

Eines ist klar: Super-Games wird man ausschließlich unter Verwendung von Maschinensprache zustande bringen (s. unser großer Spielkurs in diesem Heft).

Und dennoch: Wem Assembler ein Buch mit sieben Siegeln ist, muß deswegen nicht seinen Programmierdrang unterdrücken: Unsere Basic-Erweiterung »Game-Basic« stellt 26 neue Basic-Befehle zur Verfügung (Tabelle). Mit etwas Geschick und Basic-Programmierwissen können Sie Ihrer Fantasie freien Lauf lassen und (fast) jede Spielidee verwirklichen.

Laden Sie das Programm mit:

LOAD "GAME-BASIC",8

Gestartet wird mit RUN.

Auf dem Bildschirm erscheint die neue Einschaltmeldung: Der C64 steht für Ihre Programmeingaben bereit. Alle wichtigen Adressen für Bildschirm, Farb-RAM, Sprites, Soundchip, Timer usw. bleiben unverändert. Der Bereich \$0800 (2048) bis \$0FFF (4095) ist für 32 Sprite-Muster reserviert. »Game-Basic« selbst braucht den Speicher ab Adresse \$1000 (4096) bis \$17B0 (6064). Der dahinterliegende Platz ab \$17B1 (6065) bis \$1FFF (8191) bleibt frei und läßt sich für eigene Maschinenprogramme nutzen. Der Zeichensatz wurde ab \$2000 (8192) plaziert. Dort läßt er sich problemlos ändern (Zeichen umdefinieren). Der Basic-Speicher für Ihre Spielprogramme beginnt jetzt bei \$2800 (10240) und erstreckt sich - wie gewohnt - bis \$9FFF (40959). Die RAMs unterm Interpreter- und Kernel-ROM werden von »Game-Basic« nicht benutzt. Achtung: Das Programm arbeitet ausschließlich im Textmodus: Es wird keine hochauflösende Grafik verwendet! Normalerweise sind Spiele nicht mit Hires-Grafik ausgestattet, sondern mit verändertem Zeichensatz (Spiellandschaften und Levels) - das gilt auch für die meisten professionellen Spiele.

fast

Game-Basic - Spiele selbst gemacht

aus dem Handgelenk

Die neuen Befehle

Alle Programme, die mit Befehlen des Basic 2.0 entwickelt wurden, laufen anstandslos mit der Basic-Erweiterung.

»Game-Basic« stellt für die neuen Anweisungen einprägsame Namen zur Verfügung, die sich nahezu selbst erklären. Sie können im Direkt- und Programm-Modus angewandt werden:

HELP

Gerade bei Basic-Erweiterungen ist es hilfreich, die Schreibweise der neuen Befehle ständig zur Hand zu haben. (bis man sich daran gewöhnt hat!). Alle neuen Befehle erscheinen auf dem Bildschirm (Abb. 1).

BASIC

Diese Anweisung deaktiviert »Game-Basic«. Der C64 verhält sich wieder normal. Mit »SYS 4096« läßt sich die Erweiterung erneut aktivieren. Achtung: Das gilt nur, solange dieser Speicherbereich nicht von anderen Programmen oder Daten überschrieben wurde!

RESET

Die Basic-Erweiterung wird neu gestartet: Es erscheint wieder die Einschaltmeldung. Alle veränderten Einstellungen (Farben, Sprites usw.) werden in den Originalzustand zurückversetzt. Ein Basic-Programm im Speicher bleibt dabei ungelöscht.

