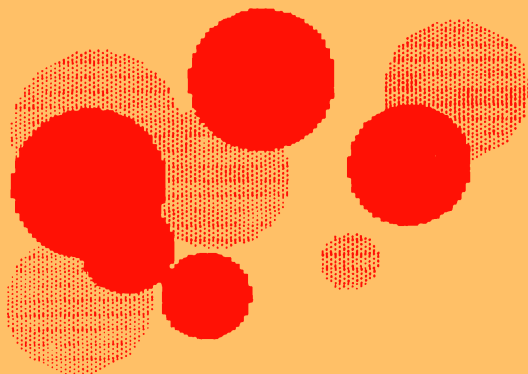


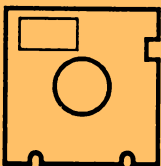
COMAL - KINDERLEICHT

von Christian & Peter Bason



COMAL Graphikprogramme für
deinen Commodore 64 oder 128

Die in diesem Buch abgedruckten Programme sind auch auf Diskette erhältlich. Sie können sie bestellen bei:



D. SPERLING
Hard- und Software
Westersteig 38
D-2260 Niebüll

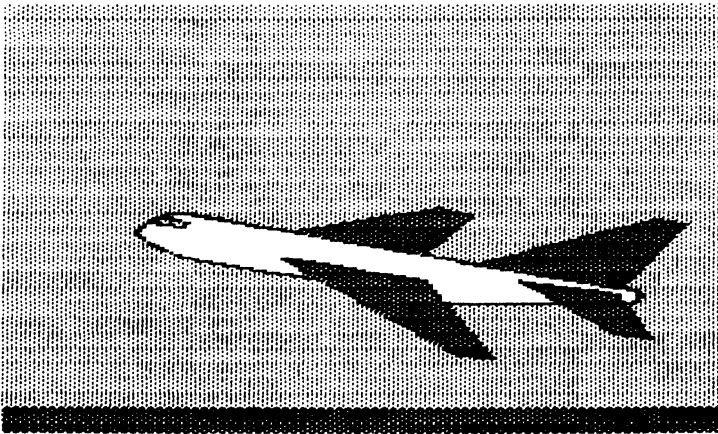
Der Preis beträgt DM 29,-- zuzüglich Porto.

COMAL - KINDERLEICHT

von Christian und Peter Bason

**ein Buch mit Beispielen zur Graphik
von Jugendlichen für Jugendliche
(oder andere junggebliebene Anfänger)**

ins Deutsche Übertragen von Karl-Uwe Sperling



**COMAL Graphikprogramme für
deinen Commodore 64 oder 128**

Copyright (C) 1986

D. SPERLING
Hard- und Software
Westersteig 38
D-2260 Niebüll

und

SILKEBORG SOLDATA
Christian & Peter Bason
Linåbakken 13
DK-8600 Silkeborg

Alle Rechte vorbehalten.

Kein Teil dieses Buches darf in irgendeiner Form (Druck, Fotokopie oder einem anderen Verfahren) ohne schriftliche Genehmigung der Firma D. Sperling reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

Printed in Denmark, Silkeborg Bogtryk A/S

ISBN: 87-982231-2-7

Vorwort

Dieses Buch ist vorwiegend für Kinder und Jugendliche geschrieben, aber auch für Erwachsene, die eine leichte Einführung in COMAL wünschen. **COMAL - kinderleicht** ist ein Beispiel für die Freude und das Engagement, das Kinder beim Spiel und bei der Arbeit mit dem Computer entwickeln.

Christian und Peter begannen mit der Anwendung der Computersprache LOGO im Jahre 1983. Bei den meisten ihrer LOGO-Projekte handelte es sich um Aktivitäten, die die meisten Kinder lieben, nämlich das Zeichnen. Sie zeichneten Figuren, Tiere und Gesichter, sie zeichneten mit Strichen und füllten Flächen mit Farben. So kann man Zeichnungen mit Hilfe eines Computers erstellen, ohne selbst Meister im Freihandzeichnen zu sein. Mit Hilfe des Computers können auch größere Zeichenprojekte bewältigt werden, wenn man nur die Regeln der Sprache beherrscht. Und dieses ist ein Gebiet, auf dem Kinder unübertroffen sind. Wenn sie motiviert sind, erlernen Kinder die Sprache auf natürliche Weise wesentlich schneller, als die meisten Erwachsenen es könnten. Commodore-COMAL haben meine Kinder gelernt, weil die Sprache ihnen half, schneller zu gelungenen Zeichnungen zu kommen.

Kinder bekommen oft sehr schnell einen unmittelbaren und natürlichen Kontakt zu Computern, nahezu so, als wären es lebende Wesen. Zudem ist der Computer ein Gerät aus der Welt der Erwachsenen, das Kinder ebenfalls benutzen können. Der Computer kann auf diese Weise zu einem kommunikationsfördernden Medium sowohl zwischen Kindern als auch zwischen Kindern und Erwachsenen werden. In meinem Buch, **Børn, unge og datamater**, (Teknisk Forlag, 1985) - 'Kinder, Jugendliche und Computer' - kann man viele weitere Beispiele zu Programmen und Sprache finden, an denen Kinder und Jugendliche Freude haben werden.

Dieses Buch von Christian und Peter soll anderen Kindern als Anregung dienen, ihnen aber auch helfen, die Sprache zu erlernen.

Die Kinder haben eine Reihe von Programmen aufgestellt in der Hoffnung, damit anderen Kindern einen Einstieg in die Sprache zu ermöglichen. Sie werden dazu aufgefordert, Teile zu ändern und so mit den Programmen zu experimentieren.

Wenn Kinder das Buch durchgehend bearbeiten, erlangen sie ein umfangreiches Wissen über die Sprache, den Umgang mit Zahlen, über Geometrie, und sie erhalten auf diese Weise grundlegende Programmierkenntnisse. Dieses wird eben dadurch möglich, daß die Kinder nicht nur "spielen", sondern ihren Computer auch tatsächlich programmieren. So wendet sich dieses Buch an Kinder wie Christian und Peter, die, wie ich glaube, ebenfalls Freude am Programmieren finden können. Vielleicht ist das Wichtigste, daß wir durch eigenes Programmieren die Fähigkeit erlangen, präzise zu planen und so zu Problemlösungen zu kommen. Nicht zuletzt gewinnt man dadurch Selbstachtung und Selbstvertrauen, wie wir es alle, sowohl die Kleinen als auch die Großen, gebrauchen können.

Viel Vergnügen beim Programmieren!

Frank Bason
Silkeborg Amtsgymnasium
August 1985

Inhaltsverzeichnis

0. Einleitung	9
1. Einfache Figuren	11
2. Es wird ein bißchen schwieriger	17
3. Fahrräder, Katzen und Elefanten	25
4. COMAL und sprites	39
5. Und was nun?	45
Anhang: weitere Programmausdrucke	46
Weiterer Bücher über COMAL	62
Stichwortverzeichnis	63

0. Einleitung

In diesem Buch wollen wir an verschiedenen Beispielen zeigen, wie man mit COMAL-80 Zeichnungen erstellen kann. Man kann alles darstellen: Vierecke oder Fahrräder, einen Elefanten oder ein Flugzeug und vieles andere mehr.

Zuerst lernst du, ein Quadrat zu zeichnen, und dann benutzt du es, um damit eine andere interessante Figur entstehen zu lassen, und später wird mit derselben Methode eine Blume gezeichnet. Du siehst im Buch jeweils das gelistete Programm, einen kurzen Text, der erklärt, wie es arbeitet, und meistens noch das entstehende Bild.

Versuche, anschließend selbst Programme auf deinem Computer zu schreiben. Um dieses tun zu können, mußt du entweder einen Commodore 64 (oder SX-64) oder einen Commodore 128 mit dem COMAL-Modul-2.01 besitzen.

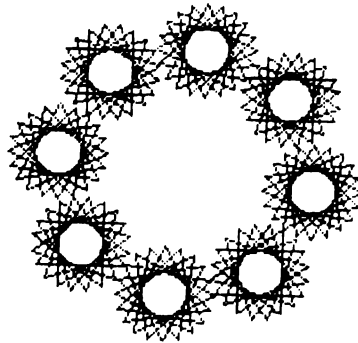
In diesem Buch erhältst du zwischendurch Aufgaben zu den Programmen, die du mit Hilfe deines Commodore versuchen solltest zu lösen. Im hinteren Teil des Buches befindet sich eine Programmsammlung mit einigen größeren Programmen (ein Garten, ein Flugzeug und andere Dinge), die du eintippen kannst, wenn du magst. Du kannst auch alle Programme dieses Buches (und noch einige zusätzliche) auf einer Diskette erhalten; benutze dazu die beiliegende Bestellkarte.

Wir finden, daß COMAL eine ausgezeichnete Sprache ist, um damit tolle Figuren und Gebilde zu zeichnen. COMAL 80 ähnelt unter anderem Logo, abgesehen von der Zeilennumerierung. COMAL ist die ideale Sprache, um damit zu zeichnen. Es wird dadurch so leicht, gute Zeichnungen zu machen - auf jeden Fall im Vergleich zu Basic.

Nachdem du die Programme in unserem Buch durchgegangen bist und sie verstanden hast, wirst du in der Lage sein, deine eigenen Programme zu machen.

