)
5560 DATA jnj j	jnnj	5970 DATA 4444	(HA)	6430 DATA 77777777722222222222	
5570 DATA jnnj j	jnnj	5980 DATA kkkkkkkkkkkkkkkk	(MO)	22222222772kkkkkkk1	(IP)
5580 DATA jnnj j	jnnj	5990 DATA kkkk9999kkk999999	(FC)	6440 DATA 777777342222222222	
5590 DATA jnnj j	jnnj	6000 DATA kkkkk9999kk9911111	(ID)	22222222772knknkk	(LL)
5600 DATA jnnj j	jnnj	999999199kkkkkkk	(FD)	6450 DATA 77 3344447777774	
5610 DATA jnnj j	jnnj	6010 DATA kkkkk99kk99999999	(BA)	444777777777kkknkk	(DI)
5620 DATA jnnj j	jnnj	99999999kk111kkk	(OD)	6460 DATA 111111 4444 111	
5630 DATA jnnj j	jnnj	6020 DATA kkkkkkkkkkkkkkkk	(BA)	4444 111 kkkk	(BA)
5640 DATA jnnj j	jnnj	6030 DATA kkkkk	(HA)	6470 DATA 4444	
5650 DATA jnnj j	jnnj	6040 DATA k	(BO)	6480 DATA 4444	
5660 DATA jnnj j	jnnj	6050 '*****	(CD)	4444	(BF)
5670 DATA jnnj j	jnnj	6060 '* Kampfunaehige ENTERPRISE *	(CD)	6490 DATA kkkkkkkkkkkkkkkk	
5680 DATA jnnj j	jnnj	6070 '*****	(FK)	kkkkkkkkkkkkkkkkkkk	(HF)
5690 DATA jnnj j	jnnj	6080 DATA 2 3	(IE)	6500 DATA kkkk9999kkk999999	
5700 DATA jnnj j	jnnj	6090 DATA 2 4444	(DA)	99999999kk111kkk	(AB)
5710 DATA jnnj j	jnnj	44444444	(EP)	6510 DATA kkkkk9999kk991111	
5720 DATA jnnj j	jnnj	6100 DATA kk 45455	(EP)	999999199kkkkkkk	(EF)
5730 DATA jnnj j	jnnj	55544444	(HA)	6520 DATA kkkkk99kk99999999	
5740 DATA jnnj j	jnnj	6110 DATA kkkk 22 444444	(KF)	99999999kkkkkkk	(NK)
5750 DATA jnnj j	jnnj	44444444	(KF)	6530 DATA kkkkkkkkkkkkkkkk	
5760 DATA jnnj j	jnnj	6120 DATA 4444444444 444 444	(KF)	kkkkkkkkkkkkkkk	(JA)
5770 DATA jnnj j	jnnj	4444	(KF)	6540 DATA kkkkk	
5780 DATA jnnj j	jnnj	6130 DATA 4 4444444444 44	(KF)	6550 DATA k	(JND)
5790 DATA jnnj j	jnnj	4	(KF)	6560 '*****	
5800 DATA jnnj j	jnnj	6140 DATA kkkkk 444 444	(KF)	****	(CG)
5810 DATA jnnj j	jnnj	6150 DATA 2 444 444	(KF)	6570 '* ENTERPRISE m. Schubstra	
5820 DATA jnnj j	jnnj	6160 DATA k444 444	(KF)	h1 *	(FK)
5830 DATA jnnj j	jnnj	6170 DATA 4444444444	(KF)	6580 '*****	
5840 DATA jnnj j	jnnj	6180 DATA 1 44 444444	(KF)	****	(HA)
5850 DATA jnnj j	jnnj	6190 DATA 144444444444	(KF)	6590 DATA nj	(KF)
5860 DATA jnnj j	jnnj	6200 DATA 1442244 4444	(KF)	6600 DATA nnnnj	(AK)
5870 DATA jnnj j	jnnj	6210 DATA 444444444	(KF)	6610 DATA 3	(AP)
5880 DATA jnnj j	jnnj	6220 DATA 44 444	(KF)	nnjjnnj	
5890 DATA jnnj j	jnnj	6230 '*****	(KF)	2 444444	(AF)
5900 DATA jnnj j	jnnj	6240 '* Enterprise u. Bergungschiff *	(KF)	4444444444njnnjjnnnj	
5910 DATA jnnj j	jnnj	6250 '*****	(KF)	6630 DATA kkk 454555	(AF)
5920 DATA jnnj j	jnnj	6260 DATA 3	(KF)	555544444njnnnnjjnnnnnj	(BF)
5930 DATA jnnj j	jnnj	6270 DATA 2 4	(KF)	6640 DATA kkkk 22 444444	
5940 DATA jnnj j	jnnj	444 44444444	(KF)	444444444njnnjjnnnj	(AE)
5950 DATA jnnj j	jnnj	6280 DATA kk 4	(KF)	6650 DATA 444444444444 444	
5960 DATA jnnj j	jnnj	5455 55544444	(KF)	4444 nnnjjnnnj	(NN)
5970 DATA jnnj j	jnnj	6290 DATA kkkk 22 4	(KF)	6660 DATA 444444444444 44	
5980 DATA jnnj j	jnnj	44444444444444	(KF)	4 nnnnj	(BK)
5990 DATA jnnj j	jnnj	6300 DATA 4444444444 444	(KF)	6670 DATA kkkk 444 444	
6000 DATA jnnj j	jnnj	444444488	(KF)	nj	(BP)
6010 DATA jnnj j	jnnj	6310 DATA 4 4444444444	(KF)	2 444 444	(CG)
6020 DATA jnnj j	jnnj	444 888	(KF)	k444 444	(JJ)
6030 DATA jnnj j	jnnj	6320 DATA kkkkk 444	(KF)	4444444444	(BP)
6040 DATA jnnj j	jnnj	444 88	(KF)	144444444444	(HB)
6050 DATA jnnj j	jnnj	6330 DATA 2 444	(KF)	4	(HB)
6060 DATA jnnj j	jnnj	444 88 j	(KF)	6720 DATA 144444444444	
6070 DATA jnnj j	jnnj	6340 DATA k444	(KF)	4	(CC)
6080 DATA jnnj j	jnnj	444 88 kkkk	(KF)	6730 DATA 144224444444	
6090 DATA jnnj j	jnnj	6350 DATA 444444	(KF)	4	(HD)
6100 DATA jnnj j	jnnj		(KF)	6740 DATA 444444444	
6110 DATA jnnj j	jnnj		(KF)	6750 DATA 4444444	(CO)
6120 DATA jnnj j	jnnj		(KF)	6760 '**** W I C H T I G ! !	(BJ)
6130 DATA jnnj j	jnnj		(KF)	! **** Das REM in dieser	
6140 DATA jnnj j	jnnj		(KF)	Zeile darf NICHT weggelassen werden !!!	(BA)
6150 DATA jnnj j	jnnj		(KF)	6770 '*** S A V E ***	(KP)
6160 DATA jnnj j	jnnj		(KF)	10000 SPEED WRITE 1	(HC)
6170 DATA jnnj j	jnnj		(KF)	10010 SAVE"STAR TREK"	(AM)
6180 DATA jnnj j	jnnj		(KF)	ENDE DES LISTINGS	

