

Sprites ohne Streß

Es soll immer noch Leute geben, die sich nicht an die Programmierung von Sprites herantrauen, weil sie von den recht unverständlichen Ausführungen im Commodore-Handbuch ziemlich verunsichert sind. Deswegen wollen wir hier die Grundlagen der Sprites darstellen.

Spielfreaks haben sie lieben gelernt: die Sprites, frei auf dem Bildschirm bewegbare Objekte in hoher grafischer Auflösung, die sogar mehrfarbig sein können. Doch nicht nur in Videospiele haben sich die Sprites bewährt: Auch als Mini-Grafik, Erkennungszeichen oder verschnörkelter Schriftzug waren sie schon zu sehen, und jede Woche entdeckt irgend-ein findiger Kopf eine Anwendungsmöglichkeit mehr.

Wer Sprites als grafische »Auflockerung« seiner Programme verwenden will, der kann mit dem ganz normalen Commodore-Basic hervorragende Ergebnisse erzielen, obwohl man dabei nicht um den berühmt-berüchtigten POKE-Befehl herumkommt. Wer allerdings ein rasantes Actionspiel programmieren will, der verlege sich doch besser auf Maschinensprache, denn die Bearbeitung von Sprites durch Basic ist dafür nicht schnell genug, es sei denn, man hat eine Basic-Erweiterung, die Sprites direkt unterstützt.

Drum also hier alles, was man wissen muß, um Sprites in den verschiedensten Variationen auf die heimische Mattscheibe zu bannen.

Am Anfang ist die Idee

Wer mit Sprites arbeiten möchte, der kommt um eines nicht herum: Ein Sprite-Entwurf muß angefertigt werden. Und das bedeutet zuerst einmal, daß man kariertes Papier sowie Bunt-, Filz- oder Bleistift aus der Schublade hervorkramt. Dann malt man einen Rahmen, der genau 24 mal 21 Kästchen einschließt. Vor sich hat man nun ein sogenanntes Sprite-Entwurfsblatt liegen. Zugegeben, einige Details fehlen noch, aber zumindest können Sie damit schon einmal genau festlegen, wie Ihr Sprite später aussehen wird.

Ein vollständiges Sprite-Entwurfsblatt sehen Sie in Bild 1. Dies sollten Sie sich am besten mehrere Male fotokopieren. Für die ersten Entwürfe ist aber normales kariertes Papier empfehlenswert, weil die meistens in endlose Radiererei ausarten. Die vielen Zahlen auf dem Entwurfsblatt sollten Sie jetzt nicht verwirren: Zu deren Bedeutung kommen wir später.

Erst einmal wollen wir uns auf einfarbige Sprites beschränken. Sie können dann jedes der 504 Kästchen auf ihrem Sprite-Entwurfsblatt mit dem Stift einfärben oder auch freilassen. Jedes ausgemalte Kästchen wird später auf dem Bildschirm in der Spritefarbe zu sehen sein. Die leeren Kästchen sind später durchsichtig. Sie können also beispielsweise einen Buchstaben umrahmen, indem Sie einen Rahmen als Sprite zeichnen und dieses über den Buchstaben legen. In der Spritemitte, in der sich nur leere Kästchen befinden, ist der Buchstabe zu sehen.

So, jetzt kommt gleich zum ersten Mal der POKE-Befehl ins Spiel. Denn die nächste Frage lautet natürlich: Wie erkläre ich meinem C 64, wie das Sprite auszusehen hat? Für den Nur-Basic-Anwender, der auch umfangreiche Programme schreiben will, gibt es da nicht viele Möglichkeiten. Dem Computer muß das Aussehen des Sprites in 63 Speicherstellen mitgeteilt werden, die an bestimmten Stellen im Arbeitsspeicher stehen müssen. Wieso gerade 63? Wir erinnern uns: Ein Punkt kann entweder eingeschaltet sein oder ausgeschaltet (durchsichtig).