Viel Vergnügen!

Christian und Peter Bason
Silkeborg, im August 1985



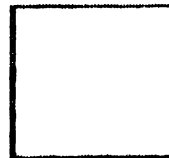
1. Einfache Figuren

In diesem Kapitel wird an Beispielen gezeigt, wie du einfache Figuren zeichnen kannst. Es beginnt mit einem Viereck und geht dann Schritt für Schritt weiter. Sieh dir die Programme der Reihe nach an, und mache vielleicht ein paar Übungen.

QUADRAT

Dieses Programm zeichnet ein Viereck (Quadrat). In den Zeilen 20 und 80 wird der Bildschirm zum Zeichnen vorbereitet. In Zeile 100 werden zwei Prozeduren aufgerufen. Die Prozedur 'quadrat' in den Zeilen 120-170 zeichnet ein Quadrat, das ist ein Viereck mit vier gleichlangen Seiten. Die Prozedur warte in den Zeilen 190-220 bewirkt, daß der Computer solange mit seiner weiteren Arbeit wartet, bis du eine Taste drückst. Sobald du das tust, ist dieses Programm beendet.

```
0010 // Programm "quadrat'zeichnen"
0020 USE turtle
0030
0040 background(2)
0050 border(1)
0060 pencolor(1)
0070
0080 graphicscreen(1)
0090
0100 quadrat; warte
0110
0120 PROC quadrat
0130   FOR i:=1 TO 4 DO
0140     forward(80)
0150     right(90)
0160   ENDFOR i
0170 ENDPROC quadrat
0180
```

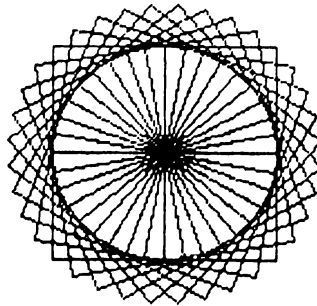


```
0190 PROC warte
0200 REPEAT
0210 UNTIL KEY$<>""
0220 ENDPROC warte
```

DREHUNG

Nun wird das Quadrat benutzt, um die abgebildete Figur zu zeichnen. Dazu wird zuerst das Quadrat gezeichnet, dann wird der Zeichenstift (turtle) ein wenig gedreht (Zeile 240), und das gleiche Quadrat wird erneut gezeichnet. Dieses wird 36 mal wiederholt.

```
0010 // Programm "drehung"
0020
0030 USE turtle
0040 USE graphics
0050
0060 graphicscreen(1)
0070
0080 pencolor(1)
0090 background(2)
0100 border(1)
0110
0120 drehung
0130
0140 PROC quadrat
0150 FOR i:=1 TO 4 DO
0160     forward(50)
0170     left(90)
0180 ENDFOR i
0190 ENDPROC quadrat
0200
0210 PROC drehung
0220 FOR i:=1 TO 36 DO
0230     quadrat
0240     left(10)
0250 ENDFOR i
0260 ENDPROC drehung
```



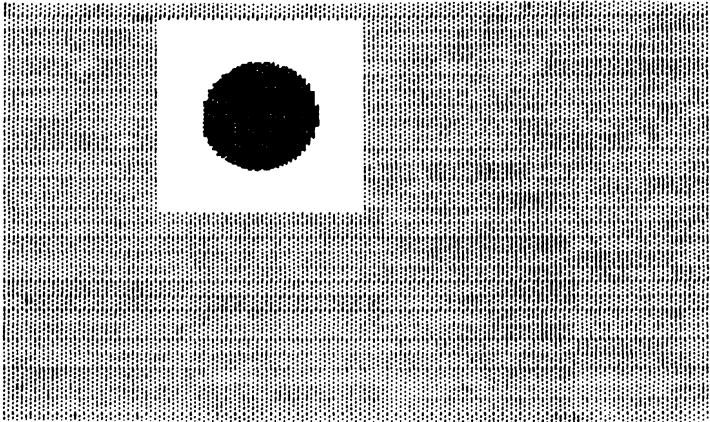
0270
0280 REPEAT
0290 UNTIL KEYS\$<>""

AUFGABE:

Versuche, eine regelmäßige Figur mit nur 5 anstatt 36 Quadraten zu zeichnen (Zeilen 220 und 240).

QUADRAT UND KREIS

Es wird wieder das Quadrat benutzt, diesmal jedoch farbig ausgemalt. Den eingezeichneten Kreis kannst du mit Hilfe des COMAL-Wortes 'circle' zeichnen, du mußt nur die Lage des Kreismittelpunktes und den Radius des Kreises angeben. Dann wird der Kreis noch mit Farbe ausgefüllt.



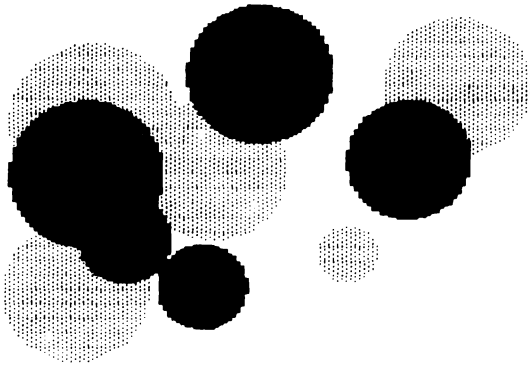
```
0010 // Programm "quadrat.kreis"
0020
0030 USE turtle
0040 USE graphics
0050
0060 graphicscreen(1)
0070
0080 pencolor(1)
0090 background(3)
0100 border(1)
0110
0120 quadrat'kreis
0130
0140 PROC quadrat'kreis
0150   FOR i:=1 TO 4 DO
0160     forward(90)
0170     left(90)
0180   ENDFOR i
0200   fill(-45,45)
0203   pencolor(2)
0205   circle(-45,45,25)
0207   fill(-45,45)
0210 ENDPROC quadrat'kreis
0290
0300 FOR i:=1 TO 50000 DO
0320 ENDFOR i
```

AUFGABE:

Mache den Kreis so groß, daß er das ganze Quadrat ausfüllt. Versuche, den Kreis außerhalb des Quadrats zu zeichnen, und wähle verschiedene Farben.

FARBKREISE

Dieses Programm zeichnet 15 verschiedene Kreise, deren Lage auf dem Bildschirm durch den Zufall bestimmt wird. Auch ihre jeweilige Größe und Farbe werden zufällig bestimmt. Die Prozedur `warte` wurde ein wenig verändert. Sie enthält jetzt eine Variable (`zeit`), die ihr aus der Zeile 140 übergeben wird. So kannst du durch Veränderung der Zahl in Zeile 140 selbst bestimmen, wie lange der Computer warten soll.



```
0010 // Programm "farbkreise"  
0020  
0030 USE graphics  
0040  
0050 graphicscreen(1)  
0060 background(1)  
0070 border(1)  
0080
```

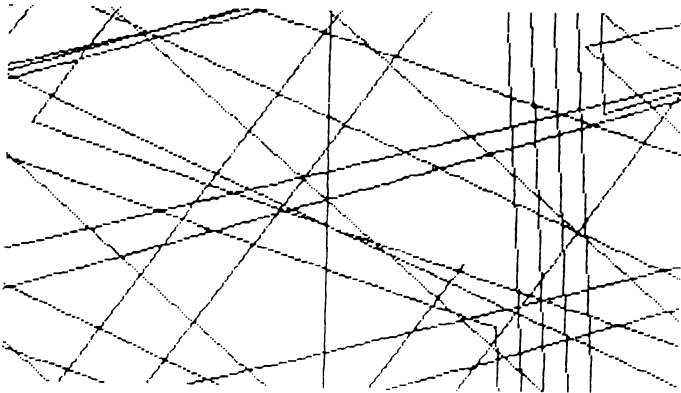
```
0090 LOOP
0100   FOR i:=1 TO 15 DO
0110     zeichne'kreis
0120   ENDFOR i
0130
0140   warte(6)
0150   clearscreen
0160 ENDLOOP
0170
0180 PROC zeichne'kreis
0190   x:=RND(40,230)
0200   y:=RND(40,158)
0210   radius:=RND(10,41)
0220   pencolor(RND(0,15))
0230   circle(x,y,radius)
0240   paint(x,y)
0250 ENDPROC zeichne'kreis
0260
0270 PROC warte(zeit)
0280   TIME (0)
0290   REPEAT
0300     UNTIL TIME>60*zeit
0310 ENDPROC warte
```

AUFGABE

Ändere das Programm doch einmal so ab, daß alle Kugeln gleich groß sind.

2. Es wird ein bißchen schwieriger

Das nächste Beispiel ist sehr einfach einzutippen, aber es ist nicht ganz so leicht zu verstehen. Sieh es dir einmal genau an! Dieses Kapitel zeigt unter anderem ein paar Beispiele, in denen die "Drehmethode" angewendet wird.



