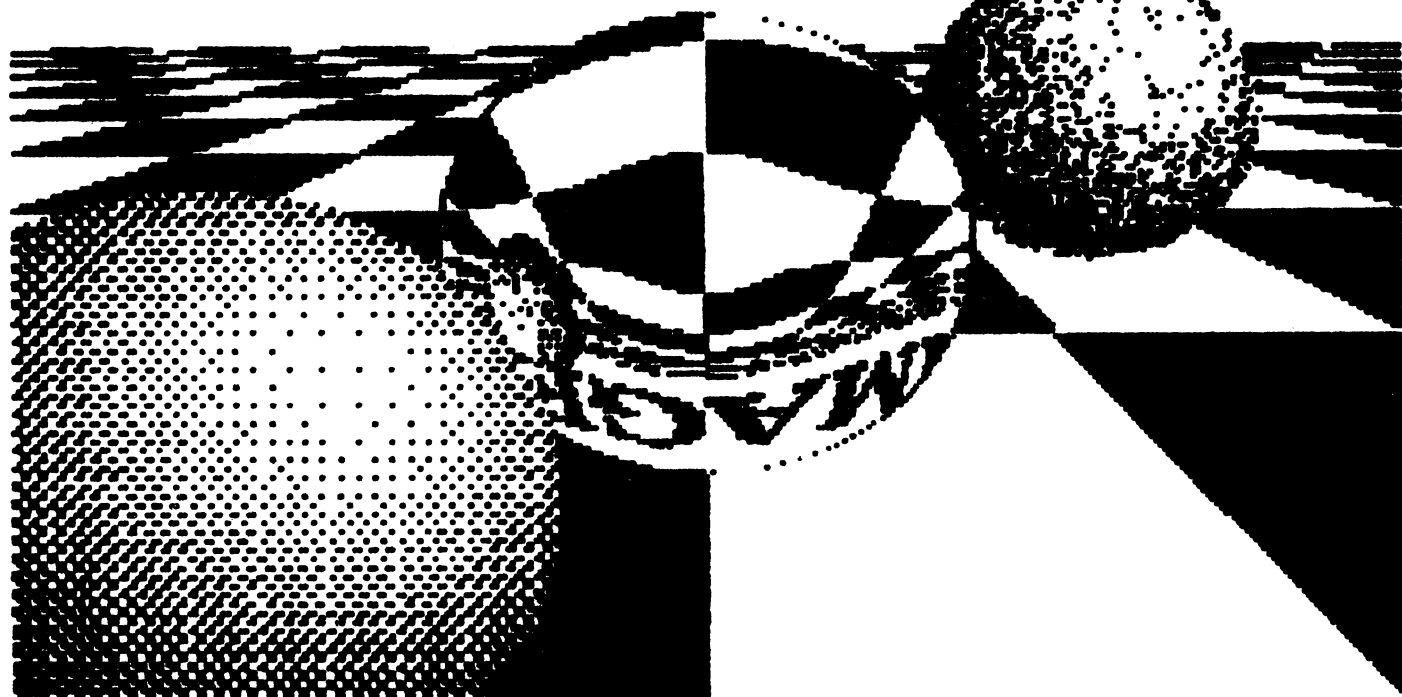


IMAGES



Räumliche Grafik auf dem C 64

Gerade auf dem Gebiet von Film- und Fernsehproduktionen werden Computergrafiken immer öfter eingesetzt, da es hiermit möglich ist, Dinge zu zeigen, die mit herkömmlichen Methoden nicht gefilmt werden konnten.

Ob nun Raumschiffe gezeigt werden, die eigentlich unmögliche Bewegungen durchführen oder der »Anflug« eines Logos (wie zum Beispiel die »1« bei der ARD), die Realisierung vereinfacht sich durch den Einsatz von Grafik-Computern enorm. Diese müssen jedoch — bedingt durch die anfallenden Datenmengen — enorm schnell sein, so daß zur Zeit nur »Supercomputer« Echtzeitsimulation erlauben.

Wenn man sich allerdings etwas mehr Zeit läßt, dann reicht auch ein Homecomputer wie der C 64 aus, um solche Grafiken (mit bestimmten Einschränkungen) zu erzeugen.

Das hier vorgestellte Grafikpaket geht über das übliche Punkte setzen und Linien ziehen weit hinaus. Es erlaubt die Erzeugung von plastischen Bildern ohne größeren Aufwand, wie zum Beispiel in Bild 1 und 2 dargestellt.

Das eigentliche Grafikpaket nimmt nur 3 KByte (!) des Speichers in Anspruch und steht im Bereich ab \$C000. Damit

bleibt der Basic-Speicher für die eigentlichen Programme frei. Auch das DOS 5.1 von Commodore kann parallel zu den Grafikroutinen verwendet werden. Aufgrund der Kompaktheit dieses Grafikpakets mußten allerdings bestimmte Dinge, wie zum Beispiel die Abfrage auf legale Koordinaten, in den Hintergrund treten; die Software fängt also solche Fehler nicht ab. Das macht das Schreiben eines eigenen Programms zwar etwas komplizierter, hat man aber erst einmal alle Fehler beseitigt, so würde eine Fehlerabfrage den Ablauf nur verlangsamen.

Beginnen wir mit den Grundlagen zur Erzeugung der Grafik. Da stellt sich zunächst einmal die Frage, wie man die verschiedenen Helligkeitswerte und Schattierungen eines darzustellenden Objektes berechnet. Danach müssen diese verschiedenen Grauwerte in einzelne Punkte umgesetzt werden, denn auf dem HiRes-Bildschirm können diese ja nur gesetzt und gelöscht werden. Dazu muß die gesamte Software selbst geschrieben werden, da der C 64 diese Art der Grafik vom Betriebssystem her nicht unterstützt.

Schnelle Rechenroutinen

Das Maschinenprogramm, das für den Aufbau der Grafik zuständig ist, besteht aus mehreren Unterprogrammen. Beginnen wollen wir mit einem Unterprogrammpaket, das für schnelles Rechnen notwendig ist.

Integer Arithmetik (Listing 1)

Diese Unterrountinen stellen schnelle Algorithmen für Multiplikation, Division, Wurzelziehen und Zufallszahlen zur Verfügung. Diese sind zwar schon im normalen Basic-Interpreter vorhanden, für unsere Anwendung sind sie jedoch aus Geschwindigkeitsgründen ungeeignet. Die Basic-Routinen sind nämlich nicht auf Geschwindigkeit, sondern auf Kürze optimiert worden, wie zum Beispiel die Funktion SQR. Die

Wurzelberechnung in Basic teilt den Logarithmus durch 2 und potenziert das Ergebnis anschließend wieder. Das hat den Vorteil, daß der Interpreter auf bereits vorhandene Unterprogramme zugreifen kann, aber den Nachteil, daß diese Berechnungsweise relativ langsam ist. Wenn man für alle 64000 Bildschirmpunkte die Quadratwurzel mit den vorhandenen Routinen berechnen würde, ergäbe sich eine Zeichenzeit von mehr als einer Stunde (und dies trotz Maschinenspracheprogramm)! Man benötigt diese Berechnung zwar nicht für jeden Punkt, aber gerade bei der Berechnung der Schattierungswerte spielt die Quadratwurzel eine große Rolle. Eine Möglichkeit, die Wurzelberechnung zu beschleunigen bietet die Newton-Methode, mit der man Fließkommawurzeln in

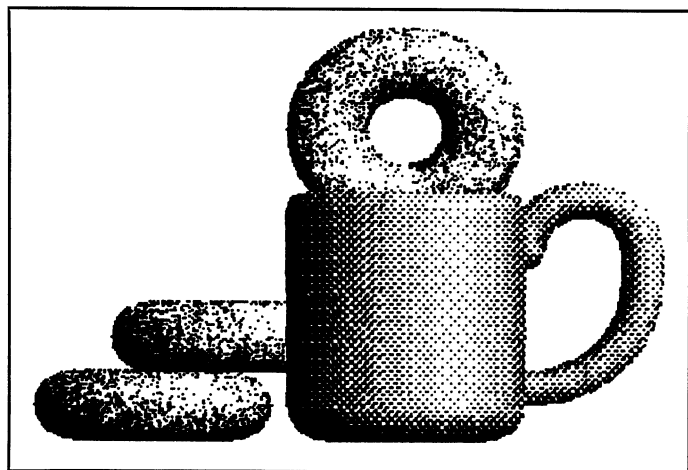


Bild 1. Eindrucksvolle räumliche Wirkung durch Kombination verschiedener Shapes

einem Bruchteil der sonst benötigten Zeit berechnen kann. Da wir, um die einzelnen Grafikpunkte zu adressieren, nur ganze Zahlen benötigen, beschleunigt sich die Abarbeitungszeit nochmals. Auch die Routine für die Erzeugung von Zufallszahlen, die später bei der »Random«-Schattierung erforderlich ist, wurde neu geschrieben.

Grafik Utilities (Listing 2)

Die nächste Unterprogrammsammlung beinhaltet Basisroutinen für die Verwaltung der Bitmap wie zum Beispiel löschen, Farbe setzen und Punkte plotten. Diese Unterroutinen sind als einzige Commodore 64-spezifisch, so daß man — möchte man das Programm auf andere 6502-Computer übertragen — nur diese an das jeweilige Gerät anpassen muß.

Ferner wurden noch zwei Schattierungsroutinen »SHADE« und »RSHADE« implementiert. »SHADE« arbeitet im Prinzip ganz einfach: Man definiert eine 8 x 8 Graustufenmatrix, in der Werte von 0 bis 63 enthalten sind (Tabelle 1). Mit Hilfe dieser Daten wird nun ein — ebenfalls 8 x 8 Pixel großes — Feld erstellt, daß dann an einer bestimmten Stelle in den HiRes-Bildschirm kopiert wird. Um eine gleichmäßige Grauschattierung zu erreichen, müssen die Pixels auch möglichst einen gleichmäßigen Abstand voneinander haben. Wenn das Programm nun einen Grauwert in die Grafik setzen will, so übergibt es einfach einen Wert zwischen 0 und 63 an diese Unteroutine. Nun werden innerhalb dieser Matrix die einzelnen Punkte der Reihe nach gesetzt, also zuerst der Punkt mit der Nummer 0 (oben links), dann 1 (4. von links und 4. von oben) und so weiter. Also, je größer die Zahl ist, desto mehr Punkte werden gesetzt (und um so dunkler wird das Ganze). Durch diese festgelegte Reihenfolge wird sichergestellt, daß die Schattierung gleichmäßig ist. Die RSHADE-Unteroutine hat genau die gleiche Aufgabe, sorgt aber für etwas ungleichmäßigere — durch den Zufallsgenerator gesteuerte — Grauwerte.

Ein weiteres zu lösendes Problem ist das etwas merkwürdige rechteckige Bildschirmformat des C 64. Die Auflösung von 320 x 200 Punkten entspricht einem Teilungsverhältnis von 8:5, wodurch es bei einem normalen Bildschirm — mit einem Verhältnis von 4:3 — zu einer horizontalen Verzerrung kommen kann (eine Kugel würde dann etwa wie ein plattgedrückter Ball aussehen). Die Unteroutine »SCALE« (in Listing 2) sorgt für diese darstellungs- und maßstabsgerechte Aufbereitung der Koordinaten.

Linien und Facettenzeichnen

Die Routinen in Listing 3 komplettieren die elementaren Grafik-Funktionen für das Ziehen von Linien und das Zeich-

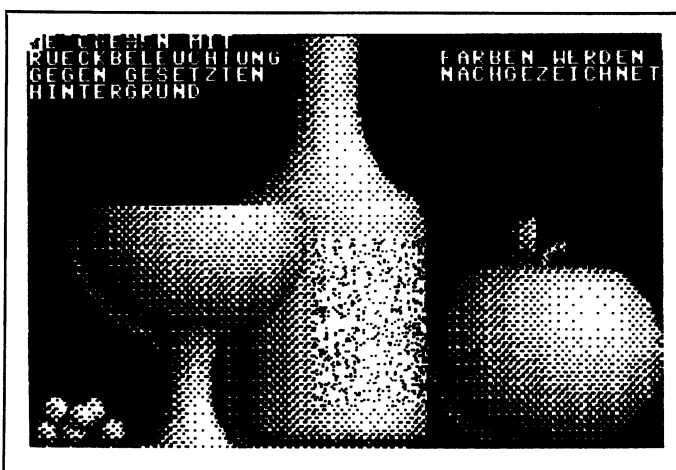


Bild 2. Verblüffende Effekte lassen sich erzielen. Beachten Sie den Lichteinfall.

nen von schattierten Polygonnetzen. Ein solches räumliches Polygonnetz ist in Bild 3 zu sehen; es besteht nur aus aneinander gesetzten und entsprechend schattierten Dreiecken. Diese Figuren können einfacher und schneller gezeichnet werden, da die Schattierung auf ebenen Flächen einfacher darzustellen ist als auf gekrümmten. Neben der üblichen Darstellungsweise (in der geschlossenen, schattierten Form) ist auch die Zeichnung von Grafiken ohne verdeckte Linien möglich.