COLOR

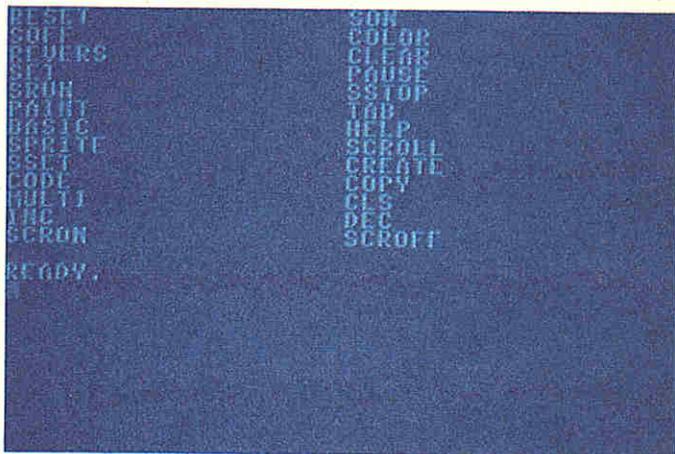
ändert die Rahmen-, Hintergrund- und Zeichenfarbe gemäß den entsprechenden Parametern, die hinter COLOR stehen müssen (s. Tabelle). Es gelten die Farbcodes von 0 bis 15. Die Anweisung ersetzt folgende Befehle:

Tolle Spiele mit geringem Programmieraufwand? Kein Problem für »Game-Basic«! Effektiv lassen sich Sprites und Bildschirm-Scrolling realisieren.

Bildschirm auch jederzeit nachträglich umfärben. Damit lassen sich reizvolle Effekte erzielen.

TAB

schaltet die übersichtliche Ausgabe eines Listings



[1] Alle neuen Befehle auf einen Blick: die HELP-Funktion bringt die neuen Anweisungen auf den Bildschirm

POKE 53281, Farbe

POKE 53280, Farbe

POKE 646, Farbe

Ein Beispiel für blauen Rahmen, weißen Hintergrund und grüne Schriftzeichen:

COLOR 6,1,5

SCROFF

schaltet den Bildschirm vorübergehend ab (während z.B. in der Zwischenzeit eine Bildschirm-Grafik durch Ihr Programm aufgebaut wird).

SCRON

Der Bildschirm wird wieder eingeschaltet.

Die beiden letztgenannten Befehle ersparen umständliche POKE-Anweisungen zur Manipulation des VIC-Chips.

PAINT

füllt den Farbspeicher fürs Bildschirm-RAM mit der angegebenen Farbe. Selbstverständlich kann man die Zeichen auf dem

auf dem Bildschirm an oder aus. Bei Programmstart ist die Funktion aktiviert (Abb. 2). Jeder Befehl wird in eine separate Bildschirmzeile geschrieben.

Achtung: Bei eingeschalteter TAB-Funktion läßt sich kein Basic-Listing vernünftig editieren - dazu muß sie abgeschaltet werden (dies erledigt ebenfalls der Befehl TAB!).

INC und DEC

»Inkrement« (Erhöhen) und »Dekrement« (Verringern) fungieren als Abwandlung des POKE-Befehls: Im Gegensatz zum POKEN wird kein bestimmter Wert in eine Speicherstelle geschrieben, sondern der darin befindliche Inhalt um den Parameter »eh« erhöht bzw. erniedrigt.

Beispiel: Der Inhalt von Speicherstelle 49152 soll um »12« erhöht werden:

INC 49152,12

PAUSE

ersetzt die umständliche Zeitschleife des Basic 2.0: FOR T=1 TO 1000: NEXT T Damit wird der Programmablauf ca. 1 s angehalten. Bei »Game-Basic« lautet die Anweisung: PAUSE 500

Bildschirmzeichen ändern

Jedes Spiel mit Profi-Touch kommt nicht ohne geänderten Zeichensatz aus. »Game-Basic« stellt auch dazu komfortable Anweisungen zur Verfügung:

CREATE

teilt dem C64 mit, welches Zeichen Sie ändern möchten. Dazu ist es nötig, den Modus »1« anzugeben. Beispiel:

Sie wollen aus dem »O« den Umlaut »Ö« machen.

CREATE 1,15
Der Modus »1« gilt für Zeichenänderungen, »0« bedeutet Sprite-Definitionen (s. dort). »15« ist der Bildschirmzeichen-Code für den Buchstaben »O«.

CODE

leitet die neue Bitmuster-Vergabe fürs neue Zeichen ein (unser Beispiel: Ö). Ein Punkt gilt als gelöscht Pixel, die »1« signalisiert, daß

30 CODE

40 CODE ..1111..

Ein weiteres Beispiel: Der Klammeraffe soll künftig als Copyright-Zeichen erscheinen.