RND'STRICHE

Der Computer zeichnet einen Strich, dann wird der Zeichenstift (turtle) um ein zufälliges Stück gedreht (RND=random, das heißt "zufällig" auf englisch) und zeichnet erneut einen Strich. Weil die Striche aber sehr lang sind, rutschen sie aus der Zeichenfläche heraus und tauchen sofort auf der anderen Seite wieder auf. Nachdem 9 solcher Striche gezeichnet wurden, wird der Bildschirm gelöscht, und der ganze Vorgang beginnt von neuem.

```
0010 // Programm "rnd'striche"
0020
0030 USE turtle
0040
0050 background(1)
0060 border(2)
0070 graphicscreen(0)
0080
0090 FOR x:=1 TO 30 DO
0100   zeichnen; clear
0110 ENDFOR x
0120
0130
0140 PROC zeichnen
0150   hideturtle
0160   pencolor(2)
0170   FOR i:=1 TO 9 DO
0180     forward(900)
0190     right(RND(100,400))
0200   ENDFOR i
0210 ENDPROC zeichnen
```

AUFGABE:

Mache die Striche kürzer (Zeile 180), und beobachte, was geschieht!

HERZ

Aus zwei Hälften wird jetzt ein Herz zusammengesetzt. Danach wird es farbig ausgefüllt.



```
0010 // Programm "herz"  
0020  
0030 USE turtle  
0040 graphicscreen(1)  
0050  
0060 background(1)  
0070 border(2)  
0080 pencolor(2)  
0090  
0100 vollherz  
0110
```

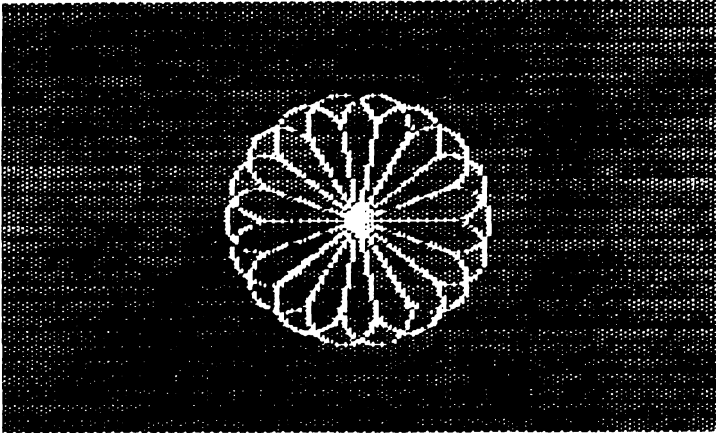
```
0120
0130 PROC vollherz
0140 penup; setxy(0,-80); pendown
0150 FOR x:=1 TO 20 DO
0160     seth(322); hideturtle
0170     herz
0180     fill(0,-30)
0190     warte
0200     clear
0210 ENDFOR x
0220 ENDPROC vollherz
0230
0240 PROC halbherz'1
0260 forward(130)
0270 arcr(45,218)
0280 ENDPROC halbherz'1
0290
0300 PROC halbherz'2
0310 right(180)
0320 arcr(45,218)
0330 forward(130)
0340 ENDPROC halbherz'2
0350
0360 PROC herz
0370 halbherz'1
0380 halbherz'2
0390 ENDPROC herz
0400
0410 PROC warte
0420 REPEAT
0430 UNTIL KEY$<>"
0440 ENDPROC warte
```

AUFGABE:

Kannst du das Herz auch kleiner machen und es an einen anderen Platz setzen? Probiere es!

DREHHERZ

Das Herz entsteht genau wie im vorigen Beispiel. Dann wird es aber 14 mal gedreht, und so entsteht die abgebildete Figur.



```
0010 // Programm "drehherz"
0020
0030 USE turtle
0040 graphicscreen(1)
0050
0060 background(2)
0070 border(1)
0080 pencolor(1)
0090 ht //Abkuerzung fuer 'hideturtle'
0100
0110 drehung
0120
```

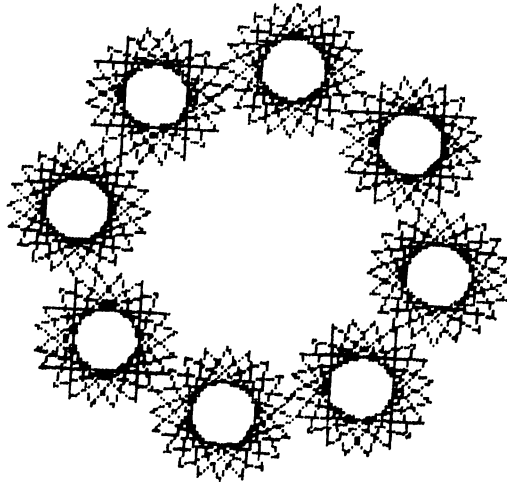
```
0130 PROC drehung
0140   FOR i:=1 TO 14 DO
0150     herz; right(10)
0160   ENDFOR i
0170   warte
0180 ENDPROC drehung
0190
0200 PROC halbherz'1
0210   left(10)
0220   forward(41)
0230   arcr(15,220)
0240 ENDPROC halbherz'1
0250
0260 PROC halbherz'2
0270   right(180)
0280   arcr(15,220)
0290   forward(41)
0300 ENDPROC halbherz'2
0310
0320 PROC herz
0330   halbherz'1
0340   halbherz'2
0350 ENDPROC herz
0360
0370 PROC warte
0380   REPEAT
0390     UNTIL KEY$<>""
0400 ENDPROC warte
```

AUFGABE:

Mache das Herz größer, so daß insgesamt eine größere Figur entsteht!

STERNE

Dieses Programm zeichnet 8 Sterne, die kreisförmig angeordnet stehen. Ein Bild, das wieder mit der Drehmethode entsteht.



```
0010 // Programm "sterne"  
0020  
0030 USE turtle  
0040 graphicscreen(0)  
0050  
0060 ht  
0070  
0080 background(1)  
0090 border(3)  
0100 pencolor(3)
```

```
0110
0120 sternkreis
0130
0140 PROC sternkreis
0150   FOR i:=1 TO 8 DO
0160     stern
0170     pu; back(30) //pu=penup
0180     right(45); pd //pd=pendown
0190   ENDFOR i
0200   warte
0210 ENDPROC sternkreis
0220
0230 PROC stern
0240   FOR i:=1 TO 20 DO
0250     quadrat
0260     right(36)
0270   ENDFOR i
0280 ENDPROC stern
0290
0300 PROC quadrat
0310   forward(40)
0320   right(90)
0330 ENDPROC quadrat
0340
0350 PROC warte
0360   REPEAT
0370     UNTIL KEY$<>""
0380 ENDPROC warte
```

AUFGABE:

Mache die Sterne kleiner, und lasse mehr Sterne zeichnen. Wenn du magst, mache sie doch einmal farbig. Du kannst die Zahlen für die verschiedenen Farben (den Farbcode) im Handbuch deines Computers finden.

3. Fahrräder, Katzen und Elefanten

Nun werden die Programme noch etwas schwieriger und die Aufgaben ebenso. Wenn du Lust hast, kannst du die Programme ganz verändern. Bevor du mit den folgenden Beispielen fortfährst, solltest du sicher sein, daß du auch alles verstanden hast.

FAHRRAD

Das Fahrrad ist sehr einfach, aber es liegt an dir, es besser zu machen.

```
0010 // Programm "fahrrad"
0020
0030 USE graphics
0040 USE turtle
0050
0060 background(5)
0070 border(7)
0080
0090 graphicscreen(0)
0100
0110 rad; speichen
0120 gestell; warte
0130
0140 PROC rad
0150   pencolor(0)
0160   circle(50,50,30)
0170   circle(130,50,30)
0180   pu; setxy(50,50)
0190   pd; speichen
0200   pu; setxy(130,50)
0210   pd; speichen
0220 ENDPROC rad
0230
0240 PROC speichen
```

```
0250  FOR i:=1 TO 18 DO
0260      pd; forward(30)
0270      back(30)
0280      right(20)
0290  ENDFOR i
0300  ENDPROC speichen
0310
0320  PROC warte
0330      REPEAT
0340      UNTIL KEY$<>""
0350  ENDPROC warte
0360
0370  PROC gestell
0380      pu; setxy(50,50)
0390      pd; right(20)
0400      forward(70)
0410      back(30)
0420      right(70)
0430      right(30)
0440      forward(100)
0450      back(100)
0460      left(100)
0470      forward(20)
0480      left(110)
0490      forward(20)
0500      back(40)
0510  ENDPROC gestell
0520
0530  PROC fahrrad
0540      rad
0550      gestell
0560      warte
0570  ENDPROC fahrrad
```

AUFGABE:

Versuche doch einmal, das Fahrrad durch einen Sattel und vielleicht sogar ein Schutzblech zu vervollständigen.

KOPF

Ein lustiger Kopf, nur aus Kreisen und einem Bogen.

```
0010 // Programm "kopf"
0020
0030 USE turtle
0040 USE graphics
0050 graphicscreen(1)
0060
0070 pencolor(0)
0080 background(3)
0090 border(2)
0100
0110 gesicht
0120
0130 PROC gesicht
0140   kopf
0150   augen
0160   mund
0170   nase
0180   ht; warte
0190 ENDPROC gesicht
0200
0210 PROC kopf
0220   circle(0,0,40)
0230   pencolor(1)
0240   fill(19,0)
0250   pencolor(0)
0260   penup
0270 ENDPROC kopf
0280
0290 PROC mund
0300   penup
0310   setxy(20,-15)
0320   pendown
0330   seth(180)
0340   arcrc(20,180)
```



```
0360 ENDPROC mund
0370
0380 PROC augen
0390   pd
0400   circle(-13,10,5)
0410   circle(13,10,5)
0420   pencolor(6)
0430   fill(-13,10)
0440   fill(13,10)
0450   pencolor(0)
0460 ENDPROC augen
0470
0480 PROC nase
0490   circle(0,-5,10)
0500   pencolor(2)
0510   fill(0,-5)
0520   pencolor(0)
0530 ENDPROC nase
0540
0550 PROC warte
0560   REPEAT
0570     UNTIL KEY$<>""
0580 ENDPROC warte
```

AUFGABE:

Lasse den Kopf mürrisch aussehen, indem du den Bogen umdrehst.