Mit diesen Routinen gerüstet, können wir uns nun auf das eigentliche Shape-Unterprogramm konzentrieren, mit dem

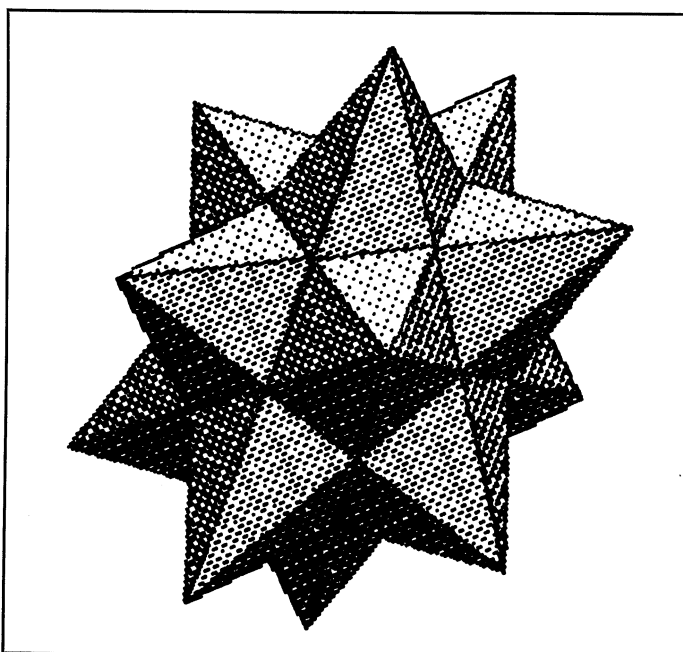


Bild 3. Ein weiteres Beispiel für die 3D-Wirkung

es möglich ist, gekrümmte, schattierte Figuren in kurzer Zeit zu erstellen.

Hier geht es wiederum als erstes um die Berechnung der verschiedenen Oberflächenreflexe, Brechungen, Streuungen und Absorptionen des Lichtes auf einer Oberfläche. Um diese, teilweise sehr komplizierten Berechnungen, zu vereinfachen, geht man einen anderen Weg: Man beschränkt sich auf einige einfache symmetrische Figuren (Bild 4), aus denen alle zu zeichnenden Objekte zusammengesetzt werden. Weiterhin werden die Figuren nur in der Frontsicht, also von vorn, dargestellt; Drehungen sind nicht möglich. Durch diese Einschränkungen braucht man bei der Programmierung nicht auf komplizierte Vektorberechnungen zurückzugreifen.

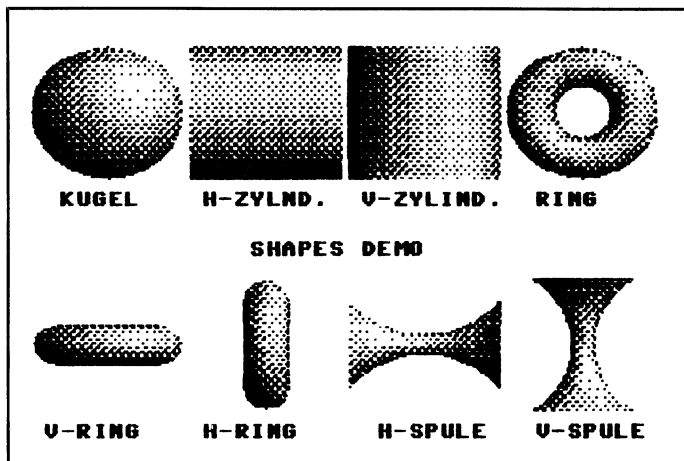


Bild 4. Die zur Verfügung stehenden Shapes, aus denen alle Figuren zusammengesetzt werden

Für die Berechnung des Lichteinfalls stelle man sich einfach eine Lampe vor, die an der rechten Schulter des Beobachters angebracht ist und das Objekt von dieser Position anleuchtet. Dementsprechend errechnet das Programm die Helligkeitswerte für jeden Punkt des Objektes. Ein Flag (Adresse 898) dient dazu, den Standort der »Lampe« (0 = Frontbeleuchtung, 1 = Beleuchtung von hinten) festzulegen.

Zeichenroutinen für Shapes

Listing 4 beinhaltet eine Sammlung von Routinen, die die acht Grundshapes (siehe Bild 4) zeichnen. Da dieses Unterprogramm auf die — an den C 64 angepaßten — Lower-Level-Routinen der Grafik-Utilities (Listing 2) zugreift, ist es selbst nicht vom jeweiligen Computer abhängig. Zusammen mit dem, in Listing 5 abgebildeten Programm, wird der Kontakt zum Basic hergestellt. Die Übergabe der Parameter geschieht mittels »SYS«-Befehl:

SYS <KOMMANDO>, <PARAM. 1>, <PARAM. 2>, ..., wobei <KOMMANDO> die Adresse des jeweiligen Grafikbefehls ist (Tabelle 2 zeigt die komplette Befehlsübersicht dieses Grafikprogramms). Über <PARAMETER 1> und <PARAMETER 2> müssen die Koordinaten des Shapes angegeben werden, wobei zu beachten ist, daß der Ursprung (dort wo die X- und Y-Koordinate null ist) in der linken unteren

0	8	53	61	2	10	55	63
16	24	37	45	18	26	39	47
49	57	4	12	51	59	6	14
33	41	20	28	35	43	22	30
3	11	54	62	1	9	52	60
19	27	38	46	17	25	36	44
50	58	7	15	48	56	5	13
34	42	23	31	32	40	21	29

Tabelle 1. Matrix der Grauwerte

Bildschirmcke liegt. Die einzusetzenden Werte gehen in der Horizontalen von 0 bis 255, in der Vertikalen von 0 bis 239.

Alle Shapes, mit Ausnahme der Kugel, können durch die Eingabe von insgesamt vier Parametern in ihrer Position und Form beeinflusst werden. Die ersten zwei bestimmen, wie oben erläutert, die Koordinaten des Figurmittelpunktes. Mit dem dritten und vierten Parameter kann man die Verzerrung in X- und Y-Richtung einstellen. Doch hier ist Vorsicht geboten, denn der X-Wert sollte hierbei immer größer als der Y-Wert sein, sonst zeichnet das Programm unerwünschte Figuren.

Eine weitere Besonderheit sind die Shape-Fenster. Damit ist es möglich, auch einzelne Teile eines Shapes auf dem Bildschirm zu bringen. Diese Funktion wurde zum Beispiel bei der Erstellung der Grafik in Bild 1 verwendet. Der Henkel die-

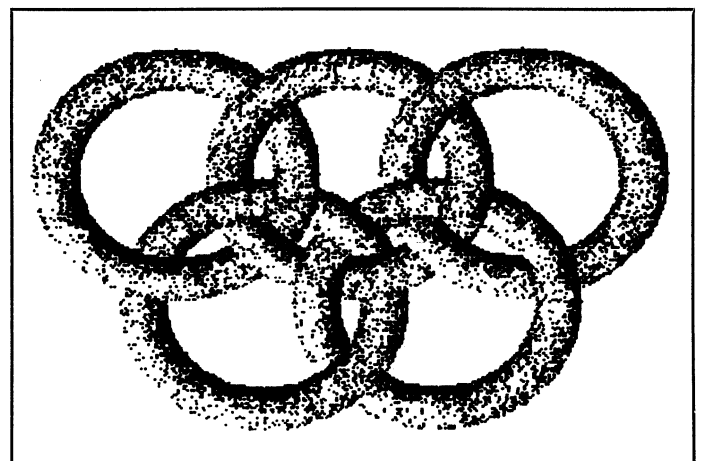


Bild 5. Aus einfachen Figuren lassen sich verblüffend »lebendige« Bilder zusammensetzen

Grafikspeicher: 40960-48959

Farbspeicher: 33792-34791

Shape-Fenster

893: Linker Rand

894: Rechter Rand

895: Unterer Rand

896: Oberer Rand

Style Flags

838: Schattierungsart (0 = zufällig, 1 = Halbton)

839: Skalierungsflag (0 = normal 1:1, 1 = skaliert 4:3)

868: Eck Flag bei Polygonen (0 = Normal, 1 = zeichnet Linien an den Ecken)

871: Eckenmodus (0 = zeichnet Linie, 1 = löscht Rand)

898: Beleuchtungsflag (0 = von vorne, 1 = von hinten)

Funktionsadressen

49378: Auf den Grafikmodus umschalten

49411: Zurück zum Textbildschirm

51979: Bitmap löschen

52001: Bitmap färben

52023: PLOT (Parameter X,Y) setzt einen Punkt

52026: UNPLOT löscht einen Punkt mit den Koordinaten X und Y

52049: LINE (Parameter X1,Y1,X2,Y2) zieht Linie von / bis

52052: FACET X1,Y1,X2,Y2,X3,Y3,SA zeichnet dreieckige Facetten mit den Koordinaten X1,Y1, X2,... und schattiert sie (SA: 0 = schwarz, 64 = weiß)

Shapes

52119: Kugel

52141: Ring

52150: vertikaler Zylinder

52153: horizontaler Zylinder

52186: vertikaler Ring

52189: horizontaler Ring

52203: vertikale Rolle

52206: horizontale Rolle

Tabelle 2. Befehlsübersicht

ser Kaffeetasse ist aus drei verschiedenen Shapes zusammengesetzt, die mit Hilfe des Fensters in ihrer Ausdehnung eingeschränkt wurden. Diesen Begrenzungsfenstern sind die Adressen 893 bis 896 zugeordnet, wobei sich die POKE-Werte auf den Mittelpunkt der Figur beziehen.

Die Kommandos, um die Bitmap zu löschen oder mit Punkten zu füllen, benötigen auch alle nur einen Parameter. So sieht zum Beispiel das Kommando zum Initialisieren des Farbspeichers so aus:

```
POKE 52001,16*DC+BC
```

wobei DC die Farbe der zu setzenden Punkte und BC die Hintergrundfarbe ist. Auch Texte können nach Wunsch auf dem Grafikbildschirm ausgegeben werden. Die dafür zuständigen Routinen sind in Listing 2 enthalten.

Die in Listing 10 und 11 abgedruckten Demonstrationsprogramme sollen helfen, die Arbeitsweise und die Kombinationsmöglichkeiten der einzelnen Shapes aufzuzeigen. Auch alle hier gezeigten Bilder stammen von diesem Programm. Die Bilder stehen im Speicher übrigens »unter« dem Basic-ROM im Bereich zwischen \$A000 und \$BFFF.

Um den Platz für das DOS 5.1 freizuhalten, wurde das Grafikpaket in zwei Teile aufgespalten. Der eine Teil mit dem Namen GRAPHICS (Listing 8) befindet sich im Bereich zwi-

schen \$C000 und \$CBFC. Der zweite Teil, auf den im folgenden noch kurz eingegangen werden soll, muß nach Adresse \$CF59 geladen werden. Dieser Programmteil enthält eine schnelle Sortieroutine (Listing 6) und ermöglicht die Darstellung von Text auf dem Grafikbildschirm (Listing 7). Die beiden Unterprogramme sind in dem MSE-Listing 9 zusammengefaßt worden und müssen gemeinsam mit »GRAPHICS« in den Speicher geladen werden.

Hinweise zum Abtippen

Sie müssen als erstes die MSE-Listings 8 und 9 eingeben und anschließend natürlich auch abspeichern. Die Listings 1 bis 7 dienen nur zur Dokumentation und brauchen nicht eingegeben werden. Nachdem Sie auch die Demos abgetippt und auf Diskette oder Band gespeichert haben, müssen die einzelnen Programmteile nacheinander in den Speicher geladen werden. Begonnen wird mit GRAPHICS, das mit LOAD-»GRAPHICS«, 8,1 in den Speicher gelesen werden muß. Nachdem man anschließend NEW eingegeben hat, muß das zweite Maschinenprogramm, wie oben, geladen werden (auch hier das NEW nicht vergessen). Anschließend kann wahlweise eines der beiden Demonstrationsprogramme (Listing 10 oder 11) geladen und ganz normal gestartet werden.