In einer Speicherstelle lassen sich acht Ein-/Aus-Informationen speichern, es finden also acht Punkte eines Sprites Platz. Ein Sprite ist nun 24 Punkte breit. 24 ist 3 mal 8, also benötigen wir drei Speicherstellen, um eine Zeile eines Sprites zu definieren. Schließlich hat ein Sprite noch 21 Zeilen, so daß wir 3 mal 21, das sind 63 Speicherstellen, für die Punkte eines Sprites benötigen.

Wo liegen nun diese 63 Speicherstellen? Hier haben wir, wollen wir nicht mit Unmengen von Tricks und noch mehr POKEs arbeiten, nur vier Möglichkeiten. Diese sind in Tabelle 1 aufgeführt.

Speicherbereich	Codezahl
704 — 766	11
832 — 894	13
896 — 958	14
960 — 1022	15

Tabelle 1: Speicherbereiche, in denen Spritedaten abgelegt werden können.

Und wie bekomme ich nun mein Sprite da rein, werden Sie jetzt vielleicht fragen? Nun, schauen wir uns nochmal das Entwurfsblatt an. Jetzt sollen die Zahlen entschlüsselt werden: Wie gesagt, es wird eine Zeile jeweils durch drei Zahlen verschlüsselt. Die drei dicken Striche geben an, welcher Zeilenbereich durch eine Zahl verschlüsselt wird.

Angenommen wir möchten in der ersten Reihe die Punkte 1, 3 und 5 setzen. Diese liegen alle in der ersten Spaltengruppe. Um den Zahlenwert zu errechnen, müssen Sie nun einfach die Zahlen addieren, die über den gesetzten Punkten stehen. In unserem Beispiel wären das die 128, die 32 und die 8. Das Ergebnis, 158, tragen Sie dann in das erste Kästchen mit der Überschrift »Zahlencodes« ein. Genauso verfahren Sie dann mit der zweiten und dritten Spaltengruppe und den weiteren 20 Zeilen. Schon haben Sie die 63 Zahlenwerte, die das Sprite codieren, ausgerechnet. Übrigens: Daß jede zweite Spaltentrennung gestrichelt gezeichnet ist, hat noch praktische Gründe, wie wir bei mehrfarbigen Sprites sehen werden.

Ein Sprite erscheint

Jetzt besteht also unser Sprite aus 63 Zahlen. Diese können wir nun in die angeführten Speicherbereiche POKEN (Tabelle 1). In einem Programm speichert man die 63 Zahlenwerte praktischerweise in DATAs und verwendet dann eine FOR-NEXT-Schleife, in der die Zahlenwerte in den Speicherbereich gePOKEt werden. Jedem Speicherbereich ist dabei eine Codezahl zugeordnet, die nachher noch von entscheidender Bedeutung sein wird. Die letzten drei Bereiche werden allerdings jedesmal gelöscht, wenn Sie die Datasette benutzen.

Das war aber noch längst nicht alles. Es ist noch einiges zu tun, bevor das Sprite auf dem Schirm zu sehen ist. Wir

beschränken uns, bescheiden wie wir sind, auf ein einziges Sprite, das die Nummer 0 erhält.

Als erstes sollten wir dem Video-Chip mitteilen, wo wir das Sprite im Speicher plaziert haben. Dazu dient die Speicherstelle 2040. In diese Speicherstelle POKEn Sie bitte die Codezahl des Speicherbereiches.