BLUME

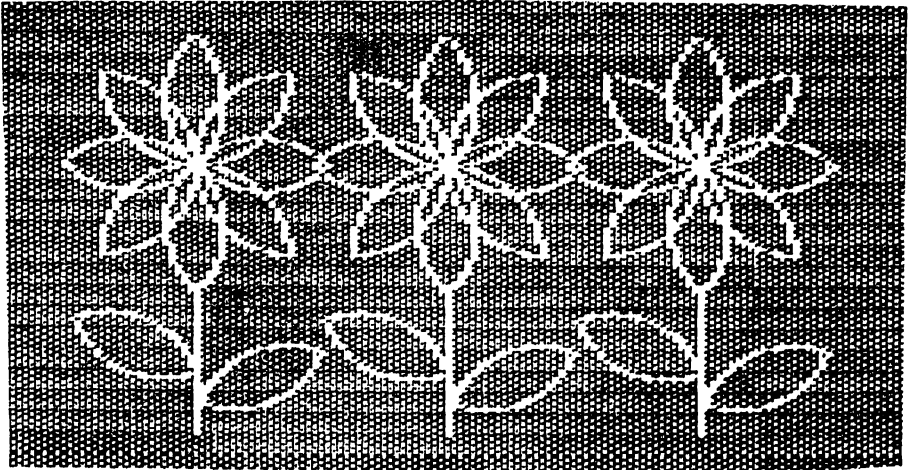
Die Blütenblätter werden genauso "gedreht" wie die Sterne. Die Blätter entstehen durch zwei Rechtsbögen (arcr).

```
0010 // Programm "blumen"
0020
0030 USE graphics
0040 USE turtle
0050
0060 graphicscreen(1)
0070
0080 pencolor(1)
0090 background(2)
0100 border(3)
0110 blumen
0120
0130 PROC blumen
0140   FOR i:=1 TO 3 DO
0150     verschiebung(80*i,100)
0160     blume
0170   ENDFOR i
0180   warte
0190 ENDPROC blumen
0200
0210 PROC blume
0220   FOR i:=1 TO 10 DO
0230     arcr(30,90); right(90)
0240     arcr(30,90)
0250     right(45)
0260   ENDFOR i
0270   stiel
0280 ENDPROC blume
0290
0300 PROC warte
0310   WHILE KEY$=CHR$(0) DO NULL
0320 ENDPROC warte
0330
```

```
0340 PROC stiel
0350   penup
0360   seth(180)
0370   pencolor(1)
0380   forward(40)
0390   pendown
0400   forward(50)
0410   blatt
0420   forward(20)
0430 ENDPROC stiel
0440
0450 PROC blatt
0460   pencolor(1)
0470   seth(0)
0480   arcr(30,110)
0490   right(90)
0500   arcr(30,90)
0510   seth(0)
0520   arcl(30,110)
0530   left(90)
0540   arcl(30,90)
0550   seth(180)
0560 ENDPROC blatt
0570
0580 PROC verschiebung(x,y)
0590   penup
0600   setxy(x,y)
0610   pendown
0620 ENDPROC verschiebung
```

AUFGABE:

Zeichne die Blumen einmal dichter zusammen, und erhöhe ihre Anzahl. Mache sie so bunt, wie es dir gefällt.



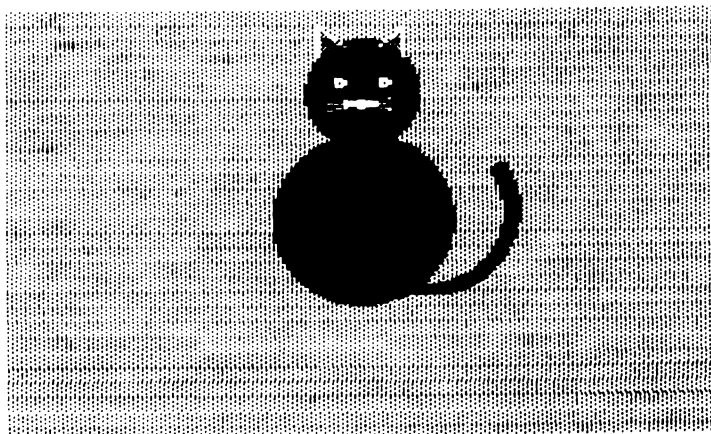
KATZE

Die Katze wird aus Kreisen und Bögen zusammengesetzt. Zwei spitze Winkel bilden die Ohren, und die Schnurrhaare bestehen aus sechs Strichen.

```
0010 // Programm "katze"
0020
0030 USE turtle
0040
0050 graphicscreen(1)
0060
0070 border(7)
0080 background(14)
0090 koerper; kopf
0100 schwanz; warte
0110
0120 PROC koerper
0130   pencolor(0)
0140   circle(0,0,40)
0150   paint(0,0)
0160 ENDPROC koerper
0170
0180 PROC kopf
0190   circle(0,60,25)
0200   augen
0210   schnurrhaare
0220   ohren
0230   pencolor(0)
0240   fill(0,50)
0250 ENDPROC kopf
0260
0270 PROC augen
0280   pencolor(1)
0290   circle(10,65,2)
0300   circle(-10,65,2)
0310 ENDPROC augen
0320
```

```
0330 PROC warte
0340   REPEAT
0350     UNTIL KEYS$<>"
0360   ENDPROC warte
0370
0380 PROC haar
0390   FOR i:=1 TO 3 DO
0400     forward(15)
0410     back(15)
0420     right(10)
0430   ENDFOR i
0440 ENDPROC haar
0450
0460 PROC schnurrhaare
0470   circle(0,55,2)
0480   pu; setxy(0,55)
0490   pd
0500   right(80)
0510   haar
0520   right(150)
0530   haar
0540 ENDPROC schnurrhaare
0550
0560 PROC ohren
0570   pu; seth(0)
0580   setxy(-19,77)
0590   pd
0600   forward(15)
0610   right(130)
0620   forward(15)
0630   pu; seth(0)
0640   setxy(19,77)
0650   pd
0660   forward(15)
0670   left(130)
0680   forward(15)
0690   pencolor(10)
0700   fill(-16,83)
0710   fill(16,83)
```

```
0720 ENDPROC ohren
0730
0740 PROC schwanz
0750 pu; seth(120)
0760 setxy(10,-30)
0770 pd
0780 pencolor(0)
0790 arcl(40,150)
0800 arcl(3,180)
0810 arcrc(35,150)
0820 fill(30,-32)
0840 ENDPROC schwanz
0850
```



AUFGABE:

Zeichne eine weiße Katze mit schwarzen Schnurrhaaren, Augen und Ohren. Mache die Schwanzspitze schwarz. Versuche, den Schwanz auch einmal länger oder kürzer zu zeichnen.

ELEFANT

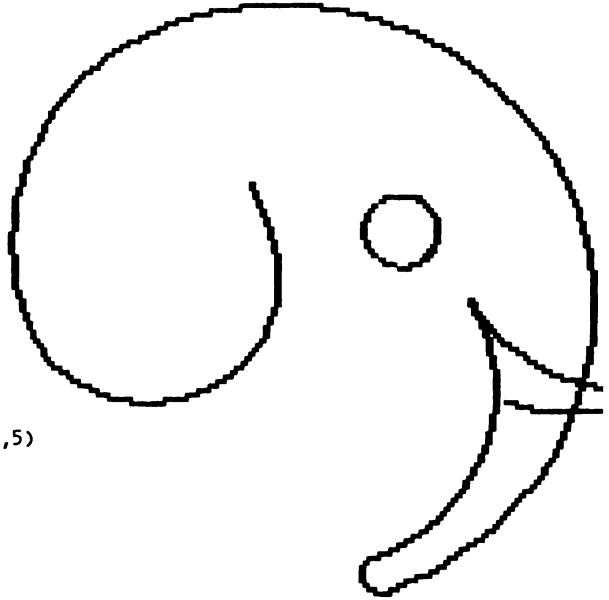
Der Elefant ist ebenfalls nur aus Kreisen und Bögen zusammengesetzt. Das Programm ist in viele einfache kurze Prozeduren unterteilt, die in der Prozedur **elefant** zusammengefaßt sind.

```
0010 // Programm "elefant"
0020
0030 USE turtle
0040
0050 graphicscreen(0)
0060
0070 background(15)
0080 border(15)
0090 pencolor(0)
0100
0110 elefant
0120
0130 PROC elefant
0140   ohr
0150   kopf
0160   ruessel
0170   zahn
0180   auge
0185   warte
0190 ENDPROC elefant
0200
0210 PROC ohr
0220   seth(180)
0230   arcrc(20,180)
0240 ENDPROC ohr
0250
0260 PROC kopf
0270   arcrc(50,210)
0280 ENDPROC kopf
0290
0300 PROC ruessel
0310   arcrc(4,180)
```

```

0320 arcl(25,80)
0330 ENDPROC ruessel
0340
0350 PROC zahn
0360 right(180)
0370 arcl(70,35)
0380 right(160)
0390 arcr(70,25); ht
0400 ENDPROC zahn
0410
0420 PROC auge
0430 pu; forward(25)
0440 right(90)
0450 forward(20); pd
0460 circle(xcor,ycor,5)
0470 ENDPROC auge
0480
0490 PROC warte
0500 REPEAT
0510 UNTIL KEY$<>""
0520 ENDPROC warte
0530

```



AUFGABE:

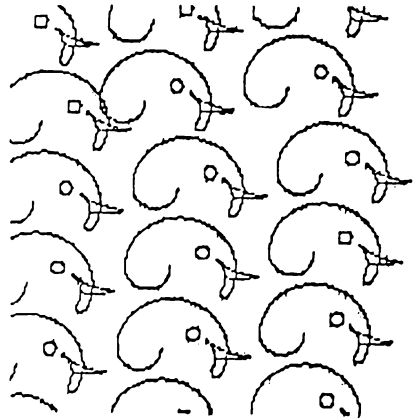
Vergrößere den Elefanten, und versuche, das Programm so zu erweitern, daß ein ganzer Elefant gezeichnet wird.