(Richard Rylander/Christoph Sauer/ev)

```
; INTEGER ARITHMETIC ROUTINES
;
; RICHARD L. RYLANDER 8/12/84
;
; REVISED 10/29/84 TO ADD FULL DOUBLE
; PRECISION ARGUMENTS IN DIVIDE ROUTINE
;
;*****
;
; USE PAGE ZERO LOCATIONS WHERE POSSIBLE FOR
; ITERATIVE PROCEDURE WORK SPACE
;
MLPCND = $AC ; MULTIPLICAND
MLPLER = $AD ; MULTIPLIER
PROD = $AE ; PRODUCT
;
DNDND = $FD ; DIVIDEND/QUOTIENT
DVSOR = $FB ; DIVISOR
RMNDR = $B4 ; REMAINDER
;
RADCND = $AC ; RADICAND
ROOT = $033C ; SQUARE ROOT
;
TEMP = $FB
;
; SET UP SEED VALUES FOR PSEUDO-RANDOM NUMBERS
;
RNDM .BYTE $FF,$55
RTEMP .BYTE $00,$00
;
;*****
;
; MULTIPLY SINGLE PRECISION MULTIPLICAND
; BY SINGLE PRECISION MULTIPLIER GIVING
; DOUBLE PRECISION PRODUCT (ENTER AT "MULT")
;
; SPECIAL CASE: ENTER AT "SQUARE" TO FIND
; SQUARE OF SIGNED 8-BIT NUMBER
;
SQUARE LDA MLPCND ; ENTRY TO SQUARE
BPL POSITV ; USE ABSOLUTE VALUE
SEC ; NEGATE IF NEEDED
LDA #$00
SBC MLPCND
STA MLPCND
STA MLPLER
LDA #$00 ; ENTRY TO MULTIPLY
LDX #$08
LSR MLPLER
BCC NOADD
CLC
ADC MLPCND
ROR A
ROR PROD
DEX
BNE MLOOP
STA PROD+1
RTS
;
;*****
;
; DIVIDE DOUBLE PRECISION DIVIDEND
; BY DOUBLE PRECISION DIVISOR GIVING
; DOUBLE PRECISION QUOTIENT
;
; DIVIDEND IS REPLACED BY QUOTIENT
; IN THE PROCESS
;
; QUOTIENT IS ROUNDED TO NEAREST INTEGER
;
DIVIDE LDA #$00
STA RMNDR
STA RMNDR+1
LDX #$10
ROR DNDND
ROR DNDND+1
ROR RMNDR
ROR RMNDR+1
SEC
LDA RMNDR
SBC DVSOR
TAY
LDA RMNDR+1
SBC DVSOR+1
BCC DECCNT
STY RMNDR
STA RMNDR+1
DEX
BNE DLOOP
```

```
ROL DNDND ; CHECK IF REMAINDER
ROL DNDND+1 ; IS >= 1/2 OF DIVIDEND
ASL RMNDR ; FOR ROUNDING
ROL RMNDR+1
BCS ROUND
SEC
LDA DVSOR
SBC RMNDR
LDA DVSOR+1
SBC RMNDR+1
BCS NOCHNG
ROUND INC DNDND
BNE NOCHNG
INC DNDND+1
NOCHNG RTS
;
;*****
;
; TAKE INTEGER SQUARE ROOT OF A
; DOUBLE PRECISION RADICAND GIVING
; SINGLE PRECISION ROOT ( <= REAL ROOT )
;
SQRT LDX #$08
LDA #$00
STA ROOT
STA ROOT+1
STA TEMP
STA TEMP+1
SQRT1 ASL ROOT
ROL ROOT+1
INC ROOT ; ASSUME CURRENT LSB OF
BNE NEXT1 ; ROOT WILL BE 1
INC ROOT+1
NEXT1 ASL RADCND ; SHIFT RADICAND LEFT
ROL RADCND+1 ; TWICE INTO TEMP
ROL TEMP
ROL TEMP+1
ASL RADCND
ROL RADCND+1
ROL TEMP
ROL TEMP+1
SEC ; SUBTRACT ROOT ESTIMATE
SBC ROOT ; FROM TEMP
TAY
LDA TEMP+1
SBC ROOT+1
BCC RESTOR
STA TEMP+1 ; SUBTRACTION OK
STY TEMP
INC ROOT
BNE NEXT2
INC ROOT+1
NEXT2 DEX
BNE SQRT1
JMP FINI
RESTOR SEC ; IGNORE SUBTRACTION
LDA ROOT ; AND RESET LSB OF ROOT
SBC #$01
STA ROOT
BCS NEXT3
DEC ROOT+1
NEXT3 DEX
BNE SQRT1
FINI ROR ROOT+1 ; FINAL /2 TO NORMALIZE
ROR ROOT
RTS
;
;*****
;
; GENERATE PSEUDO-RANDOM BYTES
; EXIT WITH P-R BYTE IN ACCUM.
;
RANDOM LDA RNDM
STA RTEMP
EOR RNDM+1
ROL RTEMP+1 ; RTEMP+1 PRESERVES
ROR A ; CARRY BIT FOR CYCLING
ROR RTEMP+1 ; RANDOM NUMBERS
STA RNDM
LDA RTEMP
STA RNDM+1
RTS
;
.END
```

Listing 1. Assembler-Quelltext der schnellen Integer-Arithmetik

```
; GRAPHICS UTILITIES
;
; RICHARD L. RYLANDER 11/4/84
;
; LOAD ARITHMETIC UTILITIES FIRST
;
RAM=$033E
ORIGIN=$00E2
;
MLPCND=$AC ; MULTIPLICAND (S)
MLPLER=$AD ; MULTIPLIER (S)
PROD=$AE ; PRODUCT (D)
MULT=$C011 ; CALL FOR MULTIPLY
;
RNDM=$C000 ; RANDOM NUMBER
RANDOM=$C008 ; CALL FOR RANDOM
; NOTE - A CALL TO 'RANDOM' LEAVES A RANDOM BYTE
; IN THE ACCUMULATOR
;
;*****
;
PLTFLG = $*+1 ; PLOT/UNPLOT FLAG
XPLT = $*+2 ; ABSOLUTE PLOT COORD
YPLT = $*+3 ; ABSOLUTE PLOT COORD
VIC1 = $*+4 ; REGISTER STORAGE
VIC2 = $*+5 ; REGISTER STORAGE
VALUE = $*+6 ; FINAL NORMALIZED SHADE VALUE
HTORRN = $*+7 ; SHADE FLAG, 1=HALFTONE
NOSCAL = $*+8 ; SCALE FLAG, 1=NO SCALE
TEMP = $*+9 ; TEMPORARY STORAGE
;
;*****
;
; TURN ON BIT MAP GRAPHICS MODE,
; SAVING REGISTER VALUES FOR
; RETURN TO TEXT MODE LATER.
;
GRFON LDA $D011
ORA $*20
STA $D011
LDA $D000
STA VIC1
AND $*FC
ORA $*01
STA $D000
LDA $D018
STA VIC2
LDA $*19
STA $D018
RTS
;
;*****
;
; RETURN TO TEXT SCREEN
;
GRFOFF LDA $D011
AND $*DF
STA $D011
LDA VIC1
STA $D000
LDA VIC2
STA $D018
RTS
;
;*****
;
; FILL COLOR MAP FOR BLACK DOTS ON WHITE
;
COLOR LDA $*01 ; POKE NEW COLORS HERE
LDX #0
COL1 STA $B400,X
STA $B500,X
STA $B600,X
STA $B700,X
DEX
BNE COL1
RTS
;
;*****
;
; CLEAR HI-RES GRAPHICS SCREEN
;
CLEAR LDA $*A0
STA $*FC
LDY #0
STY $*FB
LDA #0 ; CLEAR BYTE
```

Listing 2. Assembler-Quelltext der Grafik-Utilities

```

CLRLP   LDX #20
        STA ($FB),Y
        INY
        BNE CLRLP
        INC #FC
        DEX
        BNE CLRLP
        RTS
;
;*****
; PLOT AND UNPLOT POINTS ON HI-RES GRAPHICS
; SCREEN. ABSOLUTE X AND Y SCREEN COORDINATES
; ARE POKED INTO XPLT, XPLT+1, AND YPLT
;
PLOT    LDA #0
        .BYTE $2C
UNPLOT  LDA #80
        STA PLTFLG
        LDA #01 ; BASIC ROM OUT
        AND #FE
        STA #01
        SEC
        LDA #C7 ; INVERT Y COORDINATE TO
        SBC YPLT ; PUT ORIGIN IN LOWER LEFT
        TAX ; CORNER OF SCREEN
        LSR A ; (199.-YPLT)
        LSR A
        LSR A
        TAX
        LDA TABLE1,Y
        STA #FB
        LDA TABLE2,Y
        STA #FC
        TAX
        AND #07
        CLC
        ADC #FB
        STA #FB
        LDA XPLT
        AND #FB
        ADC #FB
        STA #FB
        LDA XPLT+1
        ADC #FC
        STA #FC
        LDA #A0
        ADC #FC
        STA #FC
        LDA XPLT
        AND #07
        EOR #07
        TAX
        LDA #01
        DEX
        BPL PLOT2
        ASL A
        BNE PLOTLP
        PLOT2 LDY #0
        BIT PLTFLG
        BPL NOPLOT
        EOR #FF
        AND ($FB),Y
        .BYTE $2C
        NOPLOT ORA ($FB),Y
        STA ($FB),Y
        LDA #01 ; BASIC ROM RESTORED
        ORA #01
        STA #01
        RTS
;
TABLE1 .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
        .BYTE $00,$40,$80,$C0
;
TABLE2 .BYTE $00,$01,$02,$03
        .BYTE $05,$06,$07,$08
        .BYTE $0A,$0B,$0C,$0D
        .BYTE $0F,$10,$11,$12
        .BYTE $14,$15,$16,$17
        .BYTE $19,$1A,$1B,$1C,$1E
;
;*****
; SHADING BY HYBRID DITHER/DOT-GROWTH
;
SHADE   LDA XPLT ; USE BITS -----
        AND #07 ; OF 'X' SCREEN COORD
        STA TEMP
        LDA YPLT ; AND BITS -----
        AND #07 ; OF 'Y' SCREEN COORD
        ASL A ; SHIFTED INTO ----
        ASL A ; POSITION TO DETERMINE
        ASL A ; 6-BIT OFFSET IN
        ORA TEMP ; THRESHOLD TABLE
        TAX
        LDA THRESH,X ; SCREEN-POSITION-WEIGHTS
        CMP VALUE ; THRESHOLD VALUE
        BPL GREATR
        JMP UNPLOT
GREATR  JMP PLOT
;
;*****
; SHADING BY RANDOM HALFTONE
;
RSHADE  JSR RANDOM
        LSR A ; REDUCE RANDOM BYTE
        LSR A ; TO 6 BITS FOR SHADE
        CMP VALUE ; VALUE COMPARISON
        BPL MORE
        JMP UNPLOT
        MORE  JMP PLOT
;
;*****
; PLOT A POINT WEIGHTED BY SHADING SCHEME
; AND SHADE VALUE
; CHECK 'NOSCAL' FLAG FOR SCALING OF Y COORD
; CHECK 'HTORRN' FLAG FOR TYPE OF SHADING
;
PLTSHD  LDA NOSCAL
        BEQ NORM
;
; SCALE Y FROM 0-239 PSEUDO-COORDINATES
; TO 0-199 TRUE SCREEN COORDINATES BY
; Y = (Y+1)*213/256
;
SCALE   LDY YPLT
        INY
        STY MLPLER
        LDA #D5 ; 213.
        STA MLPCND
        JSR MULT ; RETURN WITH HIGH BYTE
        STA YPLT ; IN ACCUMULATOR
        LDA HTORRN
        BEQ RPLT
        JMP SHADE
        RPLT  JMP RSHADE
;
;
;

```

```

THRESH .BYTE $00,$08,$35,$3D
        .BYTE $02,$0A,$37,$3F
        .BYTE $10,$18,$23,$2D
        .BYTE $12,$1A,$27,$2F
        .BYTE $31,$39,$04,$0C
        .BYTE $33,$3B,$06,$0E
        .BYTE $21,$29,$14,$1C
        .BYTE $23,$2B,$16,$1E
        .BYTE $03,$0B,$36,$3E
        .BYTE $01,$09,$34,$3C
        .BYTE $13,$1B,$26,$2E
        .BYTE $11,$19,$24,$2C
        .BYTE $32,$3A,$07,$0F
        .BYTE $30,$38,$05,$0D
        .BYTE $22,$2A,$17,$1F
        .BYTE $20,$28,$15,$1D
;
.END

```

Listing 2. Assembler-Quelltext der Grafik-Utilities (Schluß)

```

; FACET - DRAW SHADED TRIANGULAR FACETS
; AND STRAIGHT LINES.
;
; RICHARD L. RYLANDER 11/4/84
;
; LOAD "ARITH.HEX" AND "GRAPH.HEX"
; BEFORE USING
;
ORIGIN = $C26F
RAM = $B34A
;
XPLT = $033F
YPLT = $0341
NORM = $C224
NOSCAL = $B347
PLOT = $E143
UNPLOT = $C146
;
MLPCND = $AC
MLPLER = $AD
PROD = $AE
MULT = $C011
;
DVIDND = $FD
DVSOR = $FB
QUOT = $FD
DIVIDE = $C025
;
;==RAM
;
XMIN = **+2
YMIN = **+1
XMID = **+2
YMID = **+1
XMAX = **+2
YMAX = **+1
YTOP = **+1
YBOT = **+1
YBASE = **+1
DLTAX1 = **+2
DLTAX2 = **+1
DLTAX3 = **+1
DLTAX = **+1
DLTAY1 = **+1
DLTAY2 = **+1
DLTAY3 = **+1
DELTAY = **+1
XDIFF = **+1
FLAG1 = **+1
FLAG2 = **+1
FLAG3 = **+1
FLAG = **+1
EDGES = **+1
ERROR = **+2
MODE = **+1
COUNT = **+2
;
;==ORIGIN
;
;*****
; SCALE ALL Y COORDINATES FROM 0..239
; PSEUDO-COORDINATE RANGE TO 0..199
; TRUE SCREEN COORDINATE RANGE
;
SCALE   LDY #6
        LDA #D5
        STA MLPCND
        SCPL  LDA YMIN,Y
        STA MLPLER
        JSR MULT
        STA YMIN,Y
        DEY
        DEY
        DEY
        BPL SCPL
        RTS
;
;*****
; EXCHANGE 'MIN' AND 'MID' COORDINATES
;
SWAP12  LDY #2
        LOOP1 LDA XMIN,Y
        PHA
        LDA XMID,Y
        STA XMIN,Y
        PLA
        STA XMID,Y
        DEY
        DEY
        BPL LOOP1
        RTS
;
;*****
; EXCHANGE 'MID' AND 'MAX' COORDINATES
;
SWAP23  LDY #2
        LOOP2 LDA XMID,Y
        PHA
        LDA XMAX,Y
        STA XMID,Y
        PLA
        STA XMAX,Y
        DEY
        DEY
        BPL LOOP2
        RTS
;
;*****
; SORT COORDINATES ACCORDING TO X COMPONENTS
;
SORTX   LDX #2
        SORTLP SEC
        LDA XMID
        SBC XMIN
        LDA XMID+1
        SBC XMIN+1

