



# Sprites schneller bewegen

Daß Sprites nützliche Hilfsmittel nicht nur bei Spielen sind, ist mittlerweile bekannt. Mit dem hier vorgestellten Programm lassen sich bis zu acht Sprites mit 255 verschiedenen Geschwindigkeiten kontinuierlich in acht Richtungen manövrieren. Das Besondere: Berechnungen im Programm haben keinen Einfluß auf die Laufgeschwindigkeit der Sprites.

Eine feine Sache ist beim Commodore 64 die Sprite-Grafik. Hat man ein solches Sprite entwickelt und auf dem Bildschirm dargestellt, so kann durch Veränderung zweier Speicherstellen das Sprite an eine andere Position gebracht werden. Um eine fließende Bewegung zu erreichen, ändert man die Speicherstellen um den Faktor 1. Nun funktioniert dies bei einem einzigen Sprite noch recht zügig. Möchte man aber mehrere Sprites bewegen, zwischendurch etwas berechnen und abfragen, ob eine Kollision zwischen zwei Sprites stattgefunden hat, so werden die Bewegungen doch recht träge. Um die Bewegungen schneller erscheinen zu lassen, kann man die Schrittweite erhöhen. Dies hat aber den Nachteil, daß die Sprites ziemlich »abgehackt« über den Bildschirm laufen. Weiterhin muß beachtet werden, ob ein Sprite auf der rechten Bildschirmseite dargestellt werden soll. Ist dies der Fall, so muß im Register 16 das Bit für das entsprechende Sprite gesetzt werden. Bei der Rückkehr auf die linke Seite muß das Bit wieder gelöscht werden. Bild 1 zeigt den Bereich, in dem sich die Sprites bewegen können. Wenn Sie das

```

5 REMSPRITE-MOVE
10 POKE183,11:POKE185,0:POKE186,1
20 POKE187,PEEK(43)+5:POKE188,PEEK(44)
30 POKE193,77:POKE194,174:POKE174,59
40 POKE175,207:SYS62954
50 REM ZEILE 5 BITTE MIT EINGEBEN
60 REM LADEN ERFOLGT MIT LOAD"SPRITE-MOVE",1,1
70 REM NACH DEM LADEN NEW EINGEBEN
    
```

Bild 3. Mit diesem kurzen Programm lassen sich die Sprites abspeichern

Programm SPRITE-MOVE (siehe Listing) benutzen, so sind diese Probleme vorbei. Sie können ein Sprite nun mit dem Befehl !RUN S,R,G auf die Reise schicken. Mit dem S geben Sie die Nummer des Sprites an (0-7), mit R die Richtung (1-9) und mit G die Geschwindigkeit (0-255). In Bild 2 können Sie sehen, welche Zahl für welche Richtung einzusetzen ist. Die

```

200 PRINT "g":V=53248:SYS 49700
210 FOR N=0 TO 125:READ Q:POKE 49900+N,Q:NEXT
215 FOR I=2040 TO 2047:POKE I,13:NEXT
220 FOR I=0 TO 7:POKE V+29,255
230 G=V:FOR I=0 TO 7:X(I)=G:G=G+1:Y(I)=G:G=G+1:NEXT
240 FOR I=0 TO 7:POKE X(I),120:POKE Y(I),120:NEXT
250 FOR I=0 TO 7:POKE V+21,255
260 !VAL1,0:POKE V+39+1,1:NEXT
270 !RUN 0,2,5:!RUN 1,1,8:!RUN 2,3,10:!RUN 3,4,0
280 !RUN 4,6,255:!RUN 5,7,35:!RUN 6,8,20:!RUN 7,9,4
290 FOR I=0 TO 200:NEXT :!VAL1,1
300 FOR I=0 TO 1500:NEXT :!VAL1,0
310 FOR I=0 TO 7:POKE V+21,1:R=8
320 POKE I=0 TO 7:STOP I:NEXT
330 !RUN 0,R,50
340 FOR I=0 TO 200:NEXT :!VAL1,1
350 IF PEEK(Y(0))<50 OR PEEK(Y(0))>200 THEN R=10-R
360 GOTO 325
1000 DATA .....32,32,32,16,80,64,8,112,128,7,255,1,252,1,136,1,140
1010 DATA .....32,80,112,127,255,240,1,252,1,136,1,140
2000 DATA .....
2010 DATA .....
    
```