ELEFANTENHERDE

Hier wird die Prozedur `elefant` benutzt, um eine ganze Herde von Elefanten zu zeichnen. Die einzelnen Elefanten dürfen dann nur halb so groß sein wie im vorigen Beispiel.

```
0010 // Programm "elefanten"
0020
0030 USE turtle
0040
0050 graphicscreen(0)
0060
0070 background(15)
0080 border(15)
0090 pencolor(0)
0100
0110 elefanten
0120
0130 PROC elefant
0140   ohr
0150   kopf
0160   ruessel
0170   zahn
0180   auge
0190 ENDPROC elefant
0200
0210 PROC ohr
0220   seth(180)
0230   arcrc(10,180)
0240 ENDPROC ohr
0250
0260 PROC kopf
0270   arcrc(25,210)
0280 ENDPROC kopf
0290
0300 PROC ruessel
0310   arcrc(2,180)
0320   arcl(12,80)
```



```
0330 ENDPROC ruessel
0340
0350 PROC zahn
0360   right(180)
0370   arcl(35,35)
0380   right(160)
0390   arcr(35,25); ht
0400 ENDPROC zahn
0410
0420 PROC auge
0430   pu; forward(12)
0440   right(90)
0450   forward(10); pd
0460   circle(xcor,ycor,3)
0470 ENDPROC auge
0480
0490 PROC warte
0500   REPEAT
0510     UNTIL KEY$<>""
0520 ENDPROC warte
0530
0540 PROC elefanten
0550   pu; setxy(-100,50)
0560   pd; ht
0570   FOR i:=1 TO 25 DO
0580     elefant
0590     pu; seth(90)
0600     forward(50); pd
0610   ENDFOR i
0620   warte
0630 ENDPROC elefanten
```

AUFGABE:

Zeichne die Elefanten noch kleiner, aber dafür doppelt so viele. Probiere auch, weniger Elefanten auf dem Bildschirm entstehen zu lassen.

4. COMAL und sprites

In diesem Kapitel steht unter anderem, wie du sprites benutzen kannst und wie du Programmteile mit sprites an deine anderen Programme anfügen ("merge") kannst. Auf diese Weise kannst du einen sprite, den du in einem anderen Programm entwickelt hast, wiederverwenden, ohne ihn neu zu zeichnen. Wie das gemacht wird, erfährst du, wenn du die Programme dieses Kapitels liest und eintippst.

```
0010 // Programm "sonnen'sprite"
0020
0030 USE graphics
0040
0050 background(6)
0060 border(7)
0070
0080 DATA %00000000,%00000000,%00000000
0090 DATA %00000000,%00000001,%11100000
0100 DATA %00001111,%11000111,%10000000
0110 DATA %00111111,%11111111,%11111100
0120 DATA %00000011,%11111111,%11111110
0130 DATA %00000111,%11000011,%11110000
0140 DATA %00001111,%10000001,%11100000
0150 DATA %00011111,%00000000,%11111000
0160 DATA %00111111,%00000000,%01111100
0170 DATA %01111110,%00000000,%01111110
0180 DATA %11111110,%00000000,%01100111
0190 DATA %11000011,%00000000,%01000011
0200 DATA %00000111,%00000000,%11100000
0210 DATA %00001111,%10000001,%11110000
0220 DATA %00011111,%11000011,%11111000
0230 DATA %00111111,%11111110,%11111100
0240 DATA %01111100,%01111100,%00001110
0250 DATA %11100000,%00001100,%00000011
0260 DATA %11000000,%00000000,%00000000
0270 DATA %00000000,%00000000,%00000000
```



```
0280 DATA %00000000,%00000000,%00000000
0290
```

SONNEN-SPRITE

In den Zeilen 80 bis 280 siehst du den 'sprite', der aus Einsern besteht. Die Befehle in den Zeilen 360 bis 530 sind nötig, um den sprite zeichnen zu lassen. Die Größe der Sonne kannst du in Zeile 510 ändern. Wenn du das Programm startest, siehst du die Sonne. Wenn du dann eine Taste drückst, bewegt sie sich von der einen Seite des Bildschirms zur anderen. Ein erneuter Tastendruck läßt das Programm stoppen.

```
0300 USE graphics
0310
0320 graphicscreen(0)
0330
0340 USE sprites
0350
0360 DIM zeichnung$ OF 64
0370
0380 FOR i:=1 TO 63 DO
0390   READ binaerzahlen
0400   zeichnung$:+CHR$(binaerzahlen)
0410 ENDFOR i
0420
0430 farbe:=7
0440 zeichnungsnr:=1
0450 spritenr:=1
0460
0470 define(zeichnungsnr,zeichnung$+""0""")
0480 identify(spritenr,zeichnungsnr)
0490 spritecolor(spritenr,farbe)
0500 spritepos(spritenr,50,100)
0510 spritesize(1,1,1)
0520 showsprite(spritenr)
0530 priority(1,1)
0540
```

```
0550
0560 WHILE KEY$=CHR$(0) DO NULL
0570
0580 movesprite(spritennr,250,150,200,0)
0590
0600 WHILE KEY$=CHR$(0) DO NULL
```

AUFGABE:

Wähle für die Sonne doch einmal eine andere Farbe. Du kannst die Sonne auch ganz verändern, indem du die Einser im sprite neu festlegst. Vergiß nicht, deine geänderten Programme auch zu speichern.

MERGE:

Du kannst die Sonne in das gezeigte Bild "mergen" (übertragen), zu dem du das Programm im Anhang dieses Buches findest. Du siehst dort das Programm "haus", in dem viele Prozeduren zu einem ganzen Bild zusammengesetzt sind. Wenn du dieses Programm eingetippt hast, kannst du die Aufgabe auf der nächsten Seite lösen.



Wenn du die folgende Aufgabe gelöst hast und das Programm "haus" startest, ist auch die Sonne in der Zeichnung zu sehen, und sie steigt hinter den Bäumen und hinter dem Haus hinauf zum Himmel.

Nachdem du soweit gekommen bist, solltest du das Programm "haus" unter einem anderen Namen speichern. Du hast dann zwei "haus"-Programme, eines mit und ein anderes ohne Sonne. Das Programm mit der Sonne kannst du beispielsweise "sonnenhaus" nennen. Wenn du eigene sprite-Programme entwerfen willst, änderst du nur das vorige sprite-Programm und speicherst es dann unter einem anderen Namen.

Denke immer daran, deine Programme zu speichern (save)!

AUFGABE:

Merge den sprite in das "haus"-Programm. Auf diese Weise mußt du nicht extra ein neues "sonne"-Programm schreiben. Du kopierst die Sonne einfach in das Programm "haus".

Dazu mußt du folgendes tun:

- * Nachdem du das Programm "sonne" geladen hast, löschst du den unnötigen Teil, indem du folgendes schreibst:

del 20-60, 300-330, 560-600 drücke dann die RETURN-Taste.
- * Schreibe: list "sprite'sonne.lst" (+ RETURN-Taste). Lade dann das Programm "haus", so wie es in der Programmsammlung im Anhang steht.
- * Schreibe: merge "sprite'sonne.lst" (+ RETURN-Taste). Nun hast du die Sonne in das andere Programm kopiert. Wenn du RUN eingibst, siehst du ein Programm mit aufgehender Sonne, das "sonnenhaus".
- * Und noch einmal: Denke daran, dein Programm zu speichern!