```

```

BCS NOSWP1
JSR SWAP12
DEX
BEQ SORTED
SEC
LDA XMAX
SBC XMID
LDA XMAX+1
SBC XMID+1
BCS SORTLP
JSR SWAP23
JMP SORTLP
SORTED  RTS
;
;*****
; DRAW A LINE BETWEEN XMIN,YMIN AND XMID,YMID
; USING FAST DDA (DIGITAL DIFFERENTIAL ANALYZER)
; TECHNIQUE
;
LINE    LDA #2 ; ENSURE XMAX IS
        STA XMAX+1 ; LARGEST BEFORE
        JSR SORTX ; ORDERING 'MIN' AND 'MID'
        LDA NOSCAL
        BEQ OUTLN
        JSR SCALE
        JSR FINDXY ; ENTRY POINT TO
        LDA XMIN ; OUTLINE FACETS
        STA XPLT
        LDA XMIN+1
        STA XPLT+1
        LDA YMIN
        STA YPLT
        LDA DLTAX1+1 ; CHECK FOR DX>DY
        BNE STEPX
        SEC
        LDA DLTAX1
        SBC DLTAY1
        BCS STEPX
        LDA DLTAY1
        STA ERROR
        STA COUNT
        LSR ERROR
        SEC
        LDA DLTAX1
        SBC ERROR
        STA ERROR
        LDA DLTAX1+1
        SBC #0
        STA ERROR+1
        INC COUNT
        LNLFP1 LDA MODE ; 0 = DRAW, 1 = ERASE
        BNE ERASE1
        JSR PLOT
        JMP SK1
        ERASE1 JSR UNPLOT
        SK1    LDA FLAG1 ; 0 = POSITIVE SLOPE
        BNE NSLOPE
        INC YPLT
        BNE SK2 ; ALWAYS BRANCH
        NSLOPE DEC YPLT
        SK2    BIT ERROR+1
        BMI SK3
        INC XPLT
        BNE NOINC1
        INC XPLT+1
        NOINC1 SEC
        LDA ERROR
        SBC DLTAY1
        STA ERROR
        LDA ERROR+1
        SBC #0
        STA ERROR+1
        SK3    LDA ERROR
        LDA ERROR
        ADC DLTAX1
        STA ERROR
        LDA ERROR+1
        ADC DLTAX1+1
        STA ERROR+1
        DEC COUNT
        BNE LNLFP1
        RTS
;
STEPX   LDA DLTAX1
        STA ERROR
        STA COUNT
        LDA DLTAX1+1
        STA ERROR+1
        STA COUNT+1
        LSR ERROR+1
        KUR ERROR
        SEC
        LDA DLTAY1
        SBC ERROR
        STA ERROR
        LDA #0
        SBC ERROR+1
        STA ERROR+1
        LNLFP2 LDA MODE
        BNE ERASE2
        JSR PLOT
        JMP SKP1
        ERASE2 JSR UNPLOT
        SKP1   INC XPLT
        BNE NOINC2
        INC XPLT+1
        NOINC2 BIT ERROR+1
        BMI SKP3
        LDA FLAG1
        BNE NSLSP
        INC YPLT
        BNE SKP2 ; ALWAYS BRANCH
        NSLSP  DEC YPLT
        SKP2   SEC
        LDA ERROR
        SBC DLTAX1
        STA ERROR
        LDA ERROR+1
        SBC DLTAX1+1
        STA ERROR+1
        SKP3    CLC
        LDA ERROR
        ADC DLTAY1
        STA ERROR
        LDA ERROR+1
        ADC #0
        STA ERROR+1
        SEC
        LDA COUNT
        SBC #1
        STA COUNT
        BCS TEST
        DEC COUNT+1
        BIT COUNT+1
        BPL LNLFP2
        RTS

```

Listing 3. Assembler-Quelltext zum Zeichnen von Linien und Dreiecken

```

; *****
; DRAW A SHADED VERTICAL LINE AT
; XPLT FROM YTOP TO YBOT
;
VLINE SEC ; MAKE SURE YTOP>YBOT
LDA YTOP
SBC YBOT
BCS DRAW
LDA YTOP
PHA
LDA YBOT
STA YTOP
PLA
STA YBOT
LDA YTOP
STA YPLT
JSR NORM ; PLOT A SHADE-WEIGHTED
LDA YTOP ; PIXEL CHECKING ONLY
CMP YBOT ; FOR SHADE STYLE
BEQ DONE
DEC YTOP
JMP DRAW
DONE RTS
; *****
; FIND ENDPOINTS FOR VERTICAL LINES
; BETWEEN FACET EDGES
;
ENDPTS LDA XDIFF
STA MLPCND
LDA DELTAY
STA MLPLER
JSR MULT
STA DVND+1
LDA PROD
STA DVND
LDA #0
STA DVSOR+1
LDA DELTAX
STA DVSOR
JSR DIVIDE
LDA FLAG
BNE NEGLP
CLC
LDA YBASE
ADC QUOT
BCC SKIP2
SEC
NEGLP LDA YBASE
SBC QUOT
SKIP2 RTS
; *****
; FIND COORDINATE DIFFERENCES
;
; ALL "DELTA X" VALUES POSITIVE,
; SINGLE PRECISION (JUST LOWER BYTE)
;
FINDXY SEC
LDA XMID
SBC XMIN
STA DLTAX1
LDA XMID+1
SBC XMIN+1
STA DLTAX1+1
SEC
LDA XMAX
SBC XMID
STA DLTAX2
SEC
LDA XMAX
SBC XMIN
STA DLTAX3
;
; USE ABS(DELTA Y) VALUES,
; FLAGS INDICATE SLOPE OF LIMIT LINES
;
LDA #000
STA FLAG1
STA FLAG2
STA FLAG3
SEC
LDA YMID
SBC YMIN
BCS STORE1
INC FLAG1
LDA YMIN
SBC YMID
STORE1 STA DLTAY1
SEC
LDA YMAX
SBC YMID
BCS STORE2
INC FLAG2
LDA YMID
SBC YMAX
STORE2 STA DLTAY2
SEC
LDA YMAX
SBC YMIN
BCS STORE3
INC FLAG3
LDA YMIN
SBC YMAX
STORE3 STA DLTAY3
RTS
; *****
; DRAW A SHADED TRIANGULAR FACET
;
FACET JSR SORTX
LDA NOSCAL
BEQ YSOK
JSR SCALE
JSR FINDXY
LDA XMIN
STA XPLT
LDA XMIN+1
STA XPLT+1
FCETLP SEC
LDA XPLT
SBC XMIN
STA XDIFF
LDA DLTAX1
BEQ CONT
STA DELTAX
LDA DLTAY1
STA DELTAY
LDA FLAG1
STA FLAG
LDA YMIN
STA YBASE
JSR ENDPTS
STA YTOP
LDA DLTAX3
BEQ CONT
STA DELTAX
LDA DLTAY3
STA DELTAY
LDA FLAG3
STA FLAG
JSR ENDPTS

```

```

STA YBOT
JSR VLINE
LDA XPLT+1
CMP XMID+1
BNE NEXTX1
LDA XPLT
CMP XMID
BEQ CONT
NEXTX1 INC XPLT
BNE SKIP3
INC XPLT+1
SKIP3 JMP FCETLP
CONT SEC
LDA XPLT
SBC XMIN
STA XDIFF
LDA DLTAX3
BEQ FINI
STA DELTAX
LDA DLTAY3
STA DELTAY
LDA FLAG3
STA FLAG
LDA YMIN
STA YBASE
JSR ENDPTS
STA YBOT
SEC
LDA XPLT
SBC XMID
STA XDIFF
LDA DLTAX2
BEQ FINI
STA DELTAX
LDA DLTAY2
STA DELTAY
LDA FLAG2
STA FLAG
LDA YMID
STA YBASE
JSR ENDPTS
STA YTOP
JSR VLINE
LDA XPLT+1
CMP XMAX+1
BNE NEXTX2
LDA XPLT
CMP XMAX
BEQ FINI
NEXTX2 INC XPLT
BNE SKIP4
INC XPLT+1
SKIP4 JMP CONT
FINI LDA EDGES
BEQ FINISH
JSR SWAP23
JSR OUTLN
JSR SWAP12
JSR SWAP23
JSR SWAP12
JSR OUTLN
FINISH RTS
.END

```

Listing 3. Assembler-Quelltext zum Zeichnen von Linien und Dreiecken (Schluß)

```

; PRIMITIVE SOLID SHAPE DRAWING
;
; RICHARD L. RYLANDER 11/7/84
;
; LOAD ARITHMETIC AND GRAPHIC UTILITIES FIRST
;
*****
RAM=$036A
ORIGIN=$C5EA
;
MLPCND=$AC ; MULTIPLICAND (S)
MLPLER=$AD ; MULTIPLIER (S)
PROD=$AE ; PRODUCT (D)
MULT=$C011 ; CALL FOR MULTIPLY
;
DVND=$FD ; DIVIDEND (D)
DVSOR=$FB ; DIVISOR (D)
QUOT=$FD ; QUOTIENT (D)
DIVIDE=$C025 ; CALL FOR DIVIDE
;
ARG=$AC ; ARGUMENT (S)
SQR=$AE ; SQUARE OF ARG (D)
SQUARE=$C004 ; CALL FOR SQUARE
;
RADCN=$AC ; RADICAND (D)
ROOT=$B33C ; SQUARE ROOT (S)
SORT=$C064 ; CALL FOR SORT
;
RNDM=$C000 ; RANDOM NUMBER
RANDOM=$C008 ; CALL FOR RANDOM
; NOTE - A CALL TO "RANDOM" LEAVES A RANDOM BYTE
; IN THE ACCUMULATOR
;
XPLT=$033F
YPLT=$0341
NORM=$C224
PLTSHD=$C20F
VALUE=$0344 ; FINAL NORMALIZED SHADE VALUE
HTORRN=$0346 ; SHADE FLAG, 1=HALFTONE
NOSCAL=$0347 ; SCALE FLAG, 1=NO SCALE
;
**RAM
XCENT **$+2 ; CENTER COORD
XREL **$+1 ; RELATIVE (TO CENTER)
XSHD **$+2 ; USED IN SHADE CALC
YCENT **$+1 ; CENTER COORD
YREL **$+1 ; RELATIVE (TO CENTER)
YSHD **$+2 ; USED IN SHADE CALC
ZREL **$+2 ; RELATIVE (TO CENTER)
ZWX **$+2 ; Z WITH X (+ OR -)
;
RADIUS **$+2 ; LOCAL RADIUS OF SURFACE
TONE **$+2 ; USED IN SHADE CALC
TNTP **$+2 ; USED IN SHADE CALC
;
CLIPL **$+1 ; LEFT CLIPPING BOUND
CLIPR **$+1 ; RIGHT CLIPPING BOUND
CLIPU **$+1 ; UP CLIPPING BOUND
CLIPD **$+1 ; DOWN CLIPPING BOUND
;
HEMI **$+1 ; PLOTTING HEMISPHERE
;
BAKLIT **$+1 ; BACKLIT FLAG
HVFLAG **$+1 ; HORIZONTAL/VERTICAL FLAG
TEMP **$+2 ; TEMPORARY STORAGE
CNTX **$+1 ; LOOP COUNTER
CNTY **$+1 ; LOOP COUNTER
MAX **$+1 ; LOOP LIMIT
;
HLEN **$+1 ; HALF-LENGTH OF CYLINDERS
RS **$+2 ; SQUARE OF TOROID RADIUS
RT **$+1 ; TOROID (RING) RADIUS
RC **$+1 ; CENTER RADIUS OF TOROID
RO **$+1 ; OUTER RADIUS OF TOROID