10 CREATE 1,0
20 CODE ..111111.
30 CODE 1.....1
40 CODE 1..11..1
50 CODE 1.1....1
60 CODE 1.1....1
70 CODE 1..11..1
80 CODE 1.....1
90 CODE ..111111.

Nach der Tastenkombination <RUN/STOP RESTORE> oder der Anweisung RESET erscheinen allerdings die Zeichen wieder normal: Also Hände weg von diesen Tasten!

Selbstverständlich läßt sich der Zeichensatz im RAM auch durch die altbewährte POKE-Methode (mit DATA-Werten) verändern. Beachten muß man dabei allerdings, daß für einen neuen Zeichensatz nur die Großschrift/Blockgrafik-Zeichen zur Verfügung stehen (neun Blöcke auf Diskette!)

MULTI

Dieser Befehl gilt ebenfalls für Zeichen oder Sprites. Das regelt der Modus-Parameter »c« (s. Tabelle). Jetzt können noch zwei Zusatzfarben bestimmt werden. Ein Beispiel:

MULTI 1,5,2,6,7

Ab sofort erscheinen alle Zeichen auf dem Bildschirm in Grün (5) und Rot (2). Sprites dagegen besitzen die Farben Blau (6) und Gelb (7). Dennoch ist unumgänglich, das entsprechende Zeichen mit CREATE und CODE vorzubereiten. Eine farbige Version des kleinen »ö« sieht z.B. so aus (wir haben das englische Pfundzeichen dazu verwendet):

5 MULTI 1,5,2
10 CREATE 1,28
20 CODE ..11..11
30 CODE ..11..11
40 CODE ..11..11
50 CODE ..222222
60 CODE ..22..22
70 CODE ..22..22
80 CODE ..33..33
90 CODE ..333333

Die Codes der Bitmuster:
- 1: Zusatzfarbe 1,
- 2: Zusatzfarbe 2,
- 3: aktuelle Zeichenfarbe.
Achten Sie darauf, daß

die Codezahlen der Bitmuster immer paarweise angegeben werden (für Multicolor unabdingbar!). Sonst erhalten Sie eine Fehlermeldung.

Sprites entwerfen

Ähnlich wie bei den Zeichen spielt sich bei »Game-Basic« auch die Entwicklung von Sprites ab. Eine durch die Normalkonfiguration des VIC-Chips bedingte Einschränkung gilt hier ebenfalls: Es lassen sich nicht mehr als acht Sprites gleichzeitig auf dem Bildschirm aktivieren!

Bei CREATE ist nun der Modus »0« und der entsprechende Speicherbereich (Sprite-Block) anzugeben, in dem das Sprite-Muster abgelegt werden soll. Sie können die Blöcke 32 bis 63 verwenden. Den Beginn des Sprite-Blocks im Speicher erhalten Sie, wenn Sie die Blocknummer mit »64« multiplizieren. Außerdem besitzt eine Bit-Zelle für die CODE-Anweisung jetzt 24 Stellen (statt acht). 21 einzelne Bit-Zellen müssen entworfen werden. Ein Beispiel: Sie wollen ein Sprite ab Adresse 2048 (Spriteblock 32) definieren:

10 CREATE 0,32
20 CODE usw.

Im Listing unseres Programmbeispiels »Game-Basic-Game« finden Sie eine Menge Sprite-Definitionen, deren Aufbau man als Hinweis für eigene Kreationen betrachten sollte.

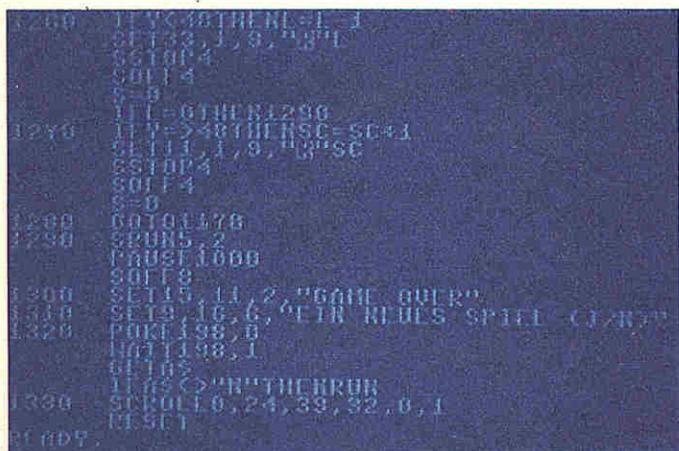
SPRITE

definiert ein Sprite, dessen Muster im entsprechenden Block steht. Folgende Parameter sind wichtig:

- Sprite-Nummer,
- Block,
- gedehnt in y-Richtung (vertikal),
- in x-Richtung vergrößert,
- Priorität vor oder hinter den Zeichen auf dem Bildschirm,
- Multicolor,
- Grundfarbe.

SSET

positioniert ein Sprite an gewünschter Stelle auf dem Bildschirm. Horizontale Werte von 0 bis 511 und ver-



[2] Übersichtliche Ausgabe eines Basic-Listings mit TAB. Diese Basic-Zeilen lassen sich nicht editieren.

Kurzinfo: Game-Basic

Programmart: Basic-Erweiterung
Laden: LOAD "GAME-BASIC",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Der Befehl COPY überträgt den Zeichensatz vom ROM ins RAM. Sprite-Bewegung im Interrupt.
Benötigte Blocks: 9
Programmautor: Ronald Meyer

COPY

verlegt den Originalzeichensatz vom ROM ins RAM (ab Adresse \$2000). Es ist auch möglich, nach dem COPY-Befehl einen geänderten Zeichensatz (nur Großschrift/Blockgrafik, 2048 Bytes) nachzuladen. Achtung: Die Startadresse dieser Zeichensatzdateien auf Diskette muß \$2000 (8192) sein!

Die Änderung eines beliebigen Zeichens läßt sich im Basic-Programm realisieren:

der entsprechende Bildpunkt eingeschaltet ist.

10 CREATE 1,15
20 CODE ..11..11.
30 CODE ..11..11.
40 CODE ..1111..
50 CODE ..11..11.
60 CODE ..11..11.
70 CODE ..11..11.
80 CODE ..11..11.
90 CODE ..1111..

Da viele Bitmuster-Zeilen mit dem Original-»O« übereinstimmen, würden auch folgende Änderungen genügen:

10 CREATE 1,15
20 CODE ..11..11.

tikale von 0 bis 255 sind möglich. »Game-Basic« berücksichtigt dabei automatisch das höchstwertige Bit im VIC-Register 16, das ab Bildschirmposition 256 Bedeutung erlangt. Beachten Sie, daß man Sprites erst dann auf dem Bildschirm sieht, wenn sie mindestens 24 Pixel von diesen Grenzen entfernt sind! Dazu geben Sie die Sprite-Nummer sowie die x- und y-Position an. Beispiel: Etwa in der Bildschirm-Mitte soll Sprite Nr.5 erscheinen.

SSET 5,184,142

SON

Das Sprite mit einer anzugebenden Nummer wird eingeschaltet.

SOFF

verbannt ein Sprite mit entsprechender Nummer wieder vom Bildschirm.

SRUN

Damit bewegt sich ein Sprite (oder mehrere) in eine von acht möglichen Richtungen. Der Interrupt übernimmt ab sofort die Sprite-Steuerung: Ihr Programm wird dadurch nicht aufgehalten. Damit kann ein Spr

te während der Bewegung die Gestalt verändern (z.B. explodieren). Sie können auch das Programm mit der STOP-Taste abbrechen: Die Sprites setzen ihre Bahn unbeirrt fort (da hilft nur <RUN/STOP RESTORE) oder

SSTOP

Auch hier muß die Sprite-Nummer angegeben werden. Dieser Befehl befiehlt dem Kobold auf dem Bildschirm, sofort anzuhalten. Möchten Sie alle acht Sprites stoppen, benutzen Sie: SSTOP 8

Komfortable Bildschirmverwaltung

»Game-Basic« verurteilt die inversen Cursor-Steuerzeichen des Commodore-Zeichensatzes zur Untätigkeit:

SET

ist ein erweiterter PRINT-Befehl zur Textausgabe an gewünschter Stelle auf dem Bildschirm. Möchten Sie z.B. das Wort »Text« mit weißen Buchstaben in der 18.