AUTOTANZ

Noch ein interessantes sprite-Programm: Jedesmal, wenn du auf eine Taste drückst, bewegt sich eines der Autos ein Stück. Es sieht lustig aus, wenn man die Leertaste gedrückt hält; es sieht so aus, als ob die Autos tanzen.

```
0010 // Programm "autotanz"
0015
0020 vorbereitung
0030
0040 LOOP
0050   FOR spritenr:=1 TO 4 DO
0060     movesprite(spritenr,10+40*spritenr,10,20,0)
0070     warte
0080     movesprite(spritenr,15+40*spritenr,184,20,0)
0090     warte
0100     movesprite(spritenr,305-40*spritenr,184,25,0)
0110     warte
0120     movesprite(spritenr,305-40*spritenr,15,25,0)
0130     warte
0140     movesprite(spritenr,15+40*spritenr,15,25,0)
0150     movesprite(spritenr,100+40*spritenr,100,40,0)
0200   ENDFOR spritenr
0210 ENDOLOOP
0220
0230 // das Auto
0240 DATA %00000000,%00000000,%00000000
0250 DATA %00000011,%00000000,%11000000
0260 DATA %00000011,%11111111,%11000000
0270 DATA %00000011,%11111111,%11000000
0280 DATA %00001101,%11111111,%10110000
0290 DATA %00001101,%11111111,%10110000
0300 DATA %00001111,%00000000,%11110000
0310 DATA %00001101,%00000000,%10110000
0320 DATA %00001101,%01111110,%10110000
0330 DATA %00000011,%01111110,%11000000
0340 DATA %00000011,%01111110,%11000000
```

```

0350 DATA %00000011,%01111110,%11000000
0360 DATA %00000011,%01111110,%11000000
0370 DATA %00000011,%00000000,%11000000
0380 DATA %00001101,%11111111,%10110000
0390 DATA %00001101,%11111111,%10110000
0400 DATA %00001111,%11111111,%11110000
0410 DATA %00001101,%11111111,%10110000
0420 DATA %00001101,%11111111,%10110000
0430 DATA %00000011,%11111111,%11000000
0440 DATA %00000000,%11000000,%00000000
0450
0460 PROC vorbereitung
0470   USE graphics
0480   graphicscreen(0)
0490   USE sprites
0500
0510   spritesize(1,2,2)
0520   spritesize(2,2,2)
0530   spritesize(3,2,2)
0540   spritesize(4,2,2)
0550
0560   DIM zeichnung$ OF 64
0570   FOR i:=1 TO 63 DO
0580     READ binaerzahlen
0590     zeichnung$:+CHR$(binaerzahlen)
0600   ENDFOR i
0610
0620   farbe:=1
0630   zeichnungsnr:=1
0640
0650   FOR spritenr:=1 TO 4 DO
0660     define(zeichnungsnr,zeichnung$+""0"")
0670     identify(spritenr,zeichnungsnr)
0680     spritecolor(spritenr,spritenr)
0690     spritepos(spritenr,50+20*spritenr,100)
0700     showsprite(spritenr)
0710   ENDFOR spritenr
0720 ENDPROC vorbereitung
0730
0740 PROC warte
0750   WHILE KEY$=CHR$(0) DO NULL
0760 ENDPROC warte
0770

```



5. Und was nun?

Du hast dir nun in einer ganzen Reihe von Beispielen angesehen, wie man COMAL 80 zum Zeichnen benutzen kann. Du solltest jetzt auch eigene Programme schreiben können, und wenn du dich an einzelne Dinge nicht erinnern kannst, dann sieh im Stichwortverzeichnis nach. Du kannst nun also zeichnen, wozu du Lust hast, und obendrein kannst du sprites verwenden, die sich über den Bildschirm bewegen.

Spiele machen sicher Spaß, aber du hast nun gesehen, daß man seine eigenen Programme machen kann, und wenn du Lust hast - auch Spiele. COMAL bietet noch viel mehr Möglichkeiten: nicht nur für Zeichenprogramme, sondern zum Beispiel auch für Musik- und Rechenprogramme. Und wenn du COMAL in der Schule hast, kannst du da natürlich auch prima Sachen machen. Wenn du das COMAL-Modul besitzt, dann hast du auch ein Handbuch, in dem COMAL eingehend beschrieben ist und wo Möglichkeiten aufgezeigt sind, von denen in diesem Buch überhaupt nicht gesprochen wurde.

Du kannst dich natürlich auch an die COMAL-Gruppe Deutschland in 2270 Utersum/Föhr (Tel.: 04683/500) wenden, die in ihren COMAL-News Programme und Kurse veröffentlicht.

Man kann in COMAL auch Programme schreiben, in denen Paddles und Joystick benutzt werden. Versuche es nur! COMAL bietet so viele Möglichkeiten, daß du fast alles machen kannst.

Es folgt in diesem Buch jetzt noch eine Programmsammlung mit einigen Programmen verschiedener Größe. Du kannst sie einmal ausprobieren, und du kannst sie verändern, wie du möchtest.

Versuche doch einmal, ein eigenes Programm zu schreiben!

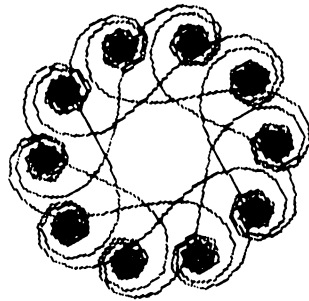
Anhang: Programmausdrucke

Hier siehst du einige Programme, die du eintippen kannst, wenn du Lust hast. Es ist unter anderem ein Programm "haus" dabei, das auch im Kapitel **COMAL und sprites** benutzt wurde.

DREHSTRICH

Ein kurzes, tolles Programm. Wenn du die Zahlen in Zeile 80 veränderst, kannst du Hunderte verschiedener Figuren gestalten.

```
0010 // Programm "drehstrich"
0020
0030 USE turtle
0040 hideturtle
0050
0060 graphicscreen(0)
0070
0080 drehstrich(10,10,5)
0090
0100 PROC drehstrich(x,y,z)
0110   forward(x)
0120   right(y)
0130   drehstrich(x,y+z,z)
0140 ENDPROC drehstrich
```



DREHFLÄCHE

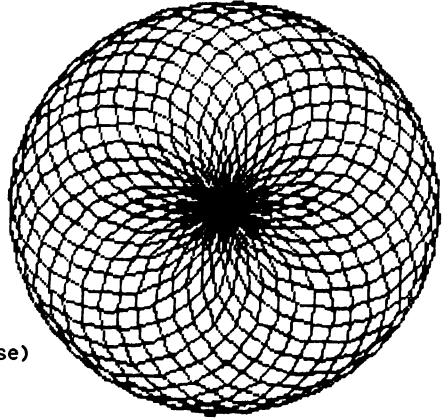
Bevor das Programm zu arbeiten beginnt, mußt du eine der Tasten "2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, oder f" drücken. Du bekommst dann ein entsprechendes Vieleck, das sich dreht.

```
0010 // Programm "drehflaeche"
0020
0030 USE turtle
0040 USE graphics
```

```

0050
0060 graphicscreen(0)
0070
0080 hole'zahl
0085 x:=eckenzahl
0090 background(x); border(x)
0100
0110 drehflaeche(x,80-6*s)
0120
0130 warte
0140
0150 PROC vieleck(eckenzahl,groesse)
0160   FOR i:=1 TO eckenzahl DO
0170     forward(groesse)
0180     right(360/eckenzahl)
0190   ENDFOR i
0200 ENDPROC vieleck
0210
0220 PROC warte
0230   WHILE KEY$="" DO NULL
0240 ENDPROC warte
0250
0260 PROC drehflaeche(eckenzahl,groesse)
0270   ht
0280   FOR i:=1 TO 36 DO
0290     right(10)
0300     vieleck(eckenzahl,groesse)
0310   ENDFOR i
0320 ENDPROC drehflaeche
0330
0340 PROC hole'zahl
0350   REPEAT
0360     zahl$:=KEY$
0370   UNTIL zahl$<>"" AND zahl$ IN "23456789abcdef"
0380   IF zahl$ IN "23456789" THEN
0390     eckenzahl:=VAL(zahl$); s:=eckenzahl
0400   ELSE
0410     s:=10
0420   CASE zahl$ OF

```



```

0430     WHEN "a"
0440         eckenzahl:=10
0450     WHEN "b"
0460         eckenzahl:=11
0470     WHEN "c"
0480         eckenzahl:=12
0490     WHEN "d"
0500         eckenzahl:=13
0510     WHEN "e"
0520         eckenzahl:=14
0530     WHEN "f"
0540         eckenzahl:=15
0550     ENDCASE
0560     ENDIF
0570 ENDPROC hole'zahl

```

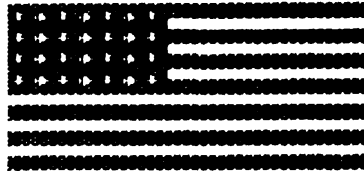
USA-FLAGGE

Ein tolles Programm - und sogar mit Sternen!

```

0010 // Programm "usa'flagge"
0020
0030 USE graphics
0050 USE turtle
0060
0070 pencolor(2)
0080 background(1)
0090 border(6)
0100 graphicscreen(1)
0110
0120 ueberschrift
0130 pu
0140 setxy(100,150); pd
0150 pd
0160 flagge(1,1)
0170 warte
0180 END
0190

