Comal — Eine

In dieser Folge unserer Comal-Einführung befassen wir uns mit einem der interessantesten Aspekte dieser Programmiersprache, nämlich der Grafik.

Die Comal-Grafik kann ihre Herkunft von einer anderen Programmiersprache nicht verheimlichen. Befehle wie »SHOWTURTLE« und »TURTLESIZE« sind verräterisch: Die Sprache Logo mit ihrer bekannten Turtle-Grafik stand neben Basic und Pascal Pate beim Entwurf von Comal.

Die Bezeichnung Turtle-Grafik stammt noch aus den Anfängen von Logo. Turtle ist das englische Wort für »Schildkröte« und bezeichnete ursprünglich eine mechanische Schildkröte, die sich per Computersteuerung über ein großes Blatt Papier bewegte und dabei mit einem Schreibstift eine sichtbare Spur hinterließ. Zur Steuerung dieser »Schildkröte« wurde eine spezielle Programmiersprache entwickelt, mit Befehlen wie »FORWARD«, »BACK«, »LEFT« etc. Das funktionierte einige Zeit ganz gut, aber dann tauchten die ersten Mikrocomputer mit Grafikbildschirmen auf, und die Schildkröte hauchte ihr mechanisches Leben aus und degenerierte zu einem kleinen Dreieck auf einem Grafikbildschirm. Doch die grafische Programmiersprache Logo war geboren und ist seitdem immer weiterentwickelt worden.

Das Konzept der Turtle-Grafik wurde vollständig in Comal integriert (Tabelle 1). Daneben gibt es auch spezielle Kommandos zum Zeichnen einzelner Punkte oder Linien, und zwar unabhängig von der jeweiligen Turtleposition. Überhaupt wird der gesamte Bereich von Grafik, Farbe und Bildschirmansteuerung beim C 64 durch Comal-Befehle abgedeckt (Tabelle 2), man muß sich nicht wie im V.2-Basic mit merkwürdigen PEEKs und POKEs herumschlagen.

Comal kennt drei verschiedene Bildschirmmodi, nämlich den normalen Textmodus (»SETTEXT«), den hochauflösenden Grafikmodus (»SETGRAPHIC 0«) und den Mehrfarben-Grafikmodus (»SETGRAPHIC 1«). Der Textmodus wird auch mit der Funktionstaste f1 eingeschaltet, während f5 die Grafik einschaltet. Im hochauflösenden Modus kann mit f3 zusätzlich noch ein Textfenster eingeblendet werden, was beim Experimentieren mit der Turtle-Grafik recht nützlich ist. Die oberen beiden Bildschirmzeilen bilden dann das Textfenster, mit dessen Hilfe man beispielsweise im Direktmodus mit entsprechenden Befehlen die Turtle steuern kann.

Geben Sie einmal im Direktmodus den Befehl »CLEAR« ein (um den Grafikbildschirm zu löschen) und schalten Sie danach durch Drücken der f3-Taste in die hochauflösende Grafik mit Textfenster um. Sie sehen in der Mitte des ansonsten leeren Bildschirms ein kleines weißes Dreieck, die Turtle. In der linken oberen Bildschirmecke blinkt der Cursor und zeigt damit an, daß die Turtle auf Befehle von Ihnen wartet.

Anweisung	Bedeutung
FORWARD n	Turtle n Schritte vorwärts
BACK n	Turtle n Schritte rückwärts
HOME	Turtle auf Grundposition zurücksetzen
LEFT w	Turtle um w Grad links drehen
RIGHT w	Turtle um w Grad rechts drehen
SETXY x,y	Turtle auf die Koordinaten x,y setzen
SETHEADING w	Turtle-Richtung absolut festlegen (w=0 bedeutet, daß Turtle nach oben zeigt)
SHOWTURTLE	Turtle sichtbar machen
HIDETURTLE	Turtle unsichtbar machen
PENUP	Turtleweg wird nicht mitgezeichnet
PENDOWN	Turtleweg wird mitgezeichnet
PENCOLOR n	Zeichenfarbe n wählen
TURTLESIZE n	Turtlegröße auf n Punkte festlegen