[3] Beispielprogramm »Gamebasic-Game«: Retten Sie die Fallschirmspringer!

Spalte und der 12. Zeile platzieren, lautet die Anweisung:

SET 18,12,1,"TEXT"

Mit dem üblichen PRINT-Befehl wären eine Unmenge Cursor-Steuerzeichen nötig. Außerdem ist die SET-Anweisung bedeutend schneller.

REVERS

Der aktuelle Inhalt des Bildschirms wird invers gezeit.

CLS

löscht den Bildschirm. Dieser Befehl ersetzt die Anweisung:

PRINT CHR\$(147)

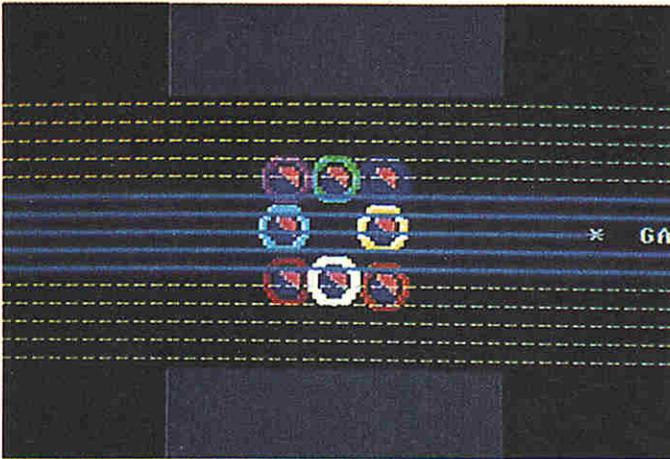
CLEAR

entfernt einzelne Zeilen oder Blöcke, entsprechend Ihrer Parameterangaben, vom Screen.

Ein Beispiel: Der Bildschirm soll zwischen der achten und einschließlich 13. Zeile gelöscht werden.

Game-Basic (Befehlsübersicht)

RESET	Neustart der Basic-Erweiterung. Programme im Speicher werden dabei nicht gelöscht.	SPRITE	bestimmt das Aussehen von Sprite Nummer
HELP BASIC	bringt alle neuen Befehle auf den Bildschirm Game-Basic wird eingeschaltet. Der Normalmodus ist wieder aktiv (Basic 2.0).	s,bk,ye,x,pr,mc-,f	»S«: - bl = Block (32 bis 64), - ye = doppelte Höhe (0 = nein, 1 = ja), - xe = doppelte Breite (0 = nein, 1 = ja), - pr = Priorität vor (= 0) oder hinter (= 1) dem Bildschirminhalt, - mc = Multicolor an (= 1) oder aus (= 0), - f = Spritefarbe.
COLOR rf,hf,zf	Rahmen- (rf), Hintergrund- (hf) und Zeichenfarbe (zf) setzen	SSET s,x,y	setzt Sprite Nummer »s« an Position x (0 bis 511!) und y (0 bis 255)
REVERS PAINT f	invertiert den aktuellen Bildschirm färbt den Bildschirminhalt mit der Farbe f (Werte von 0 bis 15)	SRUN s,ri	Sprite »s« bewegt sich in die angegebene Richtung:
CLS	Bildschirm löschen	SSTOP s	Sprite »s« hält wieder an
SCROFF	Bildschirm abschalten	MULTI	Multicolormodus ein-/ausschalten:
SCRON	schaltet Bildschirm wieder ein	c,z1,z2,s1,s2	- c: 0 = Multicolor für Zeichen an, 1 = Multicolor für Zeichen aus, 2 = Angabe von Zusatzfarben, Bei Verwendung von 0 und 1 muß man die Zusatzfarben nicht angeben. - z1: Zusatzfarbe 1 (Zeichen), - z2: Zusatzfarbe 2 (Zeichen), - s1: Zusatzfarbe 1 (Sprites), - s2: Zusatzfarbe 2 (Sprites).
CLEAR an,en	Bildschirm wird ab Zeile »an« bis Zeile »en« gelöscht	CREATE m,bf	neues Zeichen oder Sprite definieren: - m: Mode (0 = Sprite, 1 = Zeichen), - b1: Spriteblock oder Zeichencode.
COPY TAB	Zeichensatz ins RAM kopieren	CODE Bitfolge	Nur in Zusammenhang mit dem CREATE-Befehl anwenden! Bei Sprites muß man 24, bei Zeichen acht Bits eintragen. - . = Hintergrundfarbe, - 1 = Farbe des Zeichens/Sprites (Multicolormodus Zusatzfarbe 1), - 2 = Zusatzfarbe 2, - 3 = Farbe des Zeichens/Sprite bei Multicolormodus.
PAUSE t	schaltet formatierten LIST-Modus ein/aus Das Programm hält t x 0,002 s an. Beispiel: »PAUSE 500« unterbricht den Programmablauf eine Sekunde.		
INC ad,eh	addiert den Wert »eh« zum Inhalt der Adresse »ad«		
DEC ad,eh	subtrahiert die Zahl »eh« vom Wert in der Speicherstelle »ad«		
SET s,z,f,"text"	gibt eine Zeichenkette »text« in der Spalte »s« und der Zeile »z« in der Farbe »f« aus.		
SCROLL az,ez,s,z,f,r	scrollt den Bildschirmbereich von der Anfangszeile »az« bis zur Endzeile »ez« in Richtung »r« (0 = rechts, 1 = links). Dabei werden Zeichen mit dem Code »z« in der Farbe »f« nachgeschrollt.		
SON s	schaltet Sprite mit der Nummer »s« ein (0 bis 7)		
SOFF s	das Sprite mit der Nummer »s« wird wieder ausgeschaltet		