```

```

RI **$+1 ; INNER RADIUS OF TOROID
XSQR **$+2
XMAX **$+1
;
R0=HLEN
;
**ORIGIN
; *****
; DIVIDE WITH SINGLE PRECISION DIVISOR
; (USED OFTEN IN SHAPE ROUTINES)
;
SDIV LDA #0
STA DVSOR+1
JMP DIVIDE
; *****
; CALCULATE SHADE VALUE (0-63) BY
; MULTIPLYING "TONE" BY 26 THEN
; DIVIDE RESULT BY RADIUS OF SURFACE
;
GETVAL BIT TONE+1
BPL CNTNU ; IF "TONE"<0, THEN
LDA BAKLIT ; MAKE VALUE 0 OR ABS(TONE)
BNE NEGATE ; DEPENDING ON BAKLIT FLAG
STA VALUE
RTS
NEGATE SEC
LDA #000
SBC TONE
STA VALUE
LDA TONE
CNTNU STA MLPCND
LDA #1A
STA MLPLER
JSR MULT
STA DVND+1
LDA PROD
STA DVND
LDA RADIUS
STA DVSOR
JSR SDIV
LDA QUOT
STA VALUE
RTS
; *****
; POINT PLOTTING BY QUADRANTS USING
; THE FOUR-FOLD SYMMETRY OF SIMPLE OBJECTS
;
; DEPENDING ON STATUS OF "HVFLAG", EXCHANGE
; X AND Y COORDINATES TO ROTATE OBJECTS 90 DEG
; SINGLE SHAPE ROUTINE CAN THEN BE USED TO
; DRAW "HORIZONTAL" OR "VERTICAL" VERSIONS
; OF AN OBJECT
;
; THE FOLLOWING IS A "BASIC SUBROUTINE"
; EQUIVALENT TO EXPLAIN ITS OPERATION
;
; NOTE THAT LABELS ARE USED IN PLACE OF
; LINE NUMBERS
;
; "PTPLOT" IF HVFLAG=0 THEN GOTO "NOROT"
; (STACK)=XREL: XREL=YREL: YREL=(STACK)
; (STACK)=XSHD: XSHD=YSHD: YSHD=(STACK)
; "NOROT" GOSUB "GETZ"
; REM CALCULATE 2*Z FROM X,Y AND RADIUS
; HEMI = 1
; IF XREL>CLIPL THEN GOTO "RHEMI"
; ZWX=2*Z-XSHD
; XPLT=XCENT-XREL: REM LEFT HEMISPHERE
; "CHCLUP" IF YREL>CLIPU THEN GOTO "DHEMI"
; TONE=ZWX+YSHD
; GOSUB "GETVAL": REM NORMALIZE SHADE VAL
; YPLT=YCENT+YREL
; GOSUB "PLTSHD": REM PLOT OR UNPLOT
; REM POINTS WEIGHTED BY SHADE VALUE
; "DHEMI" IF YREL>CLIPD THEN GOTO "RHEMI"
; TONE=ZWX-YSHD
; GOSUB "GETVAL"
; YPLT=YCENT-YREL
; GOSUB "PLTSHD"
; "RHEMI" IF HEMI=0 THEN RETURN
; HEMI=0
; IF XREL>CLIPR THEN RETURN
; ZWX=2*Z+YSHD
; XPLT=XCENT+XREL
; GOSUB "CHCLUP"
; RETURN
;
PTPLOT BIT HVFLAG
BPL NOROT
LDA XREL
PHA
PHA
LDA YREL
STA XREL
PLA
STA YREL
LDA XSHD
PHA
PHA
LDA YSHD
STA XSHD
PLA
STA YSHD
LDA XSHD+1
PHA
PHA
LDA YSHD+1
STA XSHD+1
PLA
STA YSHD+1
NOROT JSR GETZ
PTPLT2 LDA #001
STA HEMI
SEC
LDA CLIPL ; CHECK LEFT HEMISPHERE
CMP XREL
BCC RHEMI
SEC
LDA ROOT
SBC XSHD
STA ZWX
LDA ROOT+1
SBC XSHD+1
STA ZWX+1
SEC
LDA XCENT
SBC XREL
STA XPLT
LDA XCENT+1
SBC #000
STA XPLT+1
SEC
CHCLUP LDA CLIPU ; CHECK FOR UP CLIPPING
CMP YREL
BCC DHEMI
CLC
LDA ZWX
ADC YSHD

```

Listing 4. Assembler-Quelltext zum Zeichnen ausgefüllter Flächen, sogenannter Shapes

```

STA TONE
LDA ZWX+1
ADC YSHD+1
STA TONE+1
JSR GETVAL
CLC
LDA YCENT
ADC YREL
STA YPLT
JSR PLTSHD

;
DHEMI SEC
LDA CLIPD ; CHECK FOR DOWN CLIPPING
CMP YREL
BCC RHEMI
SEC
LDA ZWX
SBC YSHD
STA TONE
LDA ZWX+1
SBC YSHD+1
STA TONE+1
JSR GETVAL
SEC
LDA YCENT
SBC YREL
STA YPLT
JSR PLTSHD

;
RHEMI LDA HEMI
BEQ PLDONE
DEC HEMI
SEC
LDA CLIPR ; CHECK FOR RIGHT CLIPPING
CMP XREL
BCC PLDONE
CLC
LDA ROOT
ADC XSHD
STA ZWX
LDA ROOT+1
ADC XSHD+1
STA ZWX+1
CLC
LDA XCENT
ADC XREL
STA XPLT
LDA XCENT+1
ADC XSHD+1
STA XPLT+1
JMP CHCLUP
PLDONE BIT HVFLAG
BPL NORSTR
LDA XSHD+1 ; RESTORE COORDS
STA YSHD+1
PLA
STA XSHD+1
LDA XSHD
STA YSHD
PLA
STA XSHD
LDA XREL
STA YREL
PLA
STA XREL
NORSTR RTS
;
; *****
; CALCULATE Z FROM LOCAL X,Y BY
; PYTHAGOREAN SUM
;
GETZ LDA RADIUS
STA ARG
JSR SQUARE
STA TINTMP+1
LDA SQR
STA TINTMP
LDA XSHD
STA ARG
JSR SQUARE
SEC
LDA TINTMP
SBC SQR
STA TINTMP
LDA TINTMP+1
SBC SQR+1
STA TINTMP+1
LDA YSHD
STA ARG
JSR SQUARE
SEC
LDA TINTMP
SBC SQR
STA RADCN
LDA TINTMP+1
SBC SQR+1
STA RADCN+1
BMI ZEROOT
JSR SORT
ASL ROOT
ROL ROOT+1
RTS
ZEROOT LDA #0
STA ROOT
STA ROOT+1
RTS

;
; *****
; SET UP PARAMETERS FOR TOROIDS
;
RT=(RO-R1)/2 RS=RT*RT RC=RT+R1
;
TPARM LDA RO
SEC
SBC R1
LSR A
STA RT
STA RADIUS
CLC
ADC R1
STA RC
LDA RT
STA ARG
JSR SQUARE
LDA SQR
STA RS
LDA SQR+1
STA RS+1
LDA #0
STA CNTX
RTS

;
; *****
; DRAW A SHADED SPHERE
;
; 'BASIC SUBROUTINE' EQUIVALENT
;
; 'SPHERE' FOR CNTX=0 TO RADIUS/SQR(2)
; XREL=CNTX; XSHD=CNTX
; FOR CNTY=CNTX TO SQR(RAD*RAD-CNTX*CNTX)
; YREL=CNTY; YSHD=CNTY
; HVFLAG=0
; GOSUB 'PTPLOT'
; REM EXCHANGE X & Y TO USE 8-FOLD SYM
; HVFLAG=-128
;
; GOSUB 'PTPLOT'
; NEXT CNTY

```

```

;
; NEXT CNTX
; RETURN
;
SPHERE LDA RADIUS
STA ARG
JSR SQUARE
ASL SQR
ROL SQR+1
LDA SQR
STA RADCN
LDA SQR+1
STA RADCN+1
JSR SORT
LSR ROOT+1
ROL ROOT
LDA ROOT
STA XMAX
LDA #0
STA CNTX
STA XSHD+1
STA YSHD+1
LDA RADIUS
STA ARG
JSR SQUARE
STA TEMP+1
LDA SQR
STA TEMP
LDA CNTX
STA CNTY
STA ARG
STA XREL
STA XSHD
JSR SQUARE
SEC
LDA TEMP
SBC SQR
STA RADCN
LDA TEMP+1
SBC SQR+1
STA RADCN+1
JSR SORT
LDA ROOT
STA MAX
LDA CNTY
STA YREL
STA YSHD
LDA #0
STA HVFLAG
JSR PTPLOT
LDA #0
STA HVFLAG
JSR PTPLOT
LDA CNTY
CMP MAX
BEQ DONEY
INC CNTY
JMP LOOPY
LDA CNTX
CMP XMAX
BEQ DONE
INC CNTX
JMP LOOPX
DONE RTS
;
; *****
; DRAW SHADED CYLINDERS
;
; 'BASIC SUBROUTINE' EQUIVALENT
;
; 'CYLND' XSHD=0
; FOR YREL=RADIUS TO 0
; YSHD=YREL
; FOR XREL=HLEN TO 0
; GOSUB 'PTPLOT'
; NEXT XREL
; NEXT YREL
; RETURN
;
CYLND LDA #0
STA XSHD
STA XSHD+1
STA YSHD+1
LDA RADIUS
STA YREL
CYLOOP LDA HLEN
STA XREL
LDA YREL
STA YSHD
JSR PTPLOT
DEC XREL
BPL CXLOOP
DEC YREL
BPL CYLOOP
RTS

;
; *****
; DRAW EDGE-VIEW TOROIDS
;
; 'BASIC SUBROUTINE' EQUIVALENT
;
; 'EDGTOR' GOSUB 'TPARM'; REM SET UP RADII
; FOR CNTX=0 TO RT
; XREL=CNTX; XSHD=CNTX
; RB=SQR(RT*RT-CNTX*CNTX)
; FOR CNTY=0 TO RB+RC
; YREL=CNTY
; YSHD=(RB+CNTY)/(RB+RC)
; GOSUB 'PTPLOT'
; NEXT CNTY
; NEXT CNTX
; RETURN
;
EDGTOR JSR TPARM
LDA #0
STA XSHD+1
STA YSHD+1
LDA CNTX
STA XREL
STA XSHD
STA ARG
JSR SQUARE
SEC
LDA RS
SBC SQR
STA RADCN
LDA RS+1
SBC SQR+1
STA RADCN+1
JSR SORT
LDA ROOT
STA R0
CLC
ADC RC
STA MAX
LDA #0
STA CNTY
LDA CNTY
STA YREL
STA MLPLER
LDA R0
STA MLPCND
JSR MULT
STA DVND+1
LDA PROD
STA DVND
JSR SDIV
SEC
LDA CNTX
SBC QUOT
STA XSHD
LDA #0
SBC QUOT+1
STA XSHD+1
LDA CNTY
STA MLPLER
LDA RC
STA MLPCND
JSR MULT
STA DVND+1
LDA PROD
STA DVND
LDA R0
STA DVSOR

```

```

LDA QUOT
STA YSHD
JSR PTPLOT
LDA CNTY
CMP MAX
BEQ DONE4
INC CNTY
JMP LOOPY4
DONE4 LDA CNTX
CMP RT
BEQ DONEHT
INC CNTX
JMP LOOPX4
DONEHT RTS
;
; *****
; DRAW A SHADED, TOP-VIEW TOROID
;
; 'BASIC SUBROUTINE' EQUIVALENT
;
; 'TOROID' GOSUB 'TPARM'
; FOR CNTX=0 TO RO/SQR(2)
; REM 8-FOLD SYMMETRY USED
; XREL=CNTX
; MAX=SQR(RO*RO-CNTX*CNTX)
; IF CNTX/R1 THEN GOTO 'GRTR'
; CNTY=SQR(R1*R1-CNTX*CNTX)
; GOTO 'LLPY1'
; 'GRTR' CNTY=CNTX
; 'LLPY1' YREL=CNTY
; RB=SQR(CNTY*CNTY+CNTX*CNTX)
; XSHD=CNTX-(CNTX*RC)/R0
; YSHD=CNTY-(CNTX*RC)/R0
; HVFLAG=0; GOSUB 'PTPLOT'
; HVFLAG=-128; GOSUB 'PTPLOT'
; IF CNTY=MAX THEN GOTO 'DDNY1'
; CNTY=CNTY+1
; GOTO 'LLPY1'
; 'DDNY1' NEXT CNTX
; RETURN
;
TOROID JSr TPARM
LDA RO
STA ARG
JSR SQUARE
ASL SQR
ROL SQR+1
LDA SQR
STA RADCN
LDA SQR+1
STA RADCN+1
JSR SORT
LSR ROOT+1
ROL ROOT
LDA ROOT
STA XMAX
LDA CNTX
STA XREL
STA ARG
JSR SQUARE
STA XSOR+1
LDA SQR
STA XSOR
LDA RO
STA ARG
JSR SQUARE
SEC
LDA SQR
SBC XSOR
STA RADCN
LDA SQR+1
SBC XSOR+1
STA RADCN+1
JSR SORT
LDA ROOT
STA MAX
SEC
LDA R1
SBC CNTX
BCC GRTR
LDA R1
STA ARG
JSR SQUARE
SEC
LDA SQR
SBC XSOR
STA RADCN
LDA SQR+1
SBC XSOR+1
STA RADCN+1
JSR SORT
LDA ROOT
STA CNTY
JMP LLPY1
LDA CNTX
STA CNTY
LLPY1 LDA CNTY
STA YREL
STA ARG
JSR SQUARE
CLC
LDA SQR
ADC XSOR
STA RADCN
LDA SQR+1
ADC XSOR+1
STA RADCN+1
JSR SORT
LDA ROOT
STA R0
STA DVSOR
LDA CNTX
STA MLPLER
LDA RC
STA MLPCND
JSR MULT
STA DVND+1
LDA PROD
STA DVND
JSR SDIV
SEC
LDA CNTX
SBC QUOT
STA XSHD
LDA #0
SBC QUOT+1
STA XSHD+1
LDA CNTY
STA MLPLER
LDA RC
STA MLPCND
JSR MULT
STA DVND+1
LDA PROD
STA DVND
LDA R0
STA DVSOR