Tabelle 1. Befehle für Turtle-Grafik

Anweisung	Bedeutung
BACKGROUND n	Hintergrundfarbe n wählen
BORDER n	Rahmenfarbe n wählen
CLEAR	Grafikbildschirm löschen
DRAWTO x,y	Zeichnet eine Linie zum Punkt x,y
FILL	Flächen mit Farbe ausfüllen
FRAME x1,x2,y1,y2	Window festlegen
FULLSCREEN	Ganzer Bildschirm für Grafik
MOVETO x,y	Grafik-Cursor auf x,y setzen
PLOT x,y	Grafikpunkt setzen
PLOTTEXT x,y,text$	Text in Grafikbildschirm einblenden
SETGRAPHIC	Grafik-Modus wählen
SETTEXT	Auf Textbildschirm umschalten
SPLITSCREEN	In Grafikbildschirm zwei Textzeilen reservieren

Tabelle 2. Allgemeine Bildschirm- und Grafikbefehle

So wird die »Schildkröte« bewegt

Geben Sie jetzt »FORWARD 50« ein. Die Turtle bewegt sich damit um 50 Einheiten vorwärts und zieht dabei eine Linie längs ihres Weges. Mit »LEFT 90« erreichen Sie eine Drehung der Turtle um 90 Grad nach links. Wenn Sie jetzt nochmals »FORWARD 50« eingeben, bewegt sich die Turtle in die neue Rich-

Einführung (3)

Anweisung	Bedeutung
DATACOLLISION n,b	Sprite-Hintergrund-Kollisionsabfrage für Sprite n (die Variable b wird 1, falls Kollision, sonst b=0)
SPRITECOLLISION n,b	Sprite-Sprite-Kollisionsabfrage für Sprite n (b wird 1, falls Kollision, sonst b=0)
DEFINE d,x$	Der Inhalt der Stringvariablen x$ definiert ein Sprite mit Definitionsnummer d
IDENTIFY n,d	Ordnet dem mit Nummer d definierten Sprite die Spritenummer n zu
PRIORITY n,p	Hintergrund hat Priorität vor Sprite n (p=0), umgekehrt für p=1
SPRITEBACK a,b	Setzt a und b als zusätzliche Farben für Multicolor-Sprites
HIDESPRITE n	Sprite n nicht anzeigen
SHOWSPRITE n	Sprite n anzeigen
SPRITECOLOR n,c	Farbe c für Sprite n wählen
SPRITEPOSITION n,x,y	Sprite n auf Position x,y setzen
SPRITESIZE n,x,y	Sprite n in x- oder y-Richtung vergrößern

Tabelle 3. Sprite-Befehle

tung. Soll eine Bildschirmstelle angefahren werden, ohne eine Linie dorthin zu ziehen, dann wird mit dem Kommando »PENUP« einfach der symbolische Schreibstift von der Zeichenfläche abgehoben. Die Turtle kann dann nach Belieben über den gesamten Bildschirm dirigiert werden, ohne Spuren zu hinterlassen. Aber keine Angst, nach »PENDOWN« zeichnet sie wieder.

Wenn Sie es gerne etwas bunter hätten, bitte sehr. »PENCOLOR« wählt die Schreibfarbe der Turtle, »BORDER« und »BACKGROUND« wählen Rahmen- und Hintergrundfarbe. Erscheint Ihnen die Turtle zu gut genährt, dann können Sie zum Beispiel mit »TURTLESIZE 5« bedenkenlos etwas Speck entfernen. Und sollte Ihnen die Turtle insgesamt nicht ganz geheuer sein, dann tippen Sie einfach »HIDETURTLE« in Ihren Computer ein. Die »Schildkröte« wird sich daraufhin schmollend von der Bildfläche zurückziehen, aber zum Glück wird ihr Wirken dadurch nicht behindert: Aus dem Unsichtbaren befolgt sie weiter Ihre Befehle und zieht fleißig ihre Linien.

Kleine Programme mit großer Wirkung

Natürlich ist die Grafikerzeugung mit dieser sehr direkten Methode auf die Dauer entschieden zu langwierig. Wenn Sie das kleine Programm aus Listing 1 einmal ausprobieren, werden Sie feststellen, daß effektvolle Grafikprogramme in Comal weder besonders komplex noch langsam zu sein brauchen. Experimentieren Sie ruhig mit ähnlichen Programmen; Sie werden sehr schnell ein gewisses Gespür dafür entwickeln, wie Sie durch Wiederholung einfacher Grundfiguren mit jeweils einem etwas geänderten Parameter interessante grafische Effekte erzielen können.