Endlich ist er da!

Der neue Checksummer für die Schneider-Computer CPC 464/ 664/ 6128

Worauf alle Schneider-User bisher sehnsüchtig gewartet haben, das wird jetzt an dieser Stelle veröffentlicht: Eine erheblich verbesserte Checksummerversion. Nun ist es möglich, schon bei der Eingabe einer Basiczeile festzustellen, ob ein Eingabefehler begangen wurde.

Ab dieser Ausgabe finden Sie hinter jeder Zeile eine zweistellige Buchstabenkombination, die von geschweiften Klammern eingrahmt wird. Anhand dieser Prüfsumme kann die Eingabe jetzt auf Ihre Richtigkeit hin untersucht werden.

Um ein beliebiges TRONIC-

Programm abzutippen, gehen Sie wie folgt vor: Laden Sie zuerst den Checksummer und starten Sie diesen mit "RUN". Auf die Frage nach dem verwendeten Computertyp antworten Sie bitte mit der entsprechenden Computerbezeichnung (464/ 664/ 6128) und geben Sie diese ein. Anschließend wird der Checksummer aktiviert und der BASIC-LOADER automatisch gelöscht. Nun können Sie mit der Eingabe des Programmes beginnen. Ist eine Zeile komplett eingegeben, so wird diese wie üblich mit "ENTER" abgeschlossen. Unterhalb der Zeile erscheint daraufhin die Checksum-

me. Stimmt diese mit der im Heft abgedruckten Summe überein, so wurde diese Zeile fehlerfrei eingegeben.

Stimmen diese beiden Zeilen nicht überein, liegt ein Tippfehler vor, den Sie wie folgt korrigieren müssen. Dabei ist bei der Eingabe des Listings einiges zu beachten: Starten Sie niemals ein Programm, wenn der Checksummer aktiv ist. Speichern Sie das Programm nach der Eingabe sofort ab. Die Checksummen selbst werden nicht eingetippt. Sie können bei der Eingabe sowohl Groß- als auch Kleinbuchstaben verwenden. Bis auf eine Ausnah-