Dann müssen Sie das Sprite noch wie eine Lampe einschalten. Dazu POKEn Sie in die Speicherstelle 53269 einfach eine 1. Nun dürfte ihr Sprite irgendwo auf dem Schirm zu sehen sein, wahrscheinlich aber nicht da, wo Sie es gern hätten. Dem kann Abhilfe geschaffen werden: In die Speicherstelle 53248 POKEn Sie die X-Koordinate, in 53249 die Y-Koordinate der gewünschten Position. In Bild 2 finden Sie dazu eine Orientierungshilfe. Sie werden sehen, daß das Koordinatensystem für die Sprites größer ist als der Bildschirm. Ein Sprite kann also hinter dem Bildschirmrand verschwinden und ist dann nicht mehr zu sehen, obwohl es eingeschaltet ist. Ein weiteres Problem ergibt sich da: Die X-Koordinate kann laut Bild 2 bis 511 gehen. Will man aber Werte größer 255 in die Speicherstelle der X-Koordinate POKEn, erhält man einen ILLEGAL QUANTITY ERROR. Um das Sprite auch in die rechte Hälfte bewegen zu können benötigt man eine weitere Speicherstelle, 53264. Soll das Sprite auf eine Position größer als 255, dann POKEt man dort eine 1 hinein. Danach zieht man 256 von der X-Koordinate ab und POKEt das Ergebnis wieder in die Speicherstelle 53248.

Somit können wir also ein Sprite auf jede Stelle am Bildschirm und sogar über den Bildschirm hinaus positionieren.

Vielleicht gefällt ihnen die Farbe Ihres Sprites aber überhaupt nicht, vielleicht konnten Sie es bisher auch gar nicht sehen, weil es dieselbe Farbe wie der Hintergrund hatte. Nun, auch hier hilft ein POKe: Die Speicherstelle 53287 bestimmt die Farbe unseres Sprites. Dort können die 16 im Commodore-Handbuch angegebenen Farbcodes hineingeschrieben werden.

Ein Weiteres läßt sich bei Sprites kontrollieren: Die Priorität. Nehmen wir einmal an, ein Sprite und ein Buchstabe oder Grafikzeichen überschneiden sich. Dann gibt es zwei Möglichkeiten: Das Sprite ist »hinter« dem Buchstaben, wird also verdeckt, oder ist »davor«, verdeckt also den Buchstaben. Sie können einstellen, wie sich das Sprite verhalten soll: Steht in Speicherstelle 53275 eine 1, so ist das Sprite »hinten«, ist dort eine 0, ist das Sprite »vorne«.

Klein, aber fein sind diese Sprites, werden Sie jetzt wahrscheinlich sagen. Warten Sie's ab, es geht auch größer. Ein Sprite läßt sich »strecken« in X- und Y-Richtung. POKEn Sie

mal eine 1 in 53277, und Sie verstehen, was damit gemeint ist. In die andere Richtung geht's mit der Speicherstelle 53271. POKEt man in beide Speicherstellen eine 0, ist der Spuk vorbei, und das Sprite hat seine »normale« Größe wieder erreicht.

Ein kleines Problem, das sogar Profis ins Grübeln gebracht hat: Streckt man ein Sprite in X-Richtung und setzt es an die X-Position Null, verschwindet es nicht vom Schirm! Die rechte Hälfte ist immer noch zu sehen. Soll nun aber das Sprite einen eleganten Abgang nach links machen, ist dies natürlich absolut störend. Doch auch hier gibt es eine bestechend einfache Lösung. Stellen Sie sich das Sprite-Koordinatensystem als Zylinder vor. Wenn Sie unser gestrecktes Sprite an die Position 511 setzen, sehen Sie wiederum fast die Hälfte des Sprites am linken Bildschirmrand. Verringert man die X-Koordinate nun langsam auf 488, wird es nach links hin entschwinden, wie wir es uns wünschen. In Y-Richtung taucht dieses Problem nicht auf.

Jetzt wird's bunt

Bisher haben wir uns auf einfarbige Sprites beschränkt. Es geht aber auch dreifarbig! Eigentlich sind es sogar vier Farben, aber die vierte Farbe ist »durchsichtig«, und wir zählen sie deswegen nicht mit.

