

# Von allen Seiten betrachtet

Um dreidimensionale Körper von allen Seiten betrachten zu können, benötigen Sie den C 64, Simons Basic, einen Drucker und dieses Programm.

Das Programm soll einmal dreidimensionale Körper auf dem Bildschirm darstellen und zum anderen die hervorragende Grafikfähigkeit des C 64 demonstrieren.

Bei der Darstellungstast entschied ich mich für die normale Axonometrie. Sie ist einerseits leicht in eine für den C 64 verständliche Syntax zu packen und weist zum anderen einen räumlichen Effekt auf.

So entstand dann bald die erste Version von »Simons-Axo«. Allerdings war ich mit dieser Version noch nicht zufrieden. So tüftelte ich noch einige Routinen aus, die dem Programm erst den richtigen Schliff geben.

Nun fügten sich an die einfache Zeichenroutine noch weitere, die das Abspeichern der Körperdaten als sequentielle Datei auf Diskette, das Drehen des Körpers um die drei Koordinatenachsen (in beliebiger Variation), das Verschieben des Körpers, das Einzeichnen der Koordinatenachsen mit ihren Bezeichnungen und das Ausdrucken des Hires-Bildschirmes auf einen angeschlossenen Drucker ermöglichen.

Das Programm beginnt in der Zeile 10, wo der Bildschirm gelöscht und ebenso wie der Rahmen auf schwarz gesetzt wird. In den folgenden Zeilen gibt das Programm eine kurze Anleitung und wartet mit der Fortführung in Zeile 95, bis Sie eine beliebige Taste gedrückt haben.

In der Zeile 100 werden die Felder PT (Eckpunktkoordinaten, mit denen laufend gearbeitet wird) und PA (Eckpunktkoordinaten des Ausgangszustandes) mit 100,3 dimensioniert. Das bedeutet, daß Sie Körper mit bis zu 100 Eckpunkten eingeben können. Das Array ZP, ebenfalls mit 100 dimensioniert, gibt die Verbindungsvorschrift (Reihenfolge, in der die Punkte miteinander verbunden werden) an.

In Zeile 105–120 erfolgt eine Abfrage, ob die Körperdaten von der Diskette eingelesen werden sollen (Einleseroutine ab 1000) oder, ob Sie diese »von Hand« eingeben wollen. Sie können hier mit 'J' oder 'N' antworten, jede andere Antwort wird ignoriert.

Ab Zeile 125 steht die eigentliche Eingaberoutine, in der Sie mittels INPUT um die einzelnen Eckpunktkoordinaten gebeten werden. Diese werden den Feldern PT und PA zugeordnet. In

dieser Eingabeschleife fungierten PX, PY und PZ als Zwischenvariablen und A als Zähler. Die Eingabe können Sie jeweils mit »Ende« abschließen. Hierbei empfiehlt es sich, den oben genannten Term einzufügen, da das Programm ja drei Variablen verlangt und sonst nur noch nachfragt, bis es alle drei hat. Mit »£« können Sie die vorige Eckpunkteingabe noch einmal korrigieren (deswegen habe ich PX\$ anstelle von PX verwendet).

Von 200 bis 270 wird die Verbindungsvorschrift eingelesen. Hier geben Sie zuerst den Startpunkt ein und dann jeweils einen weiteren Eckpunkt. Die Nummern dieser Eckpunkte werden nacheinander in ZP(B) abgelegt. In der Zeichenroutine werden die Eckpunkte dann in dieser Reihenfolge durch Linien verbunden, wodurch der Körper gezeichnet wird. Auch hier können Sie mit der »£«-Taste Korrekturen ausführen. In dieser Schleife dient B als Zähler und A\$ als Zwischenspeicher. A1 enthält die Nummer des zuvor eingegebenen Eckpunktes (für die Korrektur notwendig).