Bild 4. Dieses Programm dient als Beispiel für Sprites mit Bewegungseffekt

Zahl 5 kann eingegeben werden, hat aber keinen Einfluß. Bei der Geschwindigkeit G bedeutet 0 die schnellste und 255 die langsamste Bewegung. Die Sprites bewegen sich nun immer in der eingegebenen Geschwindigkeit. Sie können in Ihrem Basic-Programm berechnen oder abfragen so viel Sie wollen, dem Bewegungsdrang der Sprites tut dies keinen Abbruch. Verläßt ein Sprite den Bereich, in dem es sich bewegen kann (siehe Bild 1), so erscheint es wieder auf der gegenüberliegenden Seite. Wenn Sie dies nicht möchten, so müssen Sie durch Abfrage der entsprechenden Speicherstellen darauf reagieren. Anhalten können Sie ein Sprite mit dem Befehl !STOP S.

Aufpassen muß man bei dem Befehl THEN. Zwischen THEN und dem neuen Befehl muß ein Doppelpunkt stehen, da es sonst zu einem SYNTAX-ERROR kommt.

## Aufbau einer Spritebibliothek

Weiterhin haben Sie die Möglichkeit, ab der Speicherstelle 49900 bis zu 50 Sprites zu speichern und mit dem Befehl

```
!VAL B1,B2
```

in den Block zu schieben, auf dem Ihr Sprite zugreift. B1 kann dabei Werte von 0 bis 7 annehmen. Und zwar bedeutet:

0 = Block 11  
ab Speicherstelle 704

1 = Block 13  
ab Speicherstelle 832

2 = Block 14  
ab Speicherstelle 896

3 = Block 15  
ab Speicherstelle 960

4 = Block 32  
ab Speicherstelle 2048

5 = Block 33  
ab Speicherstelle 2112

6 = Block 34  
ab Speicherstelle 2176

7 = Block 35  
ab Speicherstelle 2240

Wenn Sie die Blöcke 32

bis 35 benützen, müssen Sie den Zeiger, der auf den Beginn des Basic-Programms zeigt, ändern. Mit POKE 2560,0 beginnt Ihr Basic-RAM-Bereich ab Speicherstelle 2561. Anschließend geben Sie noch NEW ein.

## Bringen Sie Bewegung in die Sprites

B2 kann Werte von 0 bis 49 annehmen. Errechnen können Sie die Anfangsspeicherstelle für B2 mit  $49900 + B2 * 63$ . Sie können sich so eine ganze Sprite-Bibliothek anlegen und bei Bedarf abrufen. In Bild 3 sehen Sie ein Programm, mit dem Sie die Sprites (die ab Speicherstelle 49900 stehen) zusammen mit dem Programm SPRITE-MOVE auf Kassette wegspeichern können. Denkbar wäre auch, daß man sich ähnlich aussehende Sprites speichert und diese abwechselnd einschaltet. Dadurch kann man den Eindruck einer sich bewegenden Figur erzeugen. (Siehe Beispielprogramm in Bild 4).

Alle Parameter für die neuen Befehle können Zahlen oder auch Variablen sein. Außerdem funktionieren die Befehle sowohl im Direktmodus als auch innerhalb eines Basicprogramms.

## Sprites for ever

Das Programm steht von Speicherstelle 49229 bis 49798. Die Speicherstellen 49152 bis 49228 werden vom Programm als Speicher benutzt. Aktiviert werden die neuen Befehle mit SYS 49700. Im Basic-Lader geschieht dies in Zeile 172. Sollten Sie die RUN/STOP- und RESTORE-Taste betätigen, so müssen Sie das Programm neu aktivieren. Und nun viel Spaß mit den neuen, schnellen Sprites.