```

Listing 4. Assembler-Quelltext zum Zeichnen ausgefüllter Flächen, sogenannter Shapes (Fortsetzung)

```

JSR SDIV
SEC
LDA CNTY
SBC QUOT
STA YSHD
LDA #000
STA HVFLAG
SBC QUOT+1
STA YSHD+1
JSR PTPLT
LDA #000
STA HVFLAG
JSR PTPLT
LDA CNTY
CMP MAX
BEQ DDNY1
INC CNTY
JMP LLPY1
LDA CNTX
CMP XMAX
BEQ DUNTOR
INC CNTX
JMP LLPX1
DUNTOR RTS
; *****
; DRAW "INSIDE VIEW" TOROIDS
;
; 'BASIC SUBROUTINE' EQUIVALENT
;
; 'SPOOL' GOSUB 'TPARM'
;   FOR CNTX=0 TO RT
;   XREL=CNTX:XSHD=CNTX
;   MAX=RC-SQR(RS-CNTX*CNTX)
;   FOR CNTY=0 TO MAX
;   YREL=CNTY
;   YSHD=(RC*CNTY/MAX)-CNTY
;   GOSUB 'PTPLT'
;   NEXT CNTY
;   NEXT CNTX
;   RETURN
;
SPOOL JSR TPARM
LLPX2 LDA CNTX
STA XREL
STA ARG
SEC
LDA #000
SBC CNTX
STA XSHD
LDA #000
SBC #000
STA XSHD+1
JSR SQUARE
SEC
LDA RS
SBC SUK
STA RADCN
LDA RS+1
SBC SQR+1
STA RADCN+1
JSR SORT
SEC
LDA RC
SBC ROOT
STA MAX
LDA #000
LDA CNTY
LLPY2 LDA CNTX
STA YREL
STA MLPLER
LDA RC
STA MLPCND
JSR MULT
STA DVDND+1
LDA PROD
STA DVDND
LDA MAX
STA DVSQR
JSR SDIV
LDA QUOT
SEC
SBC CNTY
STA YSHD
LDA QUOT+1
SBC #000
STA YSHD+1
JSR PTPLT
LDA CNTY
CMP MAX
BEQ DDNY2
INC CNTY
JMP LLPY2
LDA CNTX
CMP RT
BEQ DUNHSP
INC CNTX
JMP LLPX2
DUNHSP RTS
.END

```

Listing 4. Assembler-Quelltext zum Zeichnen ausgefüllter Flächen, sogenannter Shapes (Schluß)

```

; INTERFACE - EASY PARAMETER SETTING FOR SHAPE
; DRAWING ROUTINES FROM BASIC.
;
; RICHARD L. RYLANDER 11/23/84
;
; *****
ORIGIN=$CACB
RAM=$0393
;
; PARAMETER LOCATIONS FOR VARIOUS SHAPES
;
XCENT=$036A
YCENT=$036F
XPLOT=$033F
YPLOT=$0341
XMIN=$034A
YMIN=$034C
XMID=$034D
YMID=$034F
XMAX=$0350
YMAX=$0352
RADIUS=$0377
HLEN=$0389
RI=$038F
RO=$038E
;
HVFLAG=$0383
VALUE=$0344
PLTFLG=$033E
;
DEFLAG=$FB
;

```

```

; *****
; FUNCTION LOCATIONS
;
GRFON=$C0E2 ; SWITCH TO GRAPHICS MODE
GRFOFF=$C1B3 ; RETURN TO TEXT DISPLAY
;
CLEARR=$C12C ; CLEAR BITMAP
CLRBYT=$C135 ; CLEAR (FILL) BYTE
COLORR=$C118 ; LOAD COLOR MAP
COLBYT=$C119 ; COLOR BYTE
;
PLOT2=$C14B ; POINT PLOT ROUTINE
LINER=$C2DB ; DRAW A LINE
FACETR=$C4E1 ; DRAW A SHADED FACET
;
; *****
; SHADED SHAPE DRAWING ROUTINES
;
SPHERR=$C7C7 ; SPHERE
CYLNDR=$C864 ; CYLINDER
TORUSR=$C90F ; TOP-VIEW TOROID
EDGTOR=$C8BF ; EDGE-VIEW TOROID
SPOULR=$CA3B ; INSIDE-VIEW TOROID
;
; *****
; BASIC ROM ROUTINES
;
CHKCOM=$AEFD ; CHECK FOR COMMA
EVAEXP=$AD9E ; EVALUATE EXPRESSION
FLTFLX=$B1AA ; CONVERT TO FIXED
;
**=RAM
LINFAC **+=1 ; LINE OR FACET FLAG
;
**=ORIGIN
;
; *****
; GET PARAMETERS FROM BASIC CALLING STATEMENT
; OF THE FORM:
; SYS(FUNCTN),PARAM1,PARAM2,PARAM3,ADPTU
; WHERE THE THIRD PARAMETER (FOR EXAMPLE)
; MAY BE OPTIONAL (A DEFAULT VALUE IS USED
; IF THE PARAMETER IS NOT SPECIFIED)
;
GETNUM JSR CHKCOM ; LOOK FOR COMMA
JSR EVAEXP ; EVALUATE EXPRESSION
JSR FLTFLX ; CHANGE TO INTEGER WITH
; HIGH BYTE IN "A" AND LOW BYTE IN "Y"
RTS
;
; CHECK FOR ADDITIONAL (OPTIONAL) PARAMETERS
;
PCHECK LDA #02C ; ", " COMMA
LDY #0
STY DEFLAG
CMP ($7A),Y
BNE NOMORE ; NO COMMA - USE DEFAULT
JMP $0073
NOMORE LDY #0
STY DEFLAG
RTS
;
; GET TWO ADDITIONAL PARAMETERS FOR TOROIDS
;
GETTWO JSR PCHECK
BIT DEFLAG
BMI DFALT
JSR EVAEXP
JSR FLTFLX
STY RI
JSR GETNUM
STY RO
DFALT RTS
;
; *****
; SET CENTER COORDINATES
;
CENTER JSR GETNUM
STY XCENT
STA XCEN+1
JSR GETNUM
STY YCENT
RTS
;
; *****
; CLEAR THE BITMAP, FILLING WITH (OPTIONAL)
; FILL VALUE SPECIFIED OR WITH (DEFAULT) "0"
;
CLEAR2 JSR PCHECK
BIT DEFLAG
BMI DEFCLR
JSR EVAEXP
JSR FLTFLX
LDY #0
STY CLRBYT
JMP CLEAR
;
; *****
; FILL COLOR MAP WITH (OPTIONAL) COLOR BYTE
; SPECIFIED OR WITH (DEFAULT) "01"
; (BLACK DOTS ON WHITE BACKGROUND)
;
COLOR2 JSR PCHECK
BIT DEFLAG
BMI DEFCLR
JSR EVAEXP
JSR FLTFLX
LDY #0
STY COLBYT
JMP COLORR
;
; *****
; PLOT OR UNPLOT POINTS
;
PLOT2 LDA #0
LDY #0
STY PLTFLG
JSR GETNUM
STY XPLOT
STA XPLOT+1
JSR GETNUM
STY YPLOT
JMP PLOT2
;
UNPLT2 LDA #0
LDY #0
STY PLTFLG
JSR GETNUM
STY XPLOT
STA XPLOT+1
JSR GETNUM
STY YPLOT
JMP PLOT2
;

```

```

; *****
; DRAW LINES BETWEEN (X1,Y1) AND (X2,Y2)
; OR SHADED FACETS BETWEEN THREE POINTS
; (X1,Y1), (X2,Y2) AND (X3,Y3)
;
LINE2 LDA #0
LDY #0
STY XMIN
STA XMIN+1
JSR GETNUM
STY YMIN
STA YMIN+1
JSR GETNUM
STY XMID
STA XMID+1
JSR GETNUM
STY YMID
BIT LINFAC
BPL LDRAW
JSR GETNUM
STY XMAX
STA XMAX+1
JSR GETNUM
STY YMAX
STA YMAX+1
JSR GETNUM
STY VALUE
JMP FACETR
LDRAW JMP LINER
;
; *****
; DRAW A SPHERE CENTERED AT (XCEN,YCEN)
; DEFAULT RADIUS IS LAST VALUE USED
;
SPHER2 JSR CENTER
JSR PCHECK
BIT DEFLAG
BMI SKIP1
JSR EVAEXP
JSR FLTFLX
STY RADIUS
SKIP1 JMP SPHERR
;
; *****
; DRAW A TOP-VIEW TOROID AT (XCEN,YCEN)
; DEFAULT INNER AND OUTER RADII ARE LAST USED
;
TORUS2 JSR CENTER
JSR GETTWO
JMP TORUSR
;
; *****
; DRAW CYLINDERS WITH AXES HORIZONTAL OR
; VERTICAL. DEFAULT RADIUS AND "HALF-LENGTH"
; ARE LAST VALUES USED.
;
VCYL2 LDA #0
LDY #0
STY HVFLAG
HCYL2 LDA #0
STA HVFLAG
JSR CENTER
JSR PCHECK
BIT DEFLAG
BMI SKIP2
JSR EVAEXP
JSR FLTFLX
STY RADIUS
JSR GETNUM
STY HLEN
SKIP2 JMP CYLNDR
;
; *****
; DRAW EDGE-VIEW TOROIDS WITH AXES HORIZONTAL
; OR VERTICAL
; INNER AND OUTER RADII ARE OPTIONAL
;
VTOR2 LDA #0
LDY #0
STY HVFLAG
HTOR2 LDA #0
STA HVFLAG
JSR CENTER
JSR GETTWO
JMP EDGTOR
;
; *****
; DRAW INSIDE-VIEW TOROIDS, "SPOOLS",
; WITH AXES HORIZONTAL OR VERTICAL
; INNER AND OUTER RADII ARE OPTIONAL
;
VSPL2 LDA #0
LDY #0
STY HVFLAG
HSPL2 LDA #0
STA HVFLAG
JSR CENTER
JSR GETTWO
JMP SPOULR
.END

```

Listing 5. Assembler-Quelltext des »Basic-Interfaces« zum einfachen Ansprechen aller Assembler-Routinen per SYS-Befehl

```

; KEYSORT - RELOCATABLE BUBBLE SORT USING KEY ARRAY
; POINTING TO INTEGER ARRAY
;
; RICHARD L. RYLANDER 1/12/85
;
ORIGIN=$CF59 ; $30B1. (FOLLOWING DOS 5.1)
;
KB=$FB ; 251. POINTER TO KEY ARRAY
ZB=$FD ; 253. POINTER TO DATA ARRAY
MAX=$BC ; 140. POKE WITH MAX ARRAY INDEX
TOP=$AC
TOPDIS=$AD
FLAG=$AE
NXTFLG=$61

```

Listing 6. Assembler-Quelltext der schnellen Bubble-Sort-Routine

```

CRRT = $62
REPEAT = $64
;
; **ORIGIN
;
; INIT LDY #FFF ; INITIALIZE KEY ARRAY
INLOOP INY
TAY
STA (KB),Y
CMP MAX
BNE INLOOP
;
; SORT STA TOPDIS ; 'A' HOLDS 'MAX'
LOOP1 LDA TOPDIS
STA TOP
LDX #0
STX NXTFLG
STX FLAG
LOOP2 STX REPEAT
;
; GET BOTH BYTES OF INTEGER POINTED TO BY
; 'KEY' ELEMENT. RETURN WITH MSB ON STACK
; AND LSB IN THE ACCUMULATOR
;
GETINT TXA
TAY
LDA (KB),Y
ASL A
BCC LOAD
DEC NXTFLG
INC ZB+1
LOAD TAY
LDA (ZB),Y
PHA
INY
LDA (ZB),Y
BIT NXTFLG
BPL NODEC
INC NXTFLG
DEC ZB+1
NODEC CPX REPEAT
BNE ORDER
STA CRRT
PLA
STA CRRT+1
INX
BNE GETINT
;
; COMPARE INTEGERS OBTAINED THROUGH KEY ARRAY
; IF 'CURRENT' >= 'NEXT' THEN SWAP KEY
; ELEMENTS, ELSE CONTINUE
;
ORDER CMP CRRT
PLA
SBC CRRT+1
BVC TEST
EOR #80
TEST BPL NOSWAP
SWAP TXA
TAY
STX TOPDIS
LDA (KB),Y
PHA
DEY
LDA (KB),Y
INY
STA (KB),Y
PLA
DEY
STA (KB),Y
INC FLAG
NOSWAP CPX TOP
BNE LOOP2
LDA FLAG
BNE LOOP1
;
; UNPACK THE BYTE ELEMENTS OF THE 'KEY' ARRAY
; INTO BASIC'S NORMAL 2-BYTE INTEGER FORMAT
;
UNPACK LDX MAX
INX
PKLOOP DEX
TXA
TAY
LDA (KB),Y
PHA
TXA
ASL A ; MOVE TO 2*+1
ORA #1
BCC STORE
INC NXTFLG
INC KB+1
STORE TAY
PLA
STA (KB),Y
LDA #0
DEY
STA (KB),Y
LDA NXTFLG
BEQ OK
DEC NXTFLG
DEC KB+1
OK TXA
BNE PKLOOP
DONE RTS
.END