me können alle Editiermöglichkeiten verwendet werden. Ausnahme: Eine Zeile kann nicht durch die Eingabe der Zeilennummer gelöscht werden. Um beispielsweise die Zeile 200 zu löschen geben Sie DELETE 200 ein. Leerzeichen die keine Bedeutung haben werden selbstständig ignoriert und gelöscht. Allerdings besitzen Leerzeichen innerhalb von Anführungsstrichen einen Wert und verändern somit die Checksumme. Für den PRINT-Befehl kann auch das Fragezeichen verwendet werden. Viel Spaß und fehlerfreies Eingeben wünscht OTTFRIED SCHMIDT !

```

1 * *****
2 * SCHNEIDER CPC *
3 * CHECKSUMMER V 2.0 *
4 * (C) 1986 *
5 * OTTFRIED SCHMIDT *
6 * *****
7 :
8 :
1000 DATA 3E,FF,32,00,AC,CD,22,BF,F5,C5
1010 DATA D5,E5,2A,20,BF,CD,61,DD,B7,2B
1020 DATA 54,E5,2A,20,BF,CD,6E,A0,E1,30
1030 DATA 4A,CD,04,EE,CD,A3,E7,CD,63,E1
1040 DATA 2A,20,BF,06,FF,AF,4F,7E,FE,00
1050 DATA 2B,05,B1,AB,23,10,F5,B1,AB,4F
1060 DATA 3E,0D,CD,5A,BB,3E,0A,CD,5A,BB
1070 DATA 3E,7B,CD,5A,BB,79,E6,F0,1F,1F
1080 DATA 1F,1F,C6,41,CD,5A,BB,79,E6,0F
1090 DATA C6,41,CD,5A,BB,E1,D1,C1,F1,3E
1100 DATA 7D,C3,5A,BB,C9,E1,D1,C1,F1,C9
1110 DATA CD,61,DD,B7,37,CB,CD,04,EE,D0
1120 DATA 7E,FE,20,20,01,23,CD,D2,E6,37
1130 DATA 9F,C9,00,00,00,00,00,00,00
1140 MEMDRY &9FFF:FOR i=&A000 TO &A0B5
1150 READ a$:POKE i,VAL("&"+a$):NEXT i
1160 CLS:INPUT"BENUTZEN SIE EINEN CPC 46
4/664 oder 6128";v$
1170 IF v$<>"464" AND v$<>"664" AND v$<>
"6128" THEN PRINT CHR$(7):GOTO 1160
1180 IF v$="464" THEN adr=&BD3A:pufa=&A4
ELSE IF v$="664" THEN adr=&BD5B:pufa=&B
A ELSE adr=&BD5E:pufa=&BA
1190 POKE &BF20,pufa:POKE &BF21,&AC
1200 POKE &BF22,PEEK(adr):POKE &BF23,PEE
K(adr+1):POKE &BF24,PEEK(adr+2)
1210 POKE adr,&C3:POKE adr+1,0:POKE adr+
2,&A0
1220 IF v$="664" THEN GOTO 1250
1230 IF v$="6128" THEN GOTO 1300
1240 GOTO 1340
1250 POKE &A010,&52:POKE &A011,&DE:POKE
&A06F,&52:POKE &A070,&DE
1260 POKE &A020,&D4:POKE &A021,&EE:POKE
&A075,&D4:POKE &A076,&EE
1270 POKE &A07F,&AA:POKE &A080,&E7:POKE
&A023,&96:POKE &A024,&EB
1280 POKE &A026,&59:POKE &A027,&E2
1290 GOTO 1340
1300 POKE &A010,&4D:POKE &A011,&DE:POKE
&A06F,&4D:POKE &A070,&DE
1310 POKE &A020,&CF:POKE &A021,&EE:POKE
&A075,&CF:POKE &A076,&EE
1320 POKE &A07F,&A5:POKE &A080,&E7:POKE
&A023,&64:POKE &A024,&EB
1330 POKE &A026,&54:POKE &A027,&E2
1340 CLS:PRINT"CHECKSUMMER V2.0 IST JETZ
T AKTIV!"
1350 NEW

```


Das nächste Heft erscheint bereits am 12. Januar 1987!

Bis dahin wünschen wir einen guten Rutsch!

* **Kleinanzeigen** * **Kleinanzeigen** * **Kleinanzeigen** *

Biete Software

VC-20 Software zu Kleinstpreisen. Original Steckmodule und Kassetten bis zu 80% reduziert. Liste gegen 80 Pf bei Mini-Soft, Stryckweg 48, 3542 Willingen

Verkaufe 5 C-16/116 Spiele auf Kassette (Sherlocks, Arrow) zusammen nur 10,-DM! Vorkasse!!! Hubert Mühlbacher, Plattenberg 3, 8221 Waging

***** C16/116 u. + 4 GRAPHIC-ADVENTURE *****

Nach den "JÜNGERN..." jetzt das Zweite, "Die Maske" -lassen Sie sich in das Reich der Phantasie entführen. Über 45 Bilder 2x10.Min., Cassette voll bespielt. Cassette 20,-DM (Schein) oder per NN+4,-DM VK. Bestellen bei: GBR U+K, Bickernstr. 161, 4650 Gelsenkirchen. Info gegen 80 Pf.

C-16 SPITZENSOFTWARE

Programmpaket:

1. Pilot (brandneues Adventure)
 2. Blackjack C-16
 3. Flugsimulator
- Cassette kommt sofort für 10,-DM Vorkasse (incl. Porto) an: Ludwig Wiesemüller, Birkenweg 37, 3408 Duderstadt 27

10 Spiele für CPC 464 auf Kassette für 20,-DM (Schein). Bernd Ehring, Eisack-str. 9, 1000 Berlin 62

Achtung C-16 User!

Tausche und verkaufe Anwender- und Spielprogramme. Gr. Auswahl! Ulf Peters, Gablonzer Str. 11, 2351 Trappenkamp

CPC-Schneider**CPC-Schneider

Tausche und verkaufe Software auf Kassette und 3"Diskette! Wer schickt mir seine Liste zum Tauschen? Auf jeden Fall Liste anfordern bei: Andreas Günther, Bürgerm.-Trupp-Str. 25, 6700 Lu.-Oppau, Bitte ausreichend frankierten Rückumschlag beilegen!