(Herbert Kunz)

Basic-Lader von »Sprite-Move« ▶

```
100 REM ***** SPRITE-MOVE *****
101 DATA 162,0,189,0,192,201,1,240,8
102 DATA 232,224,8,208,244,76,76,193
103 DATA 56,189,26,192,233,1,157,26,192
104 DATA 144,3,76,86,192,189,16,192,157
105 DATA 26,192,142,24,192,189,8,192
106 DATA 201,1,240,31,201,2,240,42,201
107 DATA 3,240,50,201,4,240,61,201,6
108 DATA 240,69,201,7,240,77,201,8,240
109 DATA 88,201,9,240,96,76,86,192,32
110 DATA 7,193,32,26,193,32,39,193,174
111 DATA 24,192,76,86,192,32,7,193,32
112 DATA 39,193,174,24,192,76,86,192
113 DATA 32,7,193,32,13,193,32,39,193
114 DATA 174,24,192,76,86,192,32,7,193
115 DATA 32,26,193,174,24,192,76,86,192
116 DATA 32,7,193,32,13,193,174,24,192
117 DATA 76,86,192,32,7,193,32,26,193
118 DATA 32,48,193,174,24,192,76,86,192
119 DATA 32,7,193,32,48,193,174,24,192
120 DATA 76,86,192,32,7,193,32,13,193
121 DATA 32,48,193,174,24,192,76,86,192
122 DATA 173,24,192,10,168,96,24,177
123 DATA 247,105,1,145,247,144,3,76,57
124 DATA 193,96,56,177,247,233,1,145
125 DATA 247,176,3,76,57,193,96,200,24
126 DATA 177,247,105,1,145,247,96,200
127 DATA 56,177,247,233,1,145,247,96
128 DATA 174,24,192,169,1,202,48,4,10
129 DATA 76,62,193,77,16,208,141,16,208
130 DATA 96,206,40,192,48,5,162,0,76
131 DATA 79,192,169,6,141,40,192,76,49
132 DATA 234,32,115,0,201,33,240,9,32
133 DATA 121,0,76,231,167,76,8,175,32
134 DATA 115,0,201,138,240,29,201,144
135 DATA 240,7,201,197,240,81,76,107
136 DATA 193,32,115,0,32,158,183,224
137 DATA 8,176,225,169,0,157,0,192,76
138 DATA 174,167,32,115,0,32,158,183
139 DATA 224,8,176,207,169,1,157,0,192
140 DATA 142,41,192,32,253,174,32,158
141 DATA 183,224,10,176,189,138,240,186
142 DATA 201,5,240,6,174,41,192,157,8
143 DATA 192,32,253,174,32,158,183,138
144 DATA 174,41,192,157,16,192,157,26
145 DATA 192,76,174,167,32,115,0,32,158
146 DATA 183,224,8,144,3,76,8,175,142
147 DATA 47,192,32,253,174,32,158,183
148 DATA 224,50,176,240,232,142,46,192
149 DATA 173,47,192,10,170,189,48,192
150 DATA 133,251,232,189,48,192,133,252
151 DATA 32,126,194,173,46,192,170,202
152 DATA 240,14,24,165,249,105,63,133
153 DATA 249,144,244,230,250,76,3,194
154 DATA 160,0,177,249,145,251,200,192
155 DATA 64,208,247,76,174,167,234,234
156 DATA 169,94,141,8,3,169,193,141,9
157 DATA 3,169,77,141,20,3,169,192,141
158 DATA 21,3,169,0,133,247,169,208,133
159 DATA 248,169,192,141,48,192,141,54
160 DATA 192,141,62,192,169,2,141,49
161 DATA 192,169,64,141,50,192,141,58
162 DATA 192,169,3,141,51,192,141,53
163 DATA 192,141,55,192,169,128,141,52
164 DATA 192,141,60,192,169,0,141,56
165 DATA 192,169,8,141,57,192,141,59
166 DATA 192,141,61,192,141,63,192,169
167 DATA 236,133,249,169,194,133,250
168 DATA 96
169 S=0:FOR I=49229 TO 49798:READ D
170 POKE I,D:S=S+D:NEXT
171 IF S<>69859 THEN PRINT "BEFehler!":STOP
172 SYS 49700
```