```

Listing 6. Assembler-Quelltext der schnellen Bubble-Sort- Routine (Schluß)

```

; "WRITE" RICHARD L. RYLANDER
; 12/30/84
; REVISED 1/19/85 - ORIGIN MOVED TO $CFE5 (53221.)
;
; PUT TEXT CHARACTERS ON GRAPHIC SCREEN
; (UNDER BASIC ROM) IN VARIOUS STYLES
;
; **CFE5 ; PUT CODE AFTER DOS 5.1
WRITE LDA #01 ; SWITCH OUT BASIC ROM
AND #FFE
STA #01
LDY #7
LOOP LDA ($FD),Y ; READ CHARACTER BYTE
AND ($FB),Y ; MODIFY W/SCREEN BYTE
STA ($FB),Y ; STORE IN SCREEN
;
; POKE NEW LOGICAL OPERATOR TO REPLACE
; 'AND' (53221.) FOR DIFFERENT STYLES
; ORA=17, BIT (NOP)=36, AND=49, EOR=81.
;
DEY
BPL LOOP
LDA #01 ; RESTORE BASIC ROM
ORA #1
STA #01
RTS
.END

```

Listing 7. Assembler-Quelltext der Routine zum Beschreiben des HiRes-Bildschirmes unterhalb des Basic-ROMs

```

programm : graphics c000 cbfc
c000 : ff 55 00 00 a5 ac 10 07 b8
c008 : 38 a9 00 e5 ac 85 ac 85 86
c010 : ad a9 00 a2 08 46 ad 90 71
c018 : 03 18 65 ac 6a 66 ae ca 40
c020 : d0 f3 85 af 60 a9 00 85 9f
c028 : b4 85 b5 a2 10 26 fd 26 d7
c030 : fe 26 b4 26 b5 38 a5 b4 50
c038 : e5 fb a8 a5 b5 e5 fc 90 99
c040 : 04 84 b4 85 b5 ca d0 e5 25
c048 : 26 fd 26 fe 06 b4 26 b5 e0
c050 : b0 0b 38 a5 fb e5 b4 a5 56
c058 : fc e5 b5 b0 06 e6 fd d0 fb
c060 : 02 e6 fe 60 a2 08 a9 00 b2
c068 : 8d 3c 03 8d 3d 03 85 fb 80
c070 : 85 fc 0e 3c 03 2e 3d 03 1b
c078 : ee 3c 03 d0 03 ee 3d 03 02
c080 : 06 ac 26 ad 26 fb 26 fc f0
c088 : 06 ac 26 ad 26 fb 26 fc f8
c090 : 38 a5 fb ed 3c 03 a8 a5 21
c098 : fc ed 3d 03 90 12 85 fc e4
c0a0 : 84 fb ee 3c 03 d0 03 ee 06
c0a8 : 3d 03 ca d0 c5 4c c1 c0 7b
c0b0 : 38 ad 3c 03 e9 01 8d 3c 84
c0b8 : 03 b0 03 ce 3d 03 ca d0 67
c0c0 : b1 6e 3d 03 6e 3c 03 60 ee
c0c8 : ad 00 c0 8d 02 c0 4d 01 b4
c0d0 : c0 2e 03 c0 6a 6e 03 c0 28
c0d8 : 8d 00 c0 ad 02 c0 8d 01 a9
c0e0 : c0 60 ad 11 d0 09 20 8d 4f
c0e8 : 11 d0 ad 00 dd 8d 42 03 26
c0f0 : 29 fc 09 01 8d 00 dd ad a5
c0f8 : 18 d0 8d 43 03 a9 19 8d 41
c100 : 18 d0 60 ad 11 d0 29 df 4a
c108 : 8d 11 d0 ad 42 03 8d 00 7a
c110 : dd ad 43 03 8d 18 d0 60 93
c118 : a9 01 a2 00 9d 00 84 9d 12
c120 : 00 85 9d 00 86 9d 00 87 af
c128 : ca d0 f1 60 a9 a0 85 fc 92
c130 : a0 00 84 fb a9 00 a2 20 d6
c138 : 91 fb c8 d0 fb e6 fc ca 93
c140 : d0 f6 60 a9 00 2c a9 80 e2
c148 : 8d 3e 03 a5 01 29 fe 85 ca
c150 : 01 38 a9 c7 ed 41 03 aa 1b
c158 : 4a 4a 4a a8 b9 ab c1 85 7a
c160 : fb b9 c4 c1 85 fc 8a 29 5e
c168 : 07 18 65 fb 85 fb ad 3f c1
c170 : 03 29 fb 65 fb 85 fb ad 2a
c178 : 40 03 65 fc 85 fc a9 a0 5b
c180 : 65 fc 85 fc ad 3f 03 29 97
c188 : 07 49 07 aa a9 01 ca 30 79
c190 : 03 0a d0 fa a0 00 2c 3e 63
c198 : 03 10 05 49 ff 31 fb 2c e0
c1a0 : 11 fb 91 fb a5 01 09 01 1b
c1a8 : 85 01 60 00 40 80 c0 00 d1
c1b0 : 40 80 c0 00 40 80 c0 00 6c
c1b8 : 40 80 c0 00 40 80 c0 00 74
c1c0 : 40 80 c0 00 00 01 02 03 87
c1c8 : 05 06 07 08 0a 0b 0c 0d d7
c1d0 : 0f 10 11 12 14 15 16 17 de
c1d8 : 19 1a 1b 1c 1e ad 3f 03 9b
c1e0 : 29 07 8d 48 03 ad 41 03 a2
c1e8 : 29 07 0a 0a 0a 0d 48 03 89
c1f0 : aa bd 2f c2 cd 44 03 10 c8
c1f8 : 03 4c 4e c1 4c 43 c1 20 11
c200 : c8 c0 4a 4a cd 44 03 10 2f
c208 : 03 4c 46 c1 4c 43 c1 ad 3c
c210 : 47 03 f0 10 ac 41 03 c8 89
c218 : 84 ad a9 d5 85 ac 20 11 f8
c220 : c0 8d 41 03 ad 46 03 f0 53
c228 : 03 4c dd c1 4c ff c1 00 cd
c230 : 08 35 cd 02 0a 37 3f 10 da
c238 : 18 25 2d 12 1a 27 31 6a
c240 : 39 04 0c 33 3b 06 0e 21 43
c248 : 29 14 1c 23 2b 16 1e 03 c9
c250 : 0b 36 3e 01 09 34 3c 13 6f
c258 : 1b 26 2e 11 19 24 32 fc
c260 : 3a 07 0f 30 38 05 0d 22 0c
c268 : 2a 17 1f 20 28 15 1d a0 cb
c270 : 06 a9 d5 85 ac b9 4c 03 41
c278 : 85 ad 20 11 c0 99 4c 03 0e
c280 : 88 88 88 10 f0 60 a0 02 09
c288 : b9 4a 03 48 b9 4d 03 99 75
c290 : 4a 03 68 99 4d 03 88 10 d8
c298 : ef 60 a0 02 b9 4d 03 48 c2
c2a0 : b9 50 03 99 4d 03 68 99 37
c2a8 : 50 03 88 10 ef 60 a2 02 2e
c2b0 : 38 ad 4d 03 ed 4a 03 ad 0b
c2b8 : 4e 03 ed 4b 03 b0 03 20 6f
c2c0 : 86 c2 ca f0 15 38 ad 50 e2
c2c8 : 03 ed 4d 03 ad 51 03 ed c3
c2d0 : 4e 03 b0 dc 20 9a c2 4c e2
c2d8 : b0 c2 60 a9 02 8d 51 03 0e
c2e0 : 20 ae c2 ad 47 03 f0 03 14
c2e8 : 20 6f c2 20 6f c4 ad 4a dd
c2f0 : 03 8d 3f 03 ad 4b 03 8d 46
c2f8 : 40 03 ad 4c 03 8d 41 03 56
c300 : ad 57 03 d0 7d 38 ad 56 31
c308 : 03 ed 5b 03 b0 74 ad 5b 55
c310 : 03 8d 65 03 8d 68 03 4e 58
c318 : 65 03 38 ad 56 03 ed 65 c3

```

```

c320 : 03 8d 65 03 ad 57 03 e9 19
c328 : 00 8d 66 03 ee 68 03 ad 82
c330 : 67 03 d0 06 20 43 c1 4c c9
c338 : 3d c3 20 46 c1 ad 60 03 39
c340 : d0 05 ee 41 03 d0 03 ce d7
c348 : 41 03 2c 66 03 30 1a ee db
c350 : 3f 03 d0 03 ee 40 03 38 13
c358 : ad 65 03 ed 5b 03 8d 65 05
c360 : 03 ad 66 03 e9 00 8d 66 d5
c368 : 03 18 ad 65 03 6d 56 03 8a
c370 : 8d 65 03 ad 66 03 6d 57 09
c378 : 03 8d 66 03 ce 68 03 d0 1a
c380 : ae 60 ad 56 03 8d 65 03 cd
c388 : 8d 68 03 ad 57 03 8d 66 50
c390 : 03 8d 69 03 4e 66 03 6e 16
c398 : 65 03 38 ad 66 03 ed 65 93
c3a0 : 03 8d 65 03 a9 00 ed 66 43
c3a8 : 03 8d 66 03 ad 67 03 d0 30
c3b0 : 06 20 43 c1 4c ba c3 20 b9
c3b8 : 46 c1 ee 3f 03 d0 03 ee 23
c3c0 : 40 03 2c 66 03 30 20 ad e7
c3c8 : 60 03 d0 05 ee 41 03 d0 25
c3d0 : 03 ce 41 03 38 ad 65 03 78
c3d8 : ed 56 03 8d 65 03 ad 66 55
c3e0 : 03 ed 57 03 8d 66 03 18 58
c3e8 : ad 65 03 6d 5b 03 8d 65 85
c3f0 : 03 ad 66 03 69 00 8d 66 5d
c3f8 : 03 38 ad 68 03 e9 01 8d 2e
c400 : 68 03 b0 03 ce 69 03 2c 13
c408 : 69 03 10 a0 60 38 ad 53 30
c410 : 03 ed 54 03 b0 0e ad 53 58
c418 : 03 48 ad 54 03 8d 53 03 25
c420 : 68 8d 54 03 ad 53 03 8d 61
c428 : 41 03 20 24 c2 ad 53 03 64
c430 : cd 54 03 f0 06 ce 53 03 30
c438 : 4c 24 c4 60 ad 5f 03 85 c0
c440 : ac ad 5e 03 85 ad 20 11 23
c448 : c0 85 fe a5 ae 85 fd a9 a1
c450 : 00 85 fc ad 5a 03 85 fb d3
c458 : 20 25 c0 ad 63 03 d0 08 92
c460 : 18 ad 55 03 65 fd 90 06 99
c468 : 38 ad 55 03 e5 fd 60 38 6d
c470 : ad 4d 03 ed 4a 03 8d 56 e2
c478 : 03 ad 4e 03 ed 4b 03 8d a6
c480 : 57 03 38 ad 50 03 ed 4d 8c
c488 : 03 8d 58 03 38 ad 50 03 01
c490 : ed 4a 03 8d 59 03 a9 00 69
c498 : 8d 60 03 8d 61 03 8d 62 f1
c4a0 : 03 38 ad 4f 03 ed 4c 03 eb
c4a8 : b0 09 ee 60 03 ad 4c 03 79
c4b0 : ed 4f 03 8d 5b 03 38 ad c1
c4b8 : 52 03 ed 4f 03 b0 09 ee a9
c4c0 : 61 03 ad 4f 03 ed 52 03 e7
c4c8 : 8d 5c 03 38 ad 52 03 ed a1
c4d0 : 4c 03 b0 09 ee 62 03 ad 54
c4d8 : 4c 03 ed 52 03 8d 52 03 84
c4e0 : 60 20 ae c2 ad 47 03 f0 57
c4e8 : 03 20 6f c2 20 6f c4 ad 1b
c4f0 : 4a 03 8d 3f 03 ad 4b 03 d8
c4f8 : 8d 40 03 38 ad 3f 03 ed 2a
c500 : 4a 03 8d 5f 03 ad 56 03 18
c508 : f0 53 8d 5a 03 ad 5b 03 62
c510 : 8d 5e 03 ad 60 03 8d 63 5e
c518 : 03 ad 4c 03 8d 55 03 20 35
c520 : 3c c4 8d 53 03 ad 59 03 95
c528 : f0 33 8d 5a 03 ad 52 03 7a
c530 : 8d 5e 03 ad 62 03 8d 63 9e
c538 : 03 20 3c c4 8d 54 03 20 bb
c540 : 0d c4 ad 40 03 cd 4e 03 01
c548 : d0 08 ad 3f 03 cd 40 03 49
c550 : f0 0b ee 3f 03 d0 03 ee 0a
c558 : 40 03 4c fb c4 38 ad 3f f0
c560 : 03 ed 4a 03 8d 5f 03 ad 88
c568 : 59 03 f0 63 8d 5a 03 ad fe
c570 : 5d 03 8d 5e 03 ad 62 03 ab
c578 : 8d 63 03 ad 4c 03 8d 55 eb
c580 : 03 20 3c c4 8d 54 03 38 33
c588 : ad 3f 03 ed 4d 03 8d 5f 35
c590 : 03 ad 58 03 f0 39 8d 5a a4
c598 : 03 ad 5c 03 8d 5e 03 ad 1d
c5a0 : 61 03 8d 63 03 ad 4f 03 33
c5a8 : 8d 55 03 20 3c c4 8d 53 6b
c5b0 : 03 20 0d c4 ad 40 03 cd 24
c5b8 : 51 03 d0 08 ad 3f 03 cd 3c
c5c0 : 50 03 f0 0b ee 3f 03 d0 c6
c5c8 : 03 ee 40 03 ac 5d c5 ad d5
c5d0 : 64 03 f0 15 20 eb c2 20 41
c5d8 : 9a c2 20 eb c2 20 86 c2 26
c5e0 : 20 9a c2 20 86 c2 20 eb d9
c5e8 : c2 60 a9 00 85 fc 4c 25 00
c5f0 : c0 2c 7a 03 10 12 ad 82 13
c5f8 : 03 d0 04 8d 44 03 60 38 64
c600 : a9 00 ed 79 03 8d 79 03 dc
c608 : ad 79 03 85 ac a9 1a 85 6f
c610 : ad 20 11 c0 85 fe a5 ae 6e