```



```
0200 PROC ueberschrift
0210   textstyle(2,2,0,0)
0220   plottext(30,170,"Flagge der USA")
0230
0240 ENDPROC ueberschrift
0250
0260 PROC flaggenrechteck(hoehe,breite)
0270   pendown
0280   forward(hoehe)
0290   right(90)
0300   forward(breite)
0310   right(90)
0320   forward(hoehe)
0330   right(90)
0340   forward(breite)
0350   right(90)
0360   ausmalen
0370 ENDPROC flaggenrechteck
0380
0390 PROC ausmalen
0400   pu
0410   forward(5*ay)
0420   right(90)
0430   forward(5*ax)
0440   pd
0450   fill(xcor,ycor)
0460   back(5*ax)
0470   left(90)
0480   back(5*ay)
0490 ENDPROC ausmalen
0500
0510 PROC flagge(ax,ay)
0520   FOR i:=1 TO 4 DO
0530     pendown
0540     flaggenrechteck(6*ay,90*ax)
0550     penup
0560     back(12*ay); pendown
0570   ENDFOR i
0580   penup
```

```
0590 left(90)
0600 forward(70*ax)
0610 right(90)
0620 pendown
0630 FOR i:=1 TO 3 DO
0640     flaggenrechteck(6*ay,160*ax)
0650     penup
0660     back(12*ay)
0670     pendown
0680 ENDFOR i
0690 platz'fuer'sterne(ay,ax)
0700 ENDPROC flagge
0710
0720 PROC sterne(zahl1,zahl2)
0730 FOR i:=1 TO zahl2 DO
0740     stricheln(zahl1)
0750 ENDFOR i
0760 forward(8); left(90); forward(2)
0770 FOR i:=1 TO zahl1 DO
0780     stricheln(zahl2)
0790 ENDFOR i
0800 forward(zahl2*10-2); right(90)
0810 back(zahl1*10+8)
0820 ENDPROC sterne
0830
0840 PROC stricheln(anzahl)
0850 FOR i:=1 TO anzahl DO
0860     penup
0870     forward(7)
0880     pendown
0890     forward(3)
0900 ENDFOR i
0910 penup
0920 back(10*anzahl)
0930 left(90)
0940 back(10)
0950 right(90)
0960 ENDPROC stricheln
0970
```

```

0980 PROC platz'fuer'sterne(ay,ax)
0990   pencolor(6)
1000   penup
1010   setxy(30,114); pendown
1020   flaggenrechteck(42*ay,70*ax)
1030   penup
1040   right(90)
1050   forward(5)
1060   left(90)
1070   back(2)
1080   pendown
1090   pencolor(1)
1100   sterne(5,7)
1110 ENDPROC platz'fuer'sterne
1120
1130 PROC warte
1140   REPEAT UNTIL KEY$<>""
1150 ENDPROC warte

```

HAUS

Ein Programm, das ein Haus mit Garten zeichnet. Es ist aufgeteilt in viele kleine Prozeduren. Allein zum Zeichnen des Baumes werden fünf verschiedene verwendet. Dieses Programm wird auch benutzt, um daran den sprite 'sonne' anzufügen (siehe Kapitel COMAL und sprites).

```

0010 // Programm "haus"
0020
0030 USE turtle
0040 USE graphics
0050 hideturtle
0060
0070 graphicscreen(1)
0080 border(7)
0090
0100 haus
0110 warte

```



```
0120
0130 PROC dach
0140   right(45)
0150   forward(60)
0160   right(90)
0170   forward(60)
0180   right(135)
0190   forward(84)
0200   pencolor(2)
0210   fill(20,20)
0220 ENDPROC dach
0230
0240 PROC rechteck(hoehe,breite)
0250   FOR i:=1 TO 2 DO
0260     forward(hoehe)
0270     right(90)
0280     forward(breite)
0290     right(90)
0300   ENDFOR i
0310 ENDPROC rechteck
0320
0330 PROC warte
0340   WHILE KEY$="" DO NULL
0350 ENDPROC warte
0360
0370 PROC fenster'tuer
0380   penup; setxy(10,-15)
0390   pendown
0400   rechteck(15,15)
0410   penup
0420   right(90)
0430   forward(25)
0440   left(90)
0450   back(15)
0460   pendown
0470   rechteck(30,15)
0490 ENDPROC fenster'tuer
0500
0510 PROC haus
```

```
0520 penup
0530 setxy(0, -30)
0540 pendown
0550 rechteck(40,80)
0560 penup
0570 setxy(-2,10)
0580 pendown
0590 fenster'tuer
0600 penup
0610 setxy(-2,10)
0620 pendown
0630 dach
0640 penup
0650 setxy(0, -30)
0660 pendown
0670 gras
0680 pencolor(3)
0690 fill(20, -20)
0700 penup
0710 setxy(-130, -30)
0720 pendown
0730 baeume(3,10)
0740 ht
0750 ENDPROC haus
0760
0770 PROC gras
0780 setheading(90)
0790 pencolor(5)
0800 forward(400)
0810 right(90)
0820 back(5)
0830 paint(0, -80)
0840 ENDPROC gras
0850
0860 PROC v(s)
0870 left(45)
0880 forward(s)
0890 back(s)
0900 right(90)
```

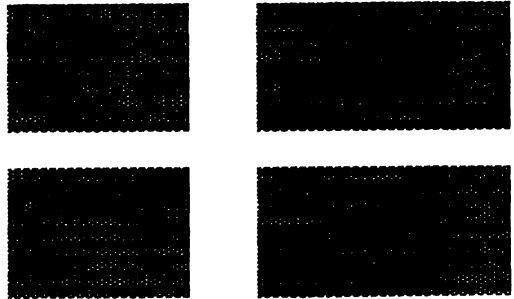
```
0910 forward(s)
0920 back(s)
0930 left(45)
0940 ENDPROC v
0950
0960 PROC ast(s)
0970 forward(s)
0980 v(s)
0990 forward(s)
1000 v(s)
1010 back(s*2)
1020 ENDPROC ast
1030
1040 PROC krone
1050 left(60)
1060 FOR i:=1 TO 7 DO
1070     ast(s)
1080     right(20)
1090 ENDFOR i
1100 //ast(s)
1110 left(60)
1120 ENDPROC krone
1130
1140 PROC baum(s)
1150 pencolor(0)
1160 setheading(0)
1170 forward(s*2)
1180 pencolor(13)
1190 krone
1200 penup
1210 back(s*3.5)
1220 ENDPROC baum
1230
1240 PROC baeume(anzahl,s)
1250 pendown
1260 FOR i:=1 TO (anzahl) DO
1270     baum(s)
1280     penup
1290     right(72)
```

```
1300 forward(50)
1310 left(90)
1320 pendown
1330 s:+1
1340 ENDFOR i
1350 ENDPROC baeume
```

DÄNISCHE FLAGGE

Dieses Programm zeichnet die dänische Flagge. Nach einer warte-
Prozedur werden 20 kleine Flaggen auf den Bildschirm gezeichnet.

```
0010 // Programm "daenische'flagge"
0020
0040 USE graphics
0050 USE turtle
0060
0070 pencolor(2)
0080 background(1)
0090 border(1)
0100 graphicscreen(1)
0110
0120 LOOP
0130 ueberschrift
0140 pu
0150 setxy(50,20); pd
0160 flagge(1.6,1.2)
0170 warte
0180 clearscreen
0190 viele'flaggen
0200 warte
0210 clearscreen
0220 ENDLOOP
0230
0240 PROC flaggenrechteck(hoehe,breite)
0250 pendown
0260 forward(hoehe)
0270 right(90)
```



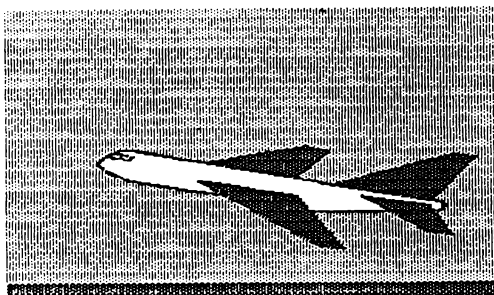
```
0280 forward(breite)
0290 right(90)
0300 forward(hoehe)
0310 right(90)
0320 forward(breite)
0330 right(90)
0340 ausfuellen
0350 ENDPROC flaggenrechteck
0360
0370 PROC ausfuellen
0380 pu
0390 forward(10*ay)
0400 right(90)
0410 forward(10*ax)
0420 pd
0430 fill(xcor,ycor)
0440 back(10*ax)
0450 left(90)
0460 back(10*ay)
0470 ENDPROC ausfuellen
0480
0490 PROC flagge(ax,ay)
0500 FOR i:=1 TO 2 DO
0510     flaggenrechteck(50*ay,50*ax)
0520     right(90)
0530     penup
0540     forward(70*ax)
0550     left(90)
0560     pendown
0570     flaggenrechteck(50*ay,70*ax)
0580     IF i=1 THEN
0590         penup
0600         forward(65*ay)
0610         left(90)
0620         forward(70*ax)
0630         right(90)
0640         pendown
0650     ENDIF
0660 ENDFOR i
```

```
0670 pu
0680 left(90)
0690 forward(90*ax)
0700 right(90)
0710 back(65*ay)
0720 ENDPROC flagge
0730
0740 PROC warte
0750 FOR i:=1 TO 10000 DO
0760 ENDFOR i
0770 ENDPROC warte
0780
0790 PROC viele'flaggen
0800 x:=0
0810 REPEAT
0820 y:=0
0830 REPEAT
0840 pu
0850 setxy(x,y)
0860 pd
0870 flagge(.3,.3)
0880 y:+50
0890 UNTIL y=200
0900 x:+60
0910 UNTIL x=300
0920 ENDPROC viele'flaggen
0930
0940 PROC ueberschrift
0950 pencolor(2)
0960 textstyle(2,2,0,0)
0970 plottext(45,175,"DAENISCHE FLAGGE")
0980 ENDPROC ueberschrift
0990
```