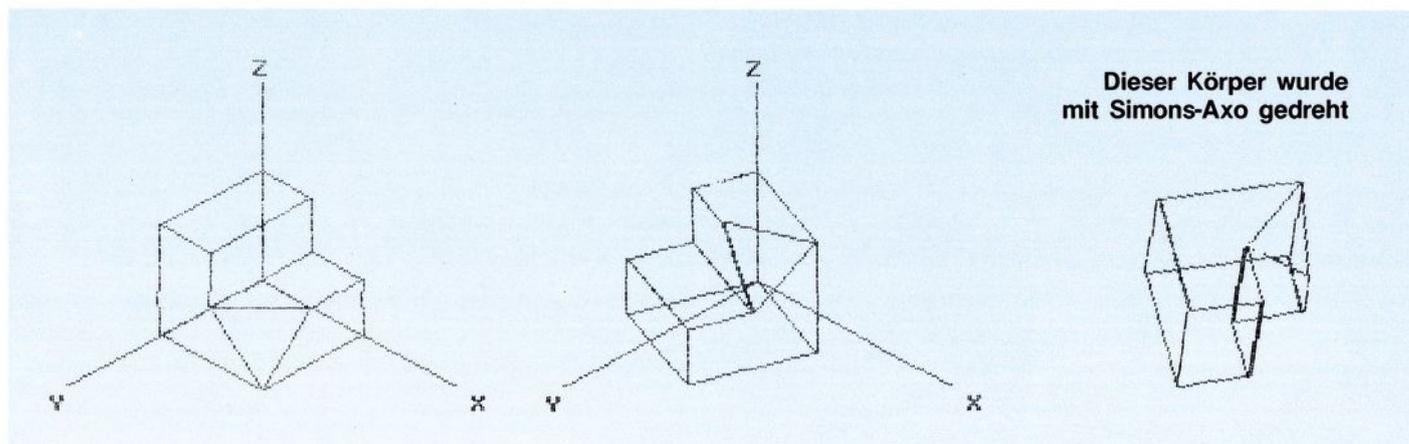
In 300 bis 360 können Sie das Koordinatenkreuz festlegen, indem Sie die Winkel zwischen der z- und y-Achse (AL) und z- und x-Achse (BT) eingeben. Das Programm ist für jeweils 120° voreingestellt, Sie können diese Vorschläge aber einfach überschreiben.

**ACHTUNG:** Sie wählen hier die Winkel zwischen den projizierten Koordinatenachsen. Die Winkel zwischen den realen Achsen im Raum sind natürlich immer je 90°!

Durch geeignete Wahl dieser Winkel können Sie den Sichtwinkel, unter dem Sie den dargestellten Körper betrachten, ändern. Sind beide Winkel 90°, so blicken Sie frontal von vorne auf den Körper, sind beide kleiner als 90°, so sehen Sie von unten her auf Körper und bei Winkeln über 90° von oben her.

Die nächsten Zeilen, von 400 bis 490 stellen die Zeichenroutine dar. Hier wird zunächst in Zeile 405 in den Hires-Modus umgeschaltet und die Zeichenfarbe grün bei schwarzem Hintergrund gewählt. In 410 bis 420 werden die Zeichenkoordinaten (zweidimensional) des Anfangspunktes, welcher durch ZP 0 gegeben ist, errechnet und in XA und YA abgelegt. In der nachfolgenden Schleife werden nach dem gleichen Schema die Endpunktkoordinaten für den LINE-Befehl des Simons Basic berechnet und die Linie gezeichnet. In Zeile 450 wird der Endpunkt nun zum Anfangspunkt für die nächste Linie definiert. Die Variable B dient hier als Zähler, während A als Index verwendet wird. Wenn alle Punkte miteinander verbunden sind, so wie es die Verbindungsvorschrift ZP(B) angibt, wird die Schleife beendet und das Programm prüft, ob das Flag für das Einzeichnen der Koordinatenachsen, KO auf 1 (zeichnen) gesetzt ist. Ist dies der Fall, so wird nach 1410 verzweigt.