Zugabe: Sprites in Comal programmiert

Der große Vorteil der Turtle-Grafik gegenüber dem Zeichnen von Linien nach festen Koordinaten ist ja gerade die erstaunliche Einfachheit, mit der sich recht komplexe Strukturen vom Standpunkt der »Schildkröte« aus darstellen. Mit Comal können Sie wirklich eine »Reise durch die Wunderwelt der Grafik« antreten — auch ohne sich mit Bits und Bytes abzuplagen und mit komplizierten mathematische Formeln zu jonglieren.

Die Turtle-Grafik ist nur die eine Seite der grafischen Fähigkeiten des C 64 - Comal. Es stehen nämlich zusätzlich eine Reihe von leistungsfähigen Anweisungen zur Erzeugung und Kontrolle von Sprites zur Verfügung.

Auch bei Sprites wird zwischen Hochauflösung (Hires) und Mehrfarbendarstellung (Multicolor) unterschieden. Ein Hires-Sprite besteht bekanntlich aus 24 x 21 Punkten oder aus 21 Reihen zu je drei Byte. Bei Multicolor-Sprites haben wir nur 12 x 21 Punkte, dafür benötigt jeder Bildpunkt aber intern zwei Bit, da vier verschiedene Farben pro Punkt möglich sind. In beiden Fällen kann die in einem Sprite enthaltene Information in 63 Bytes dargestellt werden. Comal verwendet ein zusätzliches Byte, das angibt, ob es sich um ein Hires-Sprite (Byte 64 = 0) oder um ein Multicolor-Sprite (Byte 64 <> 0) handelt. Beide Spritearten werden im übrigen

```
0001 //
0002 // Grafik-Demo
0003 //
0010 print chr$(147)
0020 pencolor 1
0030 input " Abstand:    1■■": abstand
0040 input " Winkel :   89■■■": winkel
0050 input " Inkrement: 1■■": i
0060 setgraphic 0
0070 while abstand<365 do
0080  forward abstand
0090  left winkel
0100  abstand:=abstand+i
0110 endwhile
```

Listing 1. Eine kleine Demonstration der Turtle Grafik

Comal — Eine Einführung (3)

```
0010 // ----------------------
0020 // Turtle- und Sprite-Demo
0030 // ----------------------
0040 //
0050 //
0060 settext
0070 input "Rahmenfarbe: ": rahmen
0080 border rahmen
0090 input "Hintergrundfarbe: ": grund
0100 background grund
0110 setgraphic 0
0120 dim sprite$ of 63
0130 dim ein$ of 4
0140 //
0150 // Turtle initialisieren
0160 //
0170 penup
0180 setxy 160,160
0190 pendown
0200 turtlesize 10
0210 showturtle
0220 //
0230 // Einfaches Sprite-Bild erzeugen
0240 //
0250 sprite$:=""
0260 ein$:=chr$(255)+chr$(0)+chr$(255)
0270 y:=50
0280 for i:=1 to 63/3 do
0290  sprite$:=sprite$+ein$
0300 endfor i
0310 //
0320 // Zwei Sprites definieren
0330 //
0340 define 0,sprite$+chr$(0)
0350 sprite$(62):=chr$(255)
0360 sprite$(2):=chr$(255)
0370 define 1,sprite$+chr$(0)
0380 //
0390 // Vier Sprites aus zwei Bildern
0400 //
0410 identify 0,0
0420 identify 1,0
0430 identify 2,1
0440 identify 3,1
0450 //
0460 // Spritefarbe setzen
0470 //
0480 spritecolor 0,0
0490 spritecolor 1,2
0500 spritecolor 2,7
0510 spritecolor 3,1
0520 //
0530 // Demo laeuft bis Taste gedrueckt wird
0540 //
0550 repeat
0560  pencolor rnd(0,15)
0570  bewegung
0580  penup
0590  right 45
0600  forward 80
0610  pendown
0620 until key$<>chr$(0) // Taste gedrueckt ?
0630 //
0640 // Sprites und Turtle bewegen
0650 //
0660 proc bewegung
0670  //
0680  // Spritegroesse zufaellig aendern
0690  //
0700  spritesize 0,rnd(0,1),rnd(0,1)
0710  spritesize 1,rnd(0,1),rnd(0,1)
0720  spritesize 2,rnd(0,1),rnd(0,1)
0730  spritesize 3,rnd(0,1),rnd(0,1)
0740  for x:=1 to 300 step 5 do
0750   //
0760   // Turtle bewegen
0770   //
0780   left 88
0790   forward 50
0800   //
0810   // Sprites bewegen
0820   //
0830   spritepos 0,x,y
0840   spritepos 1,300-x,y+50+x/3
0850   spritepos 2,x,y+50+x/3
0860   spritepos 3,300-x,y
0870  endfor x
0880 endproc bewegung
```