FLUGZEUG

Ein schönes Flugzeug - sogar mit Ton. Drehe den Lautstärke-
regler auf, und höre den Düsenlärm.

```
0010 // Programm "flugzeug"
0020
0030 USE turtle
0040 USE sound
0050
0060 graphicscreen(1)
0070
0080 pencolor(0)
0090 border(0)
0100 background(3)
0110
0120 meer; flugzeug
0130
0140 PROC flugzeug
0150   rumpf
0160   leitwerk
0170   fluegel'1
0180   fluegel'2
0190   hinterer'fluegel
0200   cockpit
0210   pencolor(1)
0220   fill(30,-20)
0230   pause
0240   duesen'motor
0250   warte
0260 ENDPROC flugzeug
0270
0280 PROC rumpf
0290   pu; setxy(40,-20); pd
0300   seth(0)
0310   left(80)
0320   forward(120)
0330   arcl(25,60)
0340   left(90)
0350   arcl(55,30)
0360   forward(60)
0370   penup
0380   forward(55)
```



```
0390 pendown
0400 forward(5)
0410 left(10)
0420 forward(44)
0430 penup
0440 forward(30)
0450 pendown
0460 forward(4)
0470 arcl(3,170)
0480 forward(80)
0490 ENDPROC rumpf
0500
0510 PROC warte
0520 WHILE KEY$=CHR$(0) DO NULL
0530 ENDPROC warte
0540
0550 PROC leitwerk
0560 back(10)
0570 right(155)
0580 forward(80)
0590 right(25)
0600 forward(15)
0610 right(125)
0620 forward(40)
0630 pencolor(2)
0640 fill(80,-25)
0650 ENDPROC leitwerk
0660
0670 PROC fluegel'1
0680 pu; setxy(-35,-20); pd
0690 pencolor(0)
0700 seth(120)
0710 forward(90)
0720 left(25)
0730 forward(15)
0740 left(135)
0750 forward(47)
0760 left(40)
0770 forward(65)
```

```
0780  pencolor(2)
0790  fill(-15,-30)
0800  pencolor(0)
0810  ENDPROC fluegel'1
0820
0830  PROC fluegel'2
0840  pu; setxy(-30,-7); pd
0850  seth(80)
0860  forward(65)
0870  right(25)
0880  forward(15)
0890  right(125)
0900  forward(27)
0910  pencolor(2)
0920  fill(0,-10)
0930  pencolor(0)
0940  ENDPROC fluegel'2
0950
0960  PROC hinterer'fluegel
0970  pu; setxy(70,-30); pd
0980  seth(120)
0990  forward(50)
1000  left(22)
1010  forward(15)
1020  left(133)
1030  forward(25)
1040  left(47)
1050  forward(40)
1060  pencolor(2)
1070  fill(80,-35)
1080  pencolor(0)
1090  ENDPROC hinterer'fluegel
1100
1110  PROC cockpit
1120  pu; setxy(-95,-3); pd
1130  seth(97)
1140  forward(15)
1150  left(45)
1160  forward(3)
```

```
1170 left(130)
1180 forward(10); ht
1190 ENDPROC cockpit
1200
1210 PROC meer
1220 pu; setxy(0,-90); pd
1230 pencolor(6)
1240 seth(90)
1250 forward(400)
1260 fill(0,-95)
1270 pencolor(0)
1280 ENDPROC meer
1290
1300 PROC duesen'motor
1310 soundtype(1,4)
1320 note(1,"c4")
1330 gate(1,1)
1340 FOR t:=0 TO 1000 DO
1350     volume(15*t/1000)
1360 ENDFOR t
1370 FOR t:=1000 TO 0 STEP -1 DO
1380     volume(15*t/1000)
1390 ENDFOR t
1410 ENDPROC duesen'motor
1420
1430 PROC pause
1440 FOR i:=1 TO 5000 DO
1450 ENDFOR i
1460 ENDPROC pause
```

Weitere Bücher über COMAL

Neuber/Sperling
COMAL-2.01-Handbuch
DM 39,--
D.Sperling Hard- und Software Niebüll 1986

Christensen, B.
Strukturierte Programmierung mit Comal
DM 29,80
Oldenbourg Verlag München, Wien 1985

Christensen, B.
Das COMAL-80 Buch
DM 25,--
Oldenbourg Verlag München 1986

Christensen, B.; Wolgast, H.
Comal 0.14-Handbuch
Schmidt und Klaunig Kiel 1984

Fischer, V.
Comal in Beispielen
DM 24,80
Teubner Stuttgart 1986

Landesinstitut Schleswig-Holstein für Praxis und Theorie der Schule
(Hrsg)
Informationstechnische Grundbildung Band 1
Materialien für die Klassenstufe 8
DM 26,--
Aulis Verlag Deubner & Co KG Köln 1986

Stichwortverzeichnis

Autotanz	43
Blume	29
Dänische Flagge	55
Drehfläche	46
Drehherz	21
Drehstrich	48
Drehung	12
Elefant	35
Elefantenherde	37
Fahrrad	25
Farbkreise	15
Flugzeug	57
Haus	51
Herz	19
Katze	32
Kopf	27
RND-Striche	17
merge	41, 42
Quadrat	11
Quadrat und Kreis	13
Sonnen-sprite	40
sprites	39
Sterne	23
USA-Flagge	48

.... und unverzichtbar für jeden COMAL-Anwender ist natürlich das

COMAL HANDBUCH

Das Handbuch bringt in einem klaren Aufbau in verständlicher Ausdrucksweise zu den einzelnen Stichwörtern viele erläuternde Beispiele, die auch dem Einsteiger die Wirkungsweise der einzelnen Befehle ausführlich erklären, und der Fortgeschrittene erhält ein Nachschlagewerk, in dem er sich immer wieder schnell über alle Möglichkeiten, die COMAL bietet, informieren kann.



er alle Möglichkeiten von COMAL voll ausschöpfen und auch in technischen Einzelheiten gut beraten sein will, dem wird dieses Handbuch bald zum unentbehrlichen Helfer werden.

204 Seiten kart. DM 39,--

HÄUFIG VERWENDETE COMAL-BEFEHLE UND SOFTWARE-PACKETE:

COMAL SCHLÜSSELWÖRTER:

Prozedur	PROC...ENDPROC
Schleifen	FOR...TO...STEP...DO...ENDFOR REPEAT...UNTIL WHILE...ENDWHILE LOOP...EXIT WHEN...ENDLOOP
Verzweigungen	IF...THEN...ELIF...ELSE...ENDIF CASE...OF...WHEN...OTHERWISE...ENDCASE
DEL	<i>löschen von Zeilen (z.B. DEL 100-200)</i>
LIST	<i>auflisten des Programms</i>
MERGE	<i>Zusammenfügen verschiedener Programme oder Programmteile</i>
LOAD	<i>lädt ein Programm von Diskette</i>
SAVE	<i>speichert ein Programm auf Diskette</i>
RUN	<i>startet ein Programm</i>

EINIGE GRAPHIK-PROZEDUREN

USE graphics	Das Software-Paket 'graphics' wird aktiviert.
USE turtle	Das Software-Paket 'turtle' wird aktiviert, es können jetzt auch die Abkürzungen pu , pd , fd , bk , rt , lt , usw. benutzt werden.
penup	<i>Zeichenstift wird angehoben</i>
pendown	<i>Zeichenstift ist bereit zum Schreiben</i>
forward(x)	<i>Zeichenstift wird um x Schritte vorwärts bewegt</i>
back(x)	<i>Zeichenstift wird um x Schritte zurück bewegt</i>
left(x)	<i>Zeichenstift wird um x Grad nach links gedreht</i>
right(x)	<i>Zeichenstift wird um x Grad nach rechts gedreht</i>
setheading(x)	<i>Zeichenstift wird in eine bestimmte Richtung gedreht, z.B. für x=0 genau nach oben, für x=90 nach rechts, entsprechend den Zahlen eines Kompasses</i>
pencolor(x)	<i>bestimmt die Schreibfarbe des Zeichenstiftes</i>
background(x)	<i>bestimmt die Untergrundfarbe des Graphikbildschirms</i>
border(x)	<i>bestimmt die Farbe des Bildschirmrandes</i>
circle(x,y,r)	<i>zeichnet einen Kreis mit dem Radius r und den Mittelpunktskoordinaten x,y</i>

FARBCODE:

0	<i>schwarz</i>	4	<i>violett</i>	8	<i>orange</i>	12	<i>grau</i>
1	<i>weiß</i>	5	<i>grün</i>	9	<i>braun</i>	13	<i>hellgrün</i>
2	<i>rot</i>	6	<i>blau</i>	10	<i>hellrot</i>	14	<i>hellblau</i>
3	<i>türkis</i>	7	<i>gelb</i>	11	<i>dunkelgrau</i>	15	<i>hellgrau</i>

Eingehendere Informationen findest du im **COMAL Handbuch**, Kapitel 'graphics/turtle'.



SILKEBORG SOLDATA

ISBN: 87-982231-2-7



**This was brought to you
from the archives of**

<http://retro-commodore.eu>