***** C 16/116/Plus 4 Zeichen- und *****

*****Grafikprogramme für 16K und*****

*****64K-Version. CAD-Basissystem*****

*****Info gegen Porto bei Dipl.Ing*****

*****M. Rätzel, Ulvenbergstr. 6,*****

*****D-6100 Darmstadt 13*****

C-64*C-64*C-64*C-64*C-64*C-64

Silicon Warrior 5 DM, Space Shuttle 10DM, Richard Petty's Rennzirkus 10 DM, MTSS All Mude America 20 DM u.a. Programme. Alle Progr. auf Disk für Floppy 1541. Info gegen 80 Pf. Rückporto von Stefan Leinauer, Ahornring 2, 8028 Taufkirchen

C-16 !!! Verkaufe TOP-Spiele für C-16,

z.B. Bandits at Zero, Rockman usw. 20% Ermäßigung auf jedes Spiel. Liste bei Christoph Furtkamp, Tel. 02851/2511

C-16/Plus 4! Spiele und Anwenderpro-

gramme nur 1,-DM! Große Liste anfordern gegen Briefmarke oder Rückumschlag bei Jürgen Roumen, Eichenstr. 49, 4054 Nettetal 1

Brauchst Du heute noch 5 Spiele

kauf bei Mini-Soft ganz viele.

Restposten: C16/116 Software

Gammelpack 1 (5 Spiele) 11,11 DM

Gammelpack 2 (10 Spiele) 18,18 DM

Die Gamespacks beinhalten: z.B. Kobra, Tennis, Moon Fighter, Jumpman u.a. Bestell es direkt bei: Mini-Soft, Stryckweg 48, 3542 Willingen

Verkaufe über 20 Originalspiele für

C16, z.B. SOCCER, SKRAMBLE,

ROCKMAN, und so. Spielekassette

von Robtek. Bitte melden bei René

Reinthal, Tel. 02225/2890 ab 14 Uhr

Programme für den C-64 & VC-20 zu

Minipreisen! Liste gibt es gegen 80 Pf.

(Briefmarke) bei SF-Soft, Mühlenweg 7,

3401 Seulingen

Beispiel: Programm berechnet die beste Möglichkeit zum Schummeln bei ARBEITEN: Individuell für jeden Schüler. Nur DM 5,- (bzw. DM 3,- bei VC20). Bitte Computertyp angeben.

C-64/128 Software (Originalprogram-

me) z.B. Bard's Tale 30,- DM

Liste gegen Freiumschlag bei: Rudolf

Lex, Prinzregentenstr. 100,

8000 München 60

COMPUTER SERVICE - DER C-16 SPEZIALIST

Postfach 1304 · 7913 Senden · Telefon (0 73 07) 6230

24 Stunden Bestellservice (Anrufbeantworter)

schnelle Lieferung · sagenhaft günstige Preise!!!

SPIELPROGRAMME		ANWENDERPROGRAMME	
Winter Olympiade	(K) und (D) DM 24,90	Musik Master	(K) und (D) DM 22,90
The Way of the Exploding Flat	(K) DM 27,90	Micro Kalk	(K) und (D) DM 22,90
ACE (R&K Version)	(K) und (D) DM 32,90	Micro Datal	(K) und (D) DM 22,90
Commando	(K) DM 24,90	Paint Box	(K) und (D) DM 19,90
Yie are Kung Fu	(K) DM 24,90	Graphic Designer	(K) DM 15,90
International Karate	(K) DM 19,90	Turbo Tape	(K) DM 15,90
Ghost'n Goblins	(K) DM 27,90	C-16 Utilities	(K) DM 27,90
Project Nova	(K) DM 22,90		
Berlin Jack	(K) DM 19,90		
Frank Bruno's Boxing	(K) DM 19,90		
Plus Paket (4 Programme)	(K) und (D) DM 29,90		
European Games (Sportspiel)	(K) DM 19,90		
Trail Blazer	(K) DM 21,90		
Computer Hits 10 Vol. 3	(K) DM 27,90		
Robo Knight	(K) DM 9,90		
Legionair	(K) DM 15,90		
Mercenary (84k)	(K) DM 28,90		
Favourite 4 (4 Programme)	(K) DM 24,90		
Classics 1+ II (je 4 Programme)	(K) je DM 27,90		
King Size (50 Programme)	(K) DM 24,90		
Lawn Tennis	(K) DM 24,90		
Video Poker	(K) DM 9,90		
Quasi (Trivia)	(K) und (D) DM 23,90		

HARDWARE UND ZUBEHÖR	
ROM Modul 64 K (Einbau ohne Klein)	DM 109,00
Abdeckhaube C-16 hartplastik	DM 12,00
Abdeckhaube C-16 weichtplastik	DM 9,00
Abdeckhaube Floppy 1551/1541	DM 3,00
Abdeckhaube Diskette 1550/1581	DM 6,00
Datencassetten C-15	10 Stück DM 15,00
Disketten No Name 2D5.25	10 Stück DM 14,90
Thermotrucker TAP 1000 (auch f. Normsp.)	DM 389,00
Farbmischer Philips CM 9503	DM 489,00
Defettenbox f. 100 Disk 5 1/4 aufd. Sohl.	DM 24,90
Joystick Quickshot II+ (mit Microsch.)	DM 33,90
Joystick Quickshot II	DM 15,90
Joystickadapter	DM 9,90

Katalog mit Preisen und Programmbeschreibungen gegen DM 0,80 in Briefmarken. Versand erfolgt gegen Vorauskasse oder Nachnahme (zzgl. DM 2,60 + Zahlkartengebühr). An Versandkosten fallen an: Bis DM 50,- 2,50; Bis DM 100,- 3,90; Bis DM 300,- 5,90; Über DM 300,- 8,50

kleinanzeigen

C-16/116 Software!!!