[4] Unser Demoprogramm zeigt die komfortablen Funktionen der neuen Basic-Befehle

CLEAR 7,12

Beachten Sie, daß die Zeilenwerte zwischen »0« (1. Zeile) und maximal »24« (25. Zeile) liegen dürfen.

SCROLL

verschiebt oder verändert beliebige Bildschirmbereiche. Eine nähere Beschreibung der Parameter finden Sie in der Tabelle. Mit dem Befehl können z.B. Laufschriften simuliert werden.

Demoprogramme auf Diskette

Auf der Diskette zu diesem Sonderheft gibt's ein Basic-Spiel, das ausschließlich mit den neuen Befehlen dieser Befehlsenerweiterung konzipiert wurde: »Gamebasic-Game«. Achten Sie darauf, daß »Game-Basic« aktiv ist, denn das normale Basic 2.0 versteht die neuen Befehle nicht.

Laden Sie das Programm mit:

```
LOAD "GAMEBASIC-GAME",8
```

und starten Sie es mit RUN. Ein neuer Zeichensatz (ZSA) wird nachgeladen.

Die Spielidee ist rasch erklärt: Sie sind Kapitän eines Flugzeugträgers, der auf hoher See aus Düsenjägern abspringende Fallschirmjäger aufnehmen soll (Abb. 3). Das Schiff befindet sich am unteren Bildschirmrand in stürmischer See und läßt sich mit dem Joystick in Port 2 nach links und rechts steuern. Achtung: Die Reaktion

des Joysticks wurde absichtlich erschwert, damit das Spiel nicht zu einfach ist. Oben zischen in rascher Folge die Düsenjäger vorbei und werfen die Fallschirmspringer ab, die Sie durch geschicktes Manövrieren des Schiffs wohlbehalten aufnehmen müssen.

Am oberen Bildschirmrand sieht man unter der Anzeige »Score«, wie viele Springer sicher gelandet sind. Wenn einer den Kreuzer verfehlt, verlieren Sie ein Leben (Anzeige »Lives«). Bei Programmstart stehen Ihnen drei zur Verfügung.