Listing 2. Turtle-Grafik und Sprites: So einfach und übersichtlich ist die Programmierung

mit völlig gleichartigen Befehlen angesprochen.

Zur Spriteerzeugung dient der Befehl »DEFINE«. Als Parameter wird eine Definitions-Nummer sowie ein String angegeben. Die Definitions-Nummer kann irgendeine Zahl von 0 bis 95 sein, denn Comal kann bis zu 96 verschiedene Spritemuster verwalten. Der anzugebende String muß eine Länge von 64 Zeichen haben und das Punktmuster des Sprites enthalten. Das Beispielprogramm in Listing 2 zeigt, wie man in Comal mit Sprites umgeht.

Wichtig ist es, genau zwischen der Definitions-Nummer eines Sprites und der eigentlichen Spritenummer zu unterscheiden. Es können hardwarebedingt ja immer nur acht Sprites gleichzeitig auf dem Bildschirm sein. Wird die Turtle benutzt, die selber ein Sprite ist, dann verringert sich diese Anzahl auf sieben.

Mit dem »IDENTIFY«-Kommando wird die Beziehung zwischen Sprite-Nummer und Definitions-Nummer hergestellt. »IDENTIFY 1,30« bewirkt beispielsweise, daß das Sprite mit der Definitions-Nummer 30 als Sprite 1 auf dem Bildschirm dargestellt wird. So kann sehr schnell zwischen verschiedenen Sprites umgeschaltet werden, was zur Erzeugung von zeichentrickähnlichen Effekten benutzt werden kann.

Wie aus Tabelle 3 zu ersehen ist, gibt es eine Vielzahl von weiteren Comal-Befehlen speziell zur Sprite-Kontrolle. Die Wirkungsweise solcher Anweisungen wie »SPRITECOLOR« oder »SPRITEPOS« dürfte wohl jedem verständlich sein, der sich schon einmal von Basic aus an die Spriteprogrammierung gewagt hat.

Comal oder Basic?

Wie die drei Folgen unserer kleinen Comal-Einführung hoffentlich gezeigt haben, hat Comal dem Commodore V.2-Basic einiges voraus. Die von Pascal entlehnte Programmstrukturierung mittels Prozeduren und die aus Logo stammende Turtle-Grafik haben sich mit dem aus Basic übernommenen Konzept einer benutzerfreundlichen, interaktiven Programmiersprache zu einer gelungenen Synthese zusammengefügt. Die hier besprochene Comal-Version 0.14 ist dabei nur als eine Art Vorabversion zu werten. Eine nochmals wesentlich erweiterte und verbesserte Version 2.0 wird von Commodore Dänemark bereits als Steckmodul (64 KByte ROM, 30 KByte RAM frei) vertrieben. Es ist zu hoffen, daß auch Commodore Deutschland diese zukunftsweisende Programmiersprache bald anbieten wird.(ev)

Info: Comal-Literatur:
Borge Christensen, Heiko Wolgast: Comal 0.14-Handbuch, 80 Seiten, Verlag Schmidt & Klaunig, Ringstraße, 2300 Kiel.
Borge Christensen: Strukturierte Programmierung mit Comal, 230 Seiten, Verlag R. Oldenbourg, München, ISBN 3-486-26901-1