20 Spiel- und Anwenderprogramme für nur 20,- DM. Die Programme sind auf Kasette. 20 DM als Schein oder Scheck bitte an: Daniel Beltlich, Kleine Hölbergstr. 6, 6000 Frankfurt 50 oder Info-Blatt für 80 Pf. in Briefmarken anfordern.

* C16 + 64K und Plus 4 *
* B Ö R S E *
* Ein Profi-Anwender-Programm in *
* hochauflösender Grafik, 15 Gra- *
* fiken, aktueller Kontost. u.v.m. *
* Auf Kasette für 8,50 DM + 1,50 *
* DM Porto + Versand 10 DM *
* Schein an: Manfred Leopold, *
* Hatzfelder-Str. 116, *
* 5600 Wuppertal 2 *
* Tel.: 0202/702876 *

Prelegünstige Programme aus Eigenentwicklung für alle Schneider-CPC-Computer (464/664/6128) erhalten Sie mit Schnellversand von Friedrich Neuper, Postfach 72, 8473 Pfreimd. Fordern Sie einfach das kostenlose Informationsmaterial an. Postkarte genügt!

C 16/116/Plus 4: TOP-Spiele + Anwender, Graphics + Musik zu Superpreisen. 150 Programme. Liste gegen Rückporto, DEMO-Cass. 6 DM. Ferner ab Januar 87 der Bombenhit: "USER" - Die C16 Zeitschrift auf Cassette. Mit Spielen, Anwender, Infos, Gratisanzeigen, Tips & Tricks, Preis 6 DM + 1,30 Porto. Anzeigentexte willkommen.

C16, C116 Superspiele!
Info gegen Rückporto bei Hannes Kaltenbach, Prielmayerstr. 16, 7990 Friedrichshafen 1

"C-16/116" Verkäufe 10 TOP-GAMES (z.B. Goldrausch, Submarine, Inv. of Space, Space Ship...) für 20,- DM auf Kasette. Info gratis. Einsenden an: Klaus-Peter Banko, Aite Poststr. 14, 4577 Nortrup, Tel: 05436/285
Suche Kontakte zwecks Erfahrungsaustausch!

C 128, C 64, C 16/116, Plus 4, VC 20
- 6 Spiele 19,90
- Textverarbeitung 12,90
- Adress, Karteikasten je 29,90
- Video-, Musikarchiv je 29,90
- Denktraining 19,90
- Vokaltrainer 19,90
- Katalog gegen 2x80 Pf-Marken bei: Computerservice Tino Hofstede, An der Windmühle 8, 5010 Berghelm 5

Verk. Orig. prg. f. C-64 auf Modul + Cass. ab 4,-! Schallplatten (LP 8,-, Single 3,-). Listen gegen 1,- in Bfm. Kaufmann, Burgstr. 61, 7000 Stuttgart 80

15 Superspiele für C-16 auf Kasette zu verkaufen! Ab 1.1.87 bei Kai Uwe Boß, Ilmenweg 5, 6422 Herbstein. 20DM in den Umschlag stecken und abschicken!!!

C-16, C-116 + Plus 4 Supersoftware, Info = 80 Pfennig. Anfordern bei Michael Husen, Ritterstr. 55, 2800 Bremen 21

10 C-16/116 Spiele für 10,- DM!!!
Mit: Pac-Man, Invaders of the Space, Cowboy-Duell, Tank-Wars, Turmspringen, SDI, Jäger des verlorenen Schatzes, Mimi, Cave, Submarine. Bitte frankierten Rückumschlag mit 10 DM an: Peter Preuss, Rimmelsweg 5, 7107 Bad Friedrichshall 2, P.S. Tausche auch Software!

Verkäufe C-64 Software (Nur Original-Programme) z.B.: Flight Simulator II 40,- DM, Print-Shop 35,- DM. Liste gegen Rückporto bei: Rudolf Lex, Prinzregentenstr. 100, 8000 München 80.