Bitkombination	Farbe
00	Durchsichtig
11	Farbe 1
01	Farbe 2
10	Farbe 3

Tabelle 2: Zusammenhang zwischen Ziffernkombinationen und Farbe bei Multicolor-Sprites.

Bevor Sie jetzt aber jubelnd die buntesten Sprites entwerfen, muß ich auf den Nachteil des »Multicolor«-Modus hinweisen: Sie haben nun nicht mehr 24, sondern nur noch 12 Punkte pro Zeile, diese sind allerdings auch doppelt so breit wie die eines tristen, einfarbigen Sprites.

Konkret bedeutet das, daß Sie schon beim Entwurf immer zwei nebeneinanderliegende Kästchen einfärben müssen, wenn Sie dort einen Punkt setzen sollen. Dabei dürfen Sie aber drei Farben benutzen. Die gestrichelten Linien in unserem Entwurfsblatt fallen also weg.

Spaltennummer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	Zählencodes
Werte	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	128	64	32	16	8	4	2	1	
Zeile 1																									
Zeile 2																									
Zeile 3																									
Zeile 4																									
Zeile 5																									
Zeile 6																									
Zeile 7																									
Zeile 8																									
Zeile 9																									
Zeile 10																									
Zeile 11																									
Zeile 12																									
Zeile 13																									
Zeile 14																									
Zeile 15																									
Zeile 16																									
Zeile 17																									
Zeile 18																									
Zeile 19																									
Zeile 20																									
Zeile 21																									

Bild 1. Ein Sprite-Entwurfsbogen

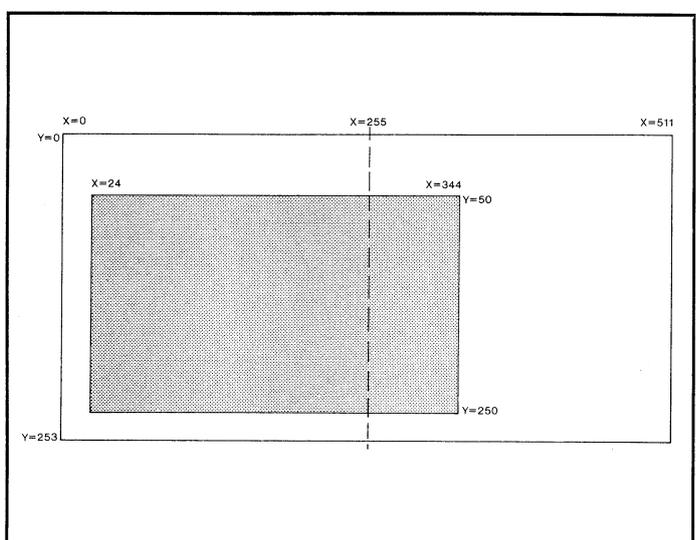


Bild 2. Mögliche Sprite-Positionen mit Bildschirmbereich

	Sprite							
	0	1	2	3	4	5	6	7
Codezahl-Speicher	2040	2041	2042	2043	2044	2045	2046	2047
Spritefarbe (1)	53287	53288	53289	53290	53291	53292	53293	53294
X-Position	53248	53250	53252	53254	53256	53258	53260	53262
Y-Position	53249	53251	53253	53255	53257	53259	53261	53263

Tabelle 3: Speicherstellen, die einzelne Sprites betreffen.

Beim Errechnen der 63 Zahlen für das Sprite geht man dann so vor: Anstelle einer 1 für jeden gesetzten Punkt müssen Sie nun zwei Ziffern in das Sprite-Entwurfsblatt einsetzen, eine in das linke, eine in das rechte Kästchen eines Punktes. Welche Ziffern das sind, ist von der Farbe abhängig und steht in Tabelle 2. Ab dann verläuft das Ausrechnen der Zahlenwerte des Sprites genauso wie bei den einfarbigen Sprites. Allerdings kommen noch drei zusätzliche POKES hinzu: In die Speicherstelle 53276 muß eine 1, damit der Video-Chip weiß, daß es sich jetzt um ein Multicolor-Sprite