In 480 und 485 wird in der linken oberen Ecke die Fertigmeldung ausgegeben und in die Endlosschleife in 490 gesprungen, die nach Drücken der F1-Taste beziehungsweise RETURN verlassen wird.





```

127 print "Bei x='ende' wird die eingabe
beendet."
128 print "Bei x='E' kann der vorige pun
kt nochein"
129 print "mal eingegeben werden."
130 print ("a"); : input px$,py,pz : px
=val(px$) : if px$="ende" then 200
135 if px$="E" then a=a-1 : goto 130
140 pt(a,1) = px
145 pt(a,2) = py
150 pt(a,3) = pz
155 pa(a,1) = pt(a,1)
160 pa(a,2) = pt(a,2)
165 pa(a,3) = pt(a,3)
170 a=a+1 : goto 130
200 rem
201 rem verbindungsordnung
202 rem
205 ep = a-1 : print "In welcher reih
enfolge sollen welche"
210 print "punkte verbunden werden ?"
215 print "ende' beendet die eingabe wi
eder."
220 print "E' laesst die korrektur der
vorigen "
225 print "eingabe zu."
230 input "Ausgangspunkt";a
235 zp(0)=a
240 b=1 : print " " : rem eingabeschleife
245 print "von (a) nach : " : input a$
: if a$="ende" then 270
250 if a$="E" then b=b-1 : a=a1 : goto
245
255 a1=a : a = val(a$) :
260 zp(b) = a
265 b=b+1 : goto 245
270 sp=b-1
300 rem
301 rem koordinatensystem festlegen
302 rem
305 print "Bestimmen sie nun das ac
hsenkreuz."
310 print "Geben sie die winkel zwisc
hen : "
315 print " "
320 print tab(19) " |"
325 print tab(19) " |"
330 print " z- und y-achse ";tab(19) "NM"
;tab(24) "z- und x-achse "
335 print tab(18) "N M"tab(24) " "lein.
"
345 print tab(17) "N M"
350 rem
355 input "alpha = 120";al:print
tab(25);:input "beta = 120";bt
360 al=al*%/180 : bt=bt*%/180
400 rem
401 rem zeichenroutine
402 rem
405 hires 5,0
410 a=zp(0)
415 xa = 160 - sin(al)*pt(a,2) + sin(bt)
*pt(a,1)
420 ya = 100 - pt(a,3) - cos(al)*pt(a,2)
- cos(bt)*pt(a,1)
425 for b = 1 to sp
430 a=zp(b)
435 xs=160-sin(al)*pt(a,2)+sin(bt)*pt(a,
1):if xs>319 then xs=319;ifxs<0thenxs=0
440 ys=100-pt(a,3)-cos(al)*pt(a,2)-cos(b

```

```

t)*pt(a,1):ifys>199 then ys=199
441 if ys<0 thenys=0
445 line xa,ya,xs,ys,1
450 xa=xs : ya=ys
455 next
460 if ko = 1 then 1410
480 d$="<fertig>"
485 text 5,5,d$,1,0,7
490 get a$ : if a$ <> chr$(133) and a$ <
> chr$(13) then 490
500 rem
501 rem hauptmenue
502 rem
505 nrm : print " "tab(10) "hauptmenue : "
510 print tab(10) "EEEEEEEEEEEEEE"
515 print " (1) = neues achsenkreuz
520 print " (2) = drehung der figur um
oo
525 print " (3) = ausgangszustand hers
tellen"
530 print " (4) = verschieben der figu
r"
535 print " (5) = koordinatenachsen ei
nzeichnen"
545 print " (6) = abspeichern der koer
perdaten"
550 print " (7) = hardcopy auf drucker
555 print " (8) = neuer start"
560 print " (9) = zur grafik zurueck"
580 print tab(10) " (0) = ende"
585 get a$ : if a$ <"0" or a$ >"9" or a$
="" then 585
590 on val(a$) goto 300,600,800,1300,140
0,1100,1500,100,1220
595 stop
600 rem
601 rem drehung der figur um oo
610 print " "drehung der figur um oo
: "print "EEEEEEEEEEEEEEEEEEEEEEEEEEEE"
615 print "bitte waehlen sie aus : "
620 print " (1) = drehung um die z-ac
hse.
625 print " (2) = drehung um die y-ach
se.
630 print " (3) = drehung um die x-ach
se."
635 get a :if a < 1 or a >3 then 635
636 z=5+2*a : inv z,3,29,1
640 print tab(7) "bitte den drehwinkel
: " :input de : de=de*%/180
645 on a goto 650,700,750
650 rem
651 rem um die z-achse
655 for x=0 to ep
660 eo=atn(pt(x,1)/(pt(x,2)-1e-32))-*(p
t(x,2)<=0)-2***(pt(x,1)<0 andpt(x,2)>0)
665 eo = eo+de : if eo >= 2** then eo =
eo-2**
670 r=sqr(pt(x,1)^2+pt(x,2)^2)
675 pt(x,1)=sin(eo)*r
680 pt(x,2)=cos(eo)*r
685 next
690 goto 790 : rem return
700 rem
701 rem um die y-achse
705 for x=0 to ep
710 eo=atn(pt(x,3)/(pt(x,1)-1e-32))-*(p
t(x,1)<=0)-2***(pt(x,3)<0andpt(x,1)>0)
715 eo=eo+de : if eo >= 2** then eo=eo-2
**