***** **C-64 TOP SOFTWARE** *****
*** Auf Diskette und Kasette! ***
* z.B. JET, PSI 5, ...! Traumpreise *
* Neuwertig! Bitte Liste anfordern! *
*** Notruf: 0491/61943 ab 16 Uhr ***

Biete Hardware

Verkäufe VC-20 mit 64K-Erweiterung + Su-per-Tool-Datasette-Anwenderhandbuch-Literatur + Spielprogramme sowie Spielkassetten für 300,-DM, Tel.: 02365/21446

Verkäufe wegen Systemwechsel C116 + 64K für 150 DM. Außerdem 21 Spielkassetten, z.B. Bomb Jack, Berks, Pluspaket, Favo-rite 4, Turbo Tape u.a. für 250 DM (wenig benutzt). Zusammen für nur 380 DM! Angebote an: Frank Roulands, Benrader-str. 135, 4154 Tönisvorst 1

Verkäufe Plus 4 nur 320,-DM, VC-20 nur 100,-DM, 16K-RAM 60,-DM, verkaufe für C-16: Gehäuse + Tastatur 25,- DM, TED-Chip 50,- DM, ROM-Chips je 15,-DM, Adress-Chip 15,-DM, Netzteil 15,-DM. Alle Preise + 6,-DM NN. Suche für Plus 4 EPROM-Brenner + Plotter 1520 je 130,-DM, evtl. Tausch. Christoph Drube, Kasseler Str. 28, 3530 Warburg, Tel.: 05641/1332

C-16/116/ + 4 Hardware & Zubehör
64K-Erweiterung & Schalter 49,50
64 K-Erweiterung o. Schalter 39,50
64 K-Erweiterung mit Einbau 65,50
Daten-Cassetten C15 5 Stück 7,80
Computape Dataset C15 11,50
Disketten Pen set 3 Stück 8,20
Endlospapier 1000 Blatt 28,50
Kunststoff-Pinzette 125 mm 0,90
Disketten 5 1/4" 10 Stück 12,80
Kopierhalter Format A4 47,50
Mikro-LötKolben 12V/8W 9,50
Daten Cassetten C15 1 Stück 2,20
Cassetten-Archiv für 24 Stück 6,50
EDV-Endlospostk. 250 Stück 22,50
Versand per Nachnahme
Fordern Sie bitte unseren Hauptkatalog an gegen 80Pf Porto.
PC-HC-ELEKTRONIC-TECHNIK
Fritz-Reuter-Str. 3, 2914 Barbel

Commodore C16 - 116- plus 4 Userport.... Wird einfach in das Expansionsport eingeschoben. Kompatibilität zum C64 gut möglich. 2 8Bit I/O/1 Serial-Port/2 16Bit Timer/1 Echtzeit-Uhr usw. für nur 99,- DM (incl. MwSt) + Port per NN bei: Frank Schmitz, Kurzer Weg 1, 5206 Neunkirchen 2, Tel.: 02247/8111

PLUS 4 ++ PLUS 4 ++ PLUS 4 ++ Datenfernübertragung jetzt nur ein Kinderspiel. DFÜ Komplettanlage bestehend aus Adapter, Modem und Supersoftware. Wählen von der Computertastatur:
* Wahlwiederholung
* Auto-Answer
* Mailboxbetrieb **229,-DM**
Ihr DFÜ-Spezialist
GUSS DATA CONNECTION,
Tel.: 02723/6679

Wir haben die Software, die Sie suchen!

Commodore 64/128

	Kassette	Diskette
Alleykat	29,95 DM	38,95 DM
Die Erbschaft	68,95 DM	78,95 DM
Eis & Feuer		68,95 DM
Hacker II	38,95 DM	54,95 DM
Knight Games	38,95 DM	38,95 DM
1942	32,95 DM	44,95 DM
Parallax	34,95 DM	
Room 10	29,95 DM	
Sanxion	32,95 DM	
Super Cycle	32,95 DM	44,95 DM
Tass Times		58,95 DM
Thrust	9,95 DM	
Trinity		98,95
Vera Cruz	48,95 DM	58,95 DM
Vietnam	32,95 DM	44,95 DM
Way of the Expl. Fist II	32,95 DM	
World Games	32,95 DM	44,95 DM

Schneider CPC (464, 664, 6128)

	Kassette	Diskette
Eden Blues	32,95 DM	44,95 DM
Jack, the Nipper	32,95 DM	32,95 DM
Saboteur	32,95 DM	32,95 DM
Star Strike II	32,95 DM	
Tarzan	32,95 DM	44,95 DM
Tau Ceti		32,95 DM
Tomahawk	32,95 DM	48,95 DM
Utchimata	32,95 DM	44,95 DM
Winter Games	32,95 DM	54,95 DM

Commodore 16/116/PLUS 4

	Kassette
ACE	32,00 DM
ACE (+ 64K)	34,95 DM
Bandits at Zero	14,95 DM
Bomb Jack	29,00 DM
Classics III	32,95 DM
Computer-Hit (10 Spiele)	32,95 DM
Finders Keepers	9,95 DM
Formula One Sim.	9,95 DM
Frank Bruno's Boxing	25,00 DM
Ghosts 'n' Goblins	29,00 DM
G-Man	9,95 DM
International Karate	19,95 DM
Jack Attack (Modul)	14,95 DM
Kikstart	9,95 DM
Matrix/Laser Zone	29,00 DM
Monty on the Run	9,95 DM
Paint Box	25,00 DM
Pod	9,95 DM
Project Nova	25,00 DM
Quiwi (+ 64K)	29,00 DM
ROM-Listing	29,00 DM
Space Pilot	19,95 DM
Speed King	9,95 DM
Text Manager (Disk)	38,95 DM
Text Manager (Kass.)	38,95 DM
Trail Blazer	19,95 DM
Winter Olympiade	29,00 DM
Yie Ar Kung-Fu	29,00 DM

Fordern Sie kostenloses Informationsmaterial über unser Liefer-Programm 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!