Wenn die Spielrunde zu Ende ist, scrollt der Bildschirm nach rechts. Mit der Taste <J> können Sie ein neues Spiel beginnen, bei <N> führt »Game-Basic« den Befehl RESET aus: Das Spiel steht allerdings unversehrt im Speicher und kann jederzeit erneut mit RUN gestartet werden.

In den Basic-Zeilen 100 bis 810 werden mit den neuen Befehlen CREATE und CODE vier Sprite-Muster definiert:

- Zeile 100 bis 250: Flugzeug von links,
- Zeile 260 bis 410: Flugzeug von rechts,
- Zeile 420 bis 630: Fallschirmspringer,
- Zeile 640 bis 810: Schiff.

Die Zeichen CHR\$(164) und CHR\$(165) ändern durch die Basic-Zeilen 820 bis 1000 ihr Aussehen: Sie dienen jetzt als Füllmuster für die Wellenbewegung des

Meeres, die mit der Anweisung SCROLL erzeugt wird (Zeile 1170). Den stetigen Flug der Düsenjäger über den Bildschirm bewirkt der Befehl »SRUN« in den entsprechenden Basic-Zeilen. Im Bereich 1050 bis 1100 werden die Sprites eingeschaltet (SPRITE), ab Zeile 1110 positioniert (SSET). Daß die Sprites bei Erreichen der bekannten Position »256« nicht vom Bildschirm verschwinden, verdanken wir den Befehlen in den Programmzeilen 1190 bis 1200. Dabei bildet die Variable »V« die Basic-Adresse des VIC-Chip: 53248. Die Joystick-Abfrage zur Bewegung des Rettungskreuzers nach links oder rechts beginnt ab Zeile 1210. Der gesamte Spielablauf (Sprite-Bewe-

Fallschirmspringers um z.B. 1000 Zähler klettern, müssen Sie die Addition in Zeile 1270 um »1000« erhöhen (statt nur um »1«):

```
IF Y=>48 THEN SC=SC+1000
```

Eine Fülle von Bildschirm- und Sprite-Manipulationen zeigt Ihnen das nächste Beispielprogramm:

```
LOAD "GAMEBASIC-DEMO",8
```

Es bietet nahezu alle Funktionen, die »Game-Basic« zur Verfügung stellt: Sprite-Bewegung in alle Richtungen, Bildschirm-Scrollen, Farbwechsel und raffiniertes Löschen bestimmter Bildschirmbereiche (Abb. 4).

Studieren Sie aufmerksam die Listings der Demoprogramme und experimentieren Sie mit »Gamebasic-Game« (denkbar wäre statt



[5] Der kleine Rockford auf Diamantsuche: »Boulder Dash« ist einer der vielen Spiele-Klassiker des C 64!

gung, Wellengang, Regulierung der Anzeigen »Score« und »Lives«) wird im Prinzip nur durch die Zeilen 1170 bis 1280 geregelt: Darin liegt der Kern des Spielprogramms. Wenn Sie bei Programmende die Frage nach einem neuen Spiel verneinen, schiebt die SCROLL-Anweisung in Zeile 1330 den gesamten Bildschirm nach rechts und kehrt mit RESET zum Einschaltbildschirm von »Game-Basic« zurück.

Falls Ihnen die eingestellten Werte für »Score« und »Lives« zu gering sind, läßt sich das jederzeit ändern: Die Anzahl der verfügbaren Leben steht in der Variablen »L« (Programmzeile 1160). Soll der Score bei jeder erfolgreichen Bergung eines

eines Schiffs auch ein Lastwagen, statt Flugzeugen könnten z.B. große Vögel Früchte abwerfen, die der Trucker aufsammeln muß). Dazu müssen nur die Sprite-Definitionen in den entsprechenden Zeilen geändert und das Programm neu gespeichert werden.

Ihrer Fantasie sind keine Grenzen gesetzt: Mit »Game-Basic« werden selbstprogrammierte Spiele zum Vergnügen! Diese Basic-Erweiterung bietet viele Möglichkeiten, auch professionell funktionierende Spiele zu entwickeln. Wie wär's z.B. mit einer eigenen Version von »Boulder Dash« (Abb. 5)? Viel Vergnügen beim Programmieren!

(Ronald Meyer/bl)