Einschalten	53269
Übertrag X-Position	53264
Vergrößern X	53277
Vergrößern Y	53271
Multicolor einsch.	53276
Priorität-Hintergrund	53275

Tabelle 4: Speicherstellen, die jeweils acht Schalter (je Sprite einen) enthalten.

handelt. In die Speicherstelle 53285 kommt dann die Farbe 2, in die Speicherstelle 53286 die Farbe 3. Die Farbe 1 haben wir vorher schon in der Speicherstelle 53287 festgelegt. Schon schimmert uns ein schönes mehrfarbiges Sprite entgegen, das wir dann genauso wie ein einfarbiges behandeln können.

Ein Sprite kommt selten allein

Schließlich steht ja im Commodore-Handbuch was von acht verschiedenen Sprites gleichzeitig. Also ohne viel Tricks kriegen wir nur vier verschiedene Sprites auf den Bildschirm, denn wir können ja nur vier Sprites in die Speicherstellen aus Tabelle 1 POKEN. Dafür haben wir die Möglichkeit, ein Sprite doppelt, dreifach oder sogar achtfach gleichzeitig auf den Bildschirm zu bringen. Mit acht Sprites ist dann allerdings auch die Kapazität unseres Video-Chips erschöpft.

Wie kann man nun über die sieben zusätzlichen Sprites verfügen? Diese Sprites haben praktischerweise die Nummern 1 bis 7. Für ihre Position, Farbe und ihre Codezahl haben sie ihre eigenen Speicherstellen, die in Tabelle 3 aufgeführt sind.

Komplizierter wird es da schon bei den anderen Speicherstellen, die in Tabelle 4 aufgelistet wurden. Jede dieser Speicherstellen hat Kontrolle über alle acht Sprites. Wenn Sie genauer hinschauen, bemerken Sie, daß alle diese Speicherstellen nur An-/Aus-Funktion haben, sie sind sozusagen Schalter. Nun können aber in einer Speicherstelle, wie wir wissen, acht An-/Aus-Informationen gespeichert werden, also für jedes Sprite genau eine. Jetzt folgen zwei Routinen, deren Wirkung ich Ihnen gerne erkläre, nicht aber die Funktionsweise. Dazu müßte ich längere Vorträge über Binärarithmetik, logische Funktionen und ähnliches halten und damit den Rahmen dieses Artikels sprengen.

POKE X, PEEK(X) OR 2 ↑ N

legt den Schalter in der Speicherstelle X für das Sprite N auf »An« um, ohne die anderen Schalter zu beeinflussen.

POKE X, PEEK(X) AND (255 - 2 ↑ N)

stellt den entsprechenden Schalter wieder auf »Aus« um, ohne die anderen zu beeinflussen.

Wichtig sind die beiden Adressen in der Tabelle 5: Sie sind uns schon bekannt, und legen die Farben 2 und 3 für **alle** Sprites verbindlich fest. Zwei Sprites im Multicolormodus, die gleichzeitig erscheinen, können sich also nur in der Farbe 1 unterscheiden.

Farbe 2	53285
Farbe 3	53286

Tabelle 5: Speicherstellen, die auf alle acht Sprites gleichzeitig wirken

Damit ist das Wichtigste über Sprites gesagt. In Basic sollte man sich auf stehende Sprites beschränken. Sich bewegende Sprites sind relativ langsam, wenn das Basic-Programm auch andere Funktionen nebenbei erfüllen muß.

Wer Sprites öfters benutzen will, für den lohnt es sich, die ebenfalls in diesem Heft abgedruckte Basic-Erweiterung »Sprite-Basic« abzutippen, und das soeben Gelernte dann nicht mit POKES, sondern mit richtigen Befehlen auszuführen. Denn dann geht der Sprite-Spaß erst richtig los!

(Boris Schneider/ev)