So bestellen Sie:

Der Mindestbestellwert von Software liegt bei 15,00 DMI

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 wochentags ab 9.30 Uhr für Sie zu erreichen!

SOFTWARE-VERSAND

ANDREAS BACHLER
POSTFACH 429
D-4290 BOCHOLT

☎ (0 28 71) 18 30 88

***Verkaufe VC-20** mit viel Zubehör (Speichererw. 64K u.a.) für 400,- DM VHB. Info bei Manfred Barth, Kandelgasse 3, 6509 Lonsheim

Wer braucht einen VC-20 gut erhalten?

V.B. 200,-DM. Angebote an: Marc Schwarzer, Höhenweg 30, 6992 Weikersheim, Tel. 07934/547

Verkaufe
C-16 + Datensette + Joystick + S/W-Monitor + 7TOP GAMES (z.B. Bomb Jack) Preis 500,-DM, Tel: 05602/5425, Walde-mar Glodny, Egerlandstr. 13, 3436 Hess. Lichtenau

SVI 328 64K + 10 Spiele 2 Joy mit Recorder + einf. Kassette. Anschl. an jedes TV-Gerät möglich.
Tel.: 09621/ 13939 Preis: DM 650,-

Verkaufe C-116 mit Joystick/Netzteil ca. 30 abget. Spiele, 8 gekaufte Spiele, z.B. Winter Olympiade, Atlantis, Bandits at Zero, Preis 250,-DM, bitte melden 04834/8322

Verkaufe Commodore C16 + Datensette und den Spielen Winter Olympiade, Space Pilot, Daley Thompsons Star Events, Plus Paket, Basic-Lernkassette und Joystick-adapt. Dazu 3 Handbücher über den C16. Preis 250,-DM. Martin Müller, Ost-str. 1, 8391 Hintereben.
Tel.: 08581/ 2261

Verkaufe C16 + Datensette + 12 Spiele + Joystick + BASIC-Cassette + BASIC-Buch! la-Zu-stand. VB-280,-DM, Tel.: 06195/4723

C-16 großer Bruder des C-64
Datenfernübertragung wird zum Kinderspiel! Erweiterungsplatine >USMO I< für C-16/116 mit RS-232 und zwei Userports für DM 138,-. Modem steckfertig mit passender Software:
* Wahl von der Computer-Tastatur
* Autoanswer
* Mailboxbetrieb DM 148,-
* Speichererweiterung DM 49,-
Guss Data Connection, Tel.: 02723/6679

C16/116 Speichererweiterung 64K
- Umrüstservice DM 74,-
- Steckbar ohne löten DM 65,-
- Bausatz kompl. DM 45,-
- Datensetteninterface DM 79,-
Preise + 6,-DM Versandkosten, Info gegen Freiumschlag, Tel.: 06203/43468 (14-17 Uhr)

+++ Verkaufe C-116 +++
Mit BASIC-Kurs, sowie Datensette und 31 Spiele: z.B. ACE, Bombjack, Reachhead, und viele mehr. Wo? Rainer Hüttinger, Kremser Str. 14, 8500 Nürnberg 50

Verkaufe Schneider CPC 664 mit Grünmonitor * Software oder Tausch gegen C64
Thorsten Mundil, Auf dem Klink 11, 3254 Emmerthal 2, Tel: 05157/473

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

Suche

Suche Tauschpartner/in für alle Arten von Programmen auf C 64 Disketten. Sendet Eure Listen an: Siegfried Braun, Zehntwiesenstr. 35, 7505 Ettlingen

Hallo, C-16-Besitzer! Wer hat Lust mit mir Spiele zu tauschen? Schreibt mir doch einfach einmal. Holger Greve, Itzehoeerstr. 15, 2351 Nutzt

VC-20 Speichererweiterung VC 1211 (Grafikhilfe) dringend gesucht. Willy Kreuzer, Weichselgasse 22, A-2500 Baden, Tel.: 02252/87970

Suche preiswerten Drucker für Commodore 64! Angebote an: Benno Frei, Geraniestr. 22, 7830 Emmendingen

Suche für Schneider CPC 464 Software auf Cass. oder 3" Diskette. Ob mit oder ohne deutsche Anleitung, bietet alles an. Schriftliche Angebote an Jörg Aglaster, Klaus-Dorfer Weg 10, 1000 Berlin 49

****CPC**CPC**CPC**CPC**CPC****
CPC-6128! Suche und tausche Programme auf 3"-Disc. Schreibt bitte an: Michael Fechtner, Cuxhavener Str. 13, 1000 Berlin 21

Suche Adventure Construction Set von Ariola-Soft: Angebote bitte an Gudlo Schmidt, Sonnenweg 6, 4901 Hiddenhausen

Lightpenprogramm für VC-20 und Compute mit 7/85 sucht Heinz Peter Lukes, Martingasse 11, A-2130 Mistelbach

I am searching for: Genesis, bitte 10,00 DM, Grandmaster, Football-Manager, Winter-Olympiade, Turbo-Tape, etc. für C-16 und Tape! R. Loepki, Heidrehmen 20, 2000 Hamburg 55

