

Zwei Quickies

Kopfstand für Sprites und ein kleiner Zeicheneditor sind zwei Einzeiler, die sich in Programme einbinden lassen.

Der folgende Einzeiler stellt ein Sprite auf den Kopf (beziehungsweise wieder auf die Füße):

Kopfstand für Sprites

```
FOR A=958 TO 896 STEP-3:POKE A-2,PEEK(832+L):
POKE A-1,PEEK(833+L):POKE A,PEEK(834+L):
L=L+3:NEXT
```

Der Einzeiler kann nur dann als solcher eingegeben werden, wenn von der Möglichkeit Gebrauch gemacht wird, die (Basic-)Kurzbeefehle einzutippen (siehe Anhang D im C 64-Handbuch). Sie können jedoch auch zwei Zeilen daraus machen. Das Sprite, das verändert werden soll, muß im Bereich 13 (Adresse 832-895) abgelegt sein; im Bereich 14 (Adresse 896-958) steht nach Programmablauf das gedrehte Sprite beziehungsweise dessen Bitmuster.

- A = Laufvariable für FOR...NEXT-Schleife
- L = Zählvariable

Da L nicht auf Null gesetzt wird, muß vermieden werden, diese Variable anderweitig im (Gesamt-) Programm zu verwenden, da es sonst zu unliebsamen Überraschungen kommen kann. (Hans Joachim Neumann/gk)

Zeichendefinitionen

Folgendes kurze Programm wandelt ein am Bildschirm entworfenen Zeichen in Zahlen um und schreibt diese dann auf den Bildschirm. Im linken oberen Eck des Bildschirms wird das Zeichen in einer 8 * 8-Matrix entworfen. Um einen gesetzten Punkt zu markieren, wird an die gewünschte Stelle ein Stern (»*«) geschrieben.

- X(0-7) =Zahl 0-7
- T Zähler der Reihe
- Y Zähler des Bits
- 1031 =linke obere Ecke des Zeichens+ 7 (1024+7)
- 42 Bildschirmcode für »*« — es können auch andere verwendet werden.

Damit Sie nicht mit Abkürzungen arbeiten müssen, ist der Einzeiler auf zwei Zeilen aufgeteilt worden (Zeile 100 und 110).

Starten Sie das Programm mit RUN. In der oberen linken Ecke wird ein Raster von 8 * 8 Punkten gezeichnet und unter dem READY erscheint GOTO 100. Mit den Cursor-Tasten (nicht die RETURN-Taste verwenden) können Sie nun ein Zeichen definieren, indem Sie anstelle eines Punktes ein »*« setzen. Wenn das Zeichen fertig ist, gehen Sie mit dem Cursor auf »GOTO 100« und drücken RETURN. Auf dem Bildschirm werden jetzt die Daten dieses Zeichens angezeigt.

Diese Daten können zum Erstellen eines eigenen Zeichensatzes (auch für Spiele) verwendet werden. Ein manuelles Errechnen dieser Werte entfällt somit.

(Wolfgang Horak/gk)

```
7 REM ZEICHENDEFINIEREN IN DER OBEREN <107>
8 REM LINKEN BILDSCHIRMECKE <174>
10 PRINT "{CLR}"; <204>
20 FOR I=1 TO 8:PRINT".....":NEXT <239>
30 PRINT "{3DOWN}GOTO 100 " <253>
40 END <042>
100 FOR T=0 TO 7:X(T)=0:FOR Y=7 TO 0 STEP- <
1:X(T)=X(T)-2+Y*(PEEK(1031-Y+40*T)=42 <055>
):NEXT Y
110 PRINT X(T);:NEXT <200>
@ 64'er
```

3D-Darstellung in 19 Zeilen

Das Programm stellt ein frei wählbares, dreidimensionales Objekt auf dem Bildschirm dar, das durch Drücken der Funktionstasten gedreht werden kann. Simons Basic ist allerdings Voraussetzung.

Das Programm (Listing 1) wird nach Aktivierung von Simons Basic normal geladen und mit »RUN« gestartet. Danach muß die Datei angegeben werden, die die Daten für den darzustellenden Körper enthält. Bevor die Eingabe mit RETURN abgeschlossen werden kann, muß jedoch erst die Diskette mit der gewünschten Datei in das Laufwerk eingeführt werden. Sind alle Werte eingelesen, erscheint die Figur in Originalform auf dem Bildschirm. Sie kann nun durch Drücken der Funktionstasten (ohne SHIFT) beliebig gedreht werden. F1, F3 und F5 bestimmen die Rotationsachse, während F7 einen Richtungswechsel bewirkt.

Die Erzeugung von Figuren

Die Daten für Figuren müssen in Form einer sequentiellen Datei (Typ SEQ) vorliegen. Eine Figur besteht dabei aus einer begrenzten Anzahl von Geradenstücken, die durch ihre Anfangs- und Endpunkte (x-, y- und z-Werte) gekennzeichnet sind. Jedes Geradenstück wird also durch sechs Zahlen bestimmt. Alle Geradenstücke werden unmittelbar hintereinander in die Datei geschrieben. Ein Endekennzeichen oder sonstige Zusatzangaben sind nicht erforderlich.

Das Demonstrationsprogramm (Listing 2) erzeugt eine Datei mit dem Namen »PYR«, welche die Daten für eine Pyramide enthält. Durch Veränderung der DATA-Zeilen können auch andere Körper definiert werden. Die Koordinaten müssen jedoch so gewählt werden, daß sie nicht außerhalb des Bildschirmbereichs liegen und diesen auch nicht durch die Rotation zu verlassen drohen. Es sei dabei darauf hingewiesen, daß die z-Achse in die Tiefe und die y-Achse nach oben zeigt. Als Orientierungshilfe kann die Demo-Pyramide verwendet werden.

Darstellung eines Punktes

Um einen Raumpunkt auf dem Bildschirm darstellen zu können, muß er zunächst an das zweidimensionale Koordinatensystem angepaßt werden. Es wird hierzu die Fluchtpunktperpektive gewählt, die über die Strahlensätze herleitbar ist:

- $x' = xf - (zf - z) * (xf - x) / zf$
- $y' = yf - (zf - z) * (yf - y) / zf$
- x' horizontale Bildschirmkoordinate
- y' vertikale Bildschirmkoordinate
- x,y,z Koordinaten des Raumpunktes
- xf,yf,zf Koordinaten des Fluchtpunktes

Anwendungsmöglichkeiten

Das Programm kann beispielsweise zur Darstellung und räumlichen Drehung von Computergrafiken (dreidimensionale Funktionen, Gegenstände, Flächen etc.) verwendet werden. Die Erzeugung der Gebilde erfolgt dabei von einem unabhängigen Programm. Nach Erstellung der sequentiellen Datei kann das Ergebnis mittels des Rotationsprogramms betrachtet werden. Dazu kann es sinnvoll sein, noch einige Erweiterungen in das Programm einzubauen. Zum Beispiel