```

Listing »Simons Axo« (Fortsetzung)

```

720 r = sqr(pt(x,1)↑2+pt(x,3)↑2)
725 pt(x,1)=cos(eo)*r
730 pt(x,3)=sin(eo)*r
735 next
740 goto 790 : rem return
750 rem
751 rem um die x-achse
755 for x=0 to ep
760 eo=atn(pt(x,2)/(pt(x,3)-1e-32))-π*(p
t(x,3)<=0)-2*π*(pt(x,2)<0andpt(x,3)>0)
765 eo=eo+de : if eo>= 2*π then eo = eo-
2*π
766 r = sqr (pt(x,2)↑2+pt(x,3)↑2)
770 pt(x,3)=cos(eo)*r
775 pt(x,2)=sin(eo)*r
780 next
790 cset 2 : goto 400 : rem return
800 rem ausganagszustand
801 rem
810 for x = 0 to ep
820 for y = 1 to 3
830 pt(x,y)=pa(x,y)
840 nexty,x
850 goto 400
999 rem
1000 rem daten vom der disk holen
1001 rem
1010 open 1,8,15 : rem fehlerkanal
1015 print"Bitte dateinamen :"; : inpu
t ns$ : n$=ns$
1020 ns$=ns$+",s,r"
1025 open 2,8,2,ns$
1030 gosub 1250
1032 if f1 = 62 then close 2 : goto 1015
1035 print"Einlesen der daten von :";n
$
1040 input#2,ep
1045 input#2,sp
1050 for x = 0 to ep
1055 for y = 1 to 3
1060 input#2,pt(x,y) : pa(x,y)=pt(x,y)
1065 next y
1070 gosub 1250
1075 next x
1080 for x = 0 to sp
1085 input#2,zp(x)
1090 next x
1095 gosub 1250
1097 close 2 : close 1 : goto 300
1100 nrm : rem koerperdaten auf disk
1101 rem
1110 print "#####koerperdaten auf
disk:"
1120 print"#####
"
1130 print"Bitte dateinamen :"; : inp
ut ns$
1140 open 1,8,15 : rem fehlerkanal
1150 ns$=ns$+",s,w"
1160 open 2,8,2,ns$
1162 print#2,ep : print#2,sp
1165 for x = 0 to ep : rem punkte
1170 for y = 1 to 3
1180 print#2,pt(x,y)
1185 next y
1190 gosub 1250
1195 next x
1200 for x = 0 to sp : rem verbindungsvo
rschrift
1205 print#2,zp(x)

```

```

1210 next
1215 gosub 1250
1220 close 2 : close 1 : cset 2 : goto 4
90
1250 rem
1251 rem fehlerkanal
1252 rem
1255 input#1,f1,f1$,f2,f3 : if f1 = 0 th
en return : rem kein fehler
1260 print "#####fehler auf der disk !"
1265 print"#####f1,f1$" "f3" "f4
1270 print"#####bitte mit <return> bestaeti
gen,oder mit"
1275 print"#####Kspace> das programm abbre
chen !"
1280 getan$ : if an$ = chr$(13) then ret
urn
1285 if an$ <> chr$(32) then 1280
1290 close 1 : stop
1300 rem
1301 rem verschieben der figur
1302 rem
1305 print"#####verschieben der figur
:";print"#####
"
1310 print"#####bitte um den verschiebe
vektor in der"
1315 print"form 'x,y,z' !"
1320 input"#####vektor :";px,py,pz
1322 print"#####bitte etwas geduld ."
1325 for a=0 to ep
1335 pt(a,1) = pt(a,1) + px
1340 pt(a,2) = pt(a,2) + py
1345 pt(a,3) = pt(a,3) + pz
1350 next a
1360 goto400
1400 rem
1401 rem koordinatenachsen
1402 rem
1405 if ko=1 then ty=0 : ko=0 : goto 141
5
1410 ko=1 : ty = 1
1415 xs=160-sin(al)*95 : x1=160-sin(al)*
105
1420 ys=100-cos(al)*95 : y1=100-cos(al)*
105
1425 char x1,y1,25,ty,1
1430 line 160,100,xs,ys,ty : rem y-achse
1435 xs=160+sin(bt)*95 : x1=160+sin(bt)*
102
1440 ys=100-cos(bt)*95 : y1=100-cos(bt)*
105
1445 line 160,100,xs,ys,ty : rem x-achse
1450 char x1,y1,24,ty,1
1455 line 160,100,160,10,ty : rem z=achs
e
1460 char 155,0,26,ty,1
1470 cset 2 : goto 480
1500 rem
1501 rem hardcopy
1502 rem
1505 text 5,5,d$,0,0,7
1510 d$="<hardcopy>"
1515 text 5,5,d$,1,0,8
1520 cset 2
1525 copy
1530 text 5,5,d$,0,0,8
1535 goto 480

```

ready.

Listing »Simons Axo« (Schluß)