******* Hey C 64 Fans *******
Tauschpartner, kaufe auch für den C 64! Antwort ist 100%! Listen an: Christian Antons, Kettelerstr. 23, 4905 Spenge

****** AN ALLE C-16-FREUNDE ******
Suche Software! Tausche auch Kiste meiner Programme gegen Rückporto an Guido Bender, Malenstr. 8, 6602 Saarbrücken-Dudweiler

Suche Kopierprogramm für C 16/64K um Progr. von Cass. auf Disk zu kopieren. H. Schrimpf, Eikelcher Str. 4, 4782 Erwitte

Suche: C 16 oder C 116 mit Datensette und Joystick und Spielen. Angebote an: Martin Kraft, Tel.: 0231/679435 ab 14 Uhr. Preis: bis 200,- DM

Suche C-16 ohne Speichererweiterung bis 100,- DM sowie Drucker für C-16 bis 200,- DM. Angebote an: Dietmar Blom, Konenhoek 13, 4407 Emsdetten

Verschiedenes

+++ Achtung C-16/116 USER +++
Tausche Original-Software
Suche C-64 zahle bis 150,- DM mit Zubehör auch mehr! MELDEN bei: Guido Lange, Bergstr. 20, 8741 Schönau

Tausche/Verkaufe Spiele für C-64 (Disk). Habe Paperboy, World Games usw. Listen an Oliver Nazareth, Feldstr. 37, 2304 Laboe, Tel.: 04343/7847
Antwort 1000% XSUCHE SPEEDOSX

Tausche Software aller Art.
Bitte meldet Euch (neue und alte Software) Daniel Schmidt; Sindelenstr. 2, CH-8340 Hinwil

Anfänger sucht komplette Computer-Ausrüstung sowie Soft- und Hardware für VC-20. Hermann Starzengruber, Hofmark 95, A-4755 Zeu/Prum

Suche Tauschpartner für C-16/+4 Software! Frank Stefan, Ortelsburger Straße 23, 3400 Göttingen

Kontakte

C-16/116/+4 User-Club für Anfänger und Fortgeschrittene! Listings, Kurse, Sammelbestellung u.v.m. Info (50 Pf. Porto) bei Volker Huppert, Ferlingsweg 15, 4150 Okrefeld 1!

C16-Einsteiger sucht Kontakte mit Profi, suche auch billig Drucker. Wer hilft mir? Antwort bitte an: A.Z. Duda, Barbarossaring 32, 6750 Kaiserslautern, Tel.: 0631/40656 nur SA-SO

Größter C 16/116/P4 CLUB
Bietet einen tollen Service u. a.: Fachzeitschrift auf Datenträger! Informationen gegen 1,30 DM Rückporto. Hacker, im Winger 10, 5440 Mayen 14

Münster K2-Commodore C16-CLUB sucht Mitsretler!
* Selbstgestrickte Software * Tausch * 64K Erweiterung * Erfahrungsaustausch * Beratung. Info: 02501-58678

Ein Panasonic-Drucker KX-P 1000 -



NLQ - Near Letter Qualität · mehrere Schriftarten: Pica, Elite, Comp. · Endlospapier und Einzelblatt
Druckwegoptimierung · Textspeicher 1K · 100 Zeichen/sec · 8 Bit Parallel Schnittstelle
Komplett anschlussfertig für C-128/C-64, C-16 mit Wiesemann-Interface

DM 779,-

Die Welt am Draht - Dataphon s 21 d

Mit Mailboxbuch · Diskettensoftware für C-64 · Anschlussfertig für C-64/C-128
Holen Sie sich Daten und Software aus der ganzen Welt.
Internationale Datenbanken stehen zu Ihrer Verfügung.
Komplett anschlussfertig für C-64/C-128



DM 319,-

Fordern Sie unsere umfangreichen Infos und Hardwarelisten an. Kostenlos! Bitte geben Sie Ihren Computertyp an.

* Computer Soft- und Hardware *
* **Versandhandel R. Lindenschmidt** *
* Schulstraße 14 · Postfach 1328 · 4972 Löhne 2 · Telefon (05732) 72849 *

Das erste und einzig SOFTWARE-MAGAZIN

Die aktuelle Ausgabe gibt's jetzt überall

AS aktueller
SM software
markt

Nr. 9 Nov./Dezember 1986
OS 50
sfr 6,-
DM 6,-

Die Computer-Software-Fachzeitschrift

**Eine „feine Gesellschaft“:
Mörderstund'
ist ungesund!**
Ein Test von A bis Z

**Im Gespräch:
Cyborg**
Die Kampfmaschine

**Secret Service:
Jack the Nipper
in 50 Bildern!**

**Mitmachen:
Über 250 Preise
und eine Reise
nach
HOLLAND
zu gewinnen!**

**Anwendersoftware:
auf „Turbo-Trip“**

**Tests und Vorstellungen:
16 Seiten mehr!**

Die Computer-Software-Fachzeitschrift
Tests und Vorstellungen
Nr. 9 Nov./Dezember 1986 OS 50 / sfr 6,- / DM 6,-
aktueller software markt

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