```

Listing 8. MSE-Listing der Assembler-Routinen aus Listing 1 bis 5. Bitte mit dem MSE (Seite 8) eingeben.

```

c618 : 85 fd ad 77 03 85 fb 20 83
c620 : ea c5 a5 fd 8d 44 03 60 de
c628 : 2c 83 03 10 2d ad 6c 03 d1
c630 : 48 48 ad 70 03 8d 6c 03 6a
c638 : 68 8d 70 03 ad 6d 03 48 c6
c640 : 48 ad 71 03 8d 6d 03 68 3d
c648 : 8d 71 03 ad 6e 03 48 48 b5
c650 : ad 72 03 8d 6e 03 68 8d 65
c658 : 72 03 20 45 c7 a9 01 8d e5
c660 : 81 03 38 ad 7d 03 cd 6c 27
c668 : 03 90 7d 38 ad 3c 03 ed be
c670 : 6d 03 8d 75 03 ad 3d 03 09
c678 : ed 6e 03 8d 76 03 38 ad cb
c680 : 6a 03 ed 6c 03 8d 7a 03 14
c688 : ad 6b 03 e9 00 8d 40 03 5c
c690 : 38 ad 7f 03 cd 70 03 90 6d
c698 : 23 18 ad 75 03 6d 71 03 49
c6a0 : 8d 79 03 ad 76 03 6d 72 7a
c6a8 : 03 8d 7a 03 20 f1 c5 18 4a
c6b0 : ad 6f 03 6d 70 03 8d 41 5b
c6b8 : 03 20 0f c2 38 ad 80 03 e0
c6c0 : cd 70 03 90 23 38 ad 75 2e
c6c8 : 03 ed 71 03 8d 79 03 ad 8b
c6d0 : 76 03 ed 72 03 8d 7a 03 1e
c6d8 : 20 f1 c5 38 ad 6f 03 ed a7
c6e0 : 70 03 8d 41 03 20 0f c2 50
c6e8 : ad 81 03 f0 34 ce 81 03 fa
c6f0 : 38 ad 7e 03 cd 6c 03 90 6c
c6f8 : 28 18 ad 3c 03 6d 6d 03 77
c700 : 8d 75 03 ad 3c 03 6d 6e 3d
c708 : 03 8d 76 03 18 ad 6a 03 6e
c710 : 6d 6c 03 8d 3f 03 ad 6b bf
c718 : 03 69 00 8d 40 03 4c 90 f0
c720 : c6 2c 83 03 10 1e ad 6e c3
c728 : 03 8d 72 03 68 8d 6e 03 a2
c730 : ad 6d 03 8d 71 03 68 8d f2
c738 : 6d 03 ad 6c 03 8d 70 03 84
c740 : 68 8d 6c 03 60 ad 77 03 42
c748 : 85 ac 20 04 c0 8d 7c 03 1c
c750 : a5 ae 8d 7b 03 ad 6d 03 78
c758 : 85 ac 20 04 c0 38 ad 7b 37
c760 : 03 e5 ae 8d 7b 03 ad 7c 33
c768 : 03 e5 af 8d 7c 03 ad 71 75
c770 : 03 85 ac 20 04 c0 38 ad e7
c778 : 7b 03 e5 ae 85 ac ad 7c 31
c780 : 03 e5 af 85 ad 30 0a 20 d7
c788 : 64 c0 0e 3c 03 2e 3d 03 f4
c790 : 60 a9 00 8d 3c 03 8d 3d 03 d3
c798 : 03 60 ad 8e 03 38 ed 8f d1
c7a0 : 03 4a 8d 8c 03 8d 77 03 3e
c7a8 : 18 6d 8f 03 8d 8d 03 ad 68
c7b0 : 8c 03 85 ac 20 04 c0 a5 25
c7b8 : ae 8d 8a 03 a5 af 8d 8b 55
c7c0 : 03 a9 00 8d 86 03 60 ad a7
c7c8 : 77 03 85 ac 20 04 c0 06 e9
c7d0 : ae 26 af a5 ae 85 ac a5 47
c7d8 : af 85 ad 20 64 c0 4e 3d b9
c7e0 : 03 6e 3c 03 ad 3c 03 8d 6e
c7e8 : 92 03 a9 00 8d 86 03 8d 9a
c7f0 : 6e 03 8d 72 03 ad 77 03 13
c7f8 : 85 ac 20 04 c0 8d 85 03 f1
c800 : a5 ae 8d 84 03 ad 86 03 ae
c808 : 8d 87 03 85 ac 8d 6c 03 b9
c810 : 8d 6d 03 20 04 c0 38 ad 9b
c818 : 84 03 e5 ae 85 ac ad 85 ec
c820 : 03 e5 af 85 ad 20 64 c0 a1
c828 : ad 3c 03 8d 88 03 ad 87 cc
c830 : 03 8d 70 03 8d 71 03 a9 3a
c838 : 00 8d 83 03 20 28 c6 a9 f2
c840 : 80 8d 83 03 20 28 c6 ad 82
c848 : 87 03 cd 88 03 f0 06 ee 83
c850 : 87 03 4c 2e c8 ad 86 03 4c
c858 : cd 92 03 f0 06 ee 86 03 45
c860 : 4c 05 c8 60 a9 00 8d 6d 19
c868 : 03 8d 6e 03 8d 72 03 ad 02
c870 : 77 03 8d 70 03 ad 89 03 a4

```

```

c878 : 8d 6c 03 ad 70 03 8d 71 ea
c880 : 03 20 28 c6 ce 6c 03 10 f3
c888 : 98 ce 70 03 10 e7 60 20 66
c890 : fa c7 a9 00 8d 6e 03 8d ec
c898 : 72 03 ad 86 03 8d 6c 03 1c
c8a0 : 8d 6d 03 85 ac 20 04 c0 b3
c8a8 : 38 ad 8a 03 e5 ae 85 ac fd
c8b0 : ad 8b 03 e5 af 85 ad 20 be
c8b8 : 64 c0 ad 3c 03 8d 89 03 38
c8c0 : 18 6d 8d 03 8d 88 03 a9 cf
c8c8 : 00 8d 87 03 ad 87 03 8d 0f
c8d0 : 70 03 85 ad ad 89 03 85 17
c8d8 : ac 20 11 c0 85 fe a5 ae 35
c8e0 : 85 fd ad 88 03 85 fb 20 6d
c8e8 : ea c5 a5 fd 8d 71 03 20 8f
c8f0 : 28 c6 ad 87 03 cd 88 03 9e
c8f8 : f0 06 ee 87 03 4c cc c8 ef
c900 : ad 86 03 cd 8c 03 f0 06 1c
c908 : ee 86 03 4c 9a c8 60 20 35
c910 : 9a c7 ad 8e 03 85 ac 20 1a
c918 : 04 c0 06 ae 26 af a5 ae a8
c920 : 85 ac a5 af 85 ad 20 64 6a
c928 : c0 4e 3d 03 6e 3c 03 ad ef
c930 : 3c 03 8d 92 03 ad 86 03 6f
c938 : 8d 6c 03 85 ac 20 04 c0 ca
c940 : 8d 91 03 a5 ae 8d 90 03 ab
c948 : ad 8e 03 85 ac 20 04 c0 0b
c950 : 38 a5 ae ed 90 03 85 ac 55
c958 : a5 af ed 91 03 85 ad 20 d6
c960 : 64 c0 ad 3c 03 8d 88 03 dc
c968 : 38 ad 8f 03 ed 86 03 90 fb
c970 : 23 ad 8f 03 85 ac 20 04 f4
c978 : c0 38 a5 ae ed 90 03 85 0e
c980 : ac a5 af ed 91 03 85 ad 4b
c988 : 20 64 c0 ad 3c 03 8d 87 e1
c990 : 03 4c 9a c9 ad 86 03 8d cf
c998 : 87 03 ad 87 03 8d 70 03 61
c9a0 : 85 ac 20 04 c0 18 a5 ae c5
c9a8 : 6d 90 03 85 ac a5 af 6d 60
c9b0 : 91 03 85 ad 20 64 c0 ad 5d
c9b8 : 3c 03 8d 89 03 85 fb ad b2
c9c0 : 86 03 85 ad ad 8d 03 85 3d
c9c8 : ac 20 11 c0 85 fe a5 ae 25
c9d0 : 85 fd 20 ea c5 38 ad 86 9b
c9d8 : 03 e5 fd 8d 6d 03 a9 00 94
c9e0 : e5 fe 8d 6e 03 ad 87 03 37
c9e8 : 85 ad ad 8d 03 85 ac 20 b0
c9f0 : 11 c0 85 fe a5 ae 85 fd 84
c9f8 : ad 89 03 85 fb 20 ea c5 d3
ca00 : 38 ad 87 03 e5 fd 8d 71 b8
ca08 : 03 a9 00 8d 83 03 e5 fe 77
ca10 : 8d 72 03 20 28 c6 a9 80 fc
ca18 : 8d 83 03 20 28 c6 ad 87 aa
ca20 : 03 cd 88 03 f0 06 ee 87 96
ca28 : 03 4c 9a c9 ad 86 03 cd e8
ca30 : 92 03 f0 06 ee 86 03 4c 08
ca38 : 35 c9 60 20 9a c7 ad 86 1a
ca40 : 03 8d 6c 03 85 ac 38 a9 77
ca48 : 00 ed 86 03 8d 6d 03 a9 e4
ca50 : 00 e9 00 8d 6e 03 20 04 7e
ca58 : c0 38 ad 8a 03 e5 ae 85 16
ca60 : ac ad 8b 03 e5 af 85 ad 73
ca68 : 20 64 c0 38 ad 8d 03 ed 21
ca70 : 3c 03 8d 88 03 a9 00 8d 3b
ca78 : 87 03 ad 87 03 8d 70 03 41
ca80 : 85 ad ad 8d 03 85 ac 20 48
ca88 : 11 c0 85 fe a5 ae 85 fd 1c
ca90 : ad 88 03 85 fb 20 ea c5 eb
ca98 : a5 fd 38 ed 87 03 8d 71 b1
caa0 : 03 a5 fe e9 00 8d 72 03 af
caa8 : 20 28 c6 ad 87 03 cd 88 1c
cab0 : 03 f0 06 ee 87 03 4c 7a 41
cab8 : ca ad 86 03 cd 8c 03 f0 8a
cac0 : 06 ee 86 03 4c 3e ca 60 e2
cac8 : 20 fd ae 20 9e ad 20 aa c4
cad0 : b1 60 a9 2c a0 00 84 fb b5

```

```

cad8 : d1 7a d0 03 4c 73 00 a0 1c
cae0 : 80 84 fb 60 20 d2 ca 24 b9
cae8 : fb 30 0f 20 9e ad 20 aa f0
caf0 : b1 8c 8f 03 20 c8 ca 8c b8
caf8 : 8e 03 60 20 c8 ca 8c 6a 0e
cb00 : 03 8d 6b 03 20 c8 ca 8c 92
cb08 : 6f 03 60 20 d2 ca 24 fb 21
cb10 : 30 07 20 9e ad 20 aa b1 8a
cb18 : 2c a0 00 8c 35 c1 4c 2c 11
cb20 : c1 20 d2 ca 24 fb 30 07 f0
cb28 : 20 9e ad 20 aa b1 2c 00 31
cb30 : 01 8c 19 c1 4c 18 c1 a9 d6
cb38 : 00 2c a9 80 8d 3e 03 20 e0
cb40 : c8 ca 8c 3f 03 8d 40 03 1c
cb48 : 20 c8 ca 8c 41 03 4c 4b 04
cb50 : c1 a9 00 2c a9 80 8d 93 67
cb58 : 03 20 c8 ca 8c 4a 03 8d 39
cb60 : 4b 03 20 c8 ca 8c 4c 03 96
cb68 : 20 c8 ca 8c 4d 03 8d 4e f0
cb70 : 03 20 c8 ca 8c 4f 03 2c b6
cb78 : 20 03 10 18 20 c8 ca 8c 21
cb80 : 50 03 8d 51 03 20 c8 ca c9
cb88 : 8c 52 03 20 c8 ca 8c 44 a0
cb90 : c8 ca 8c 3f 03 8d c2 20 b9
cb98 : fb 20 20 d2 ca 24 fb 30 79
cba0 : 09 20 9e ad 20 aa b1 8c 4e
cba8 : 77 03 4c c7 c7 20 fb ca b0
cbb0 : 20 e4 ca 4c 0f c9 a9 80 65
cbb8 : 2c a9 00 8d 83 03 20 fb 33
cbc0 : ca 20 d2 ca 24 fb 30 0f a9
cbc8 : 20 9e ad 20 aa b1 8c 77 00
cbd0 : 03 20 c8 ca 8c 89 03 4c 28
cbd8 : 64 c8 a9 80 2c a9 00 8d 46
cbe0 : 83 03 20 fb ca 20 e4 ca 43
cbe8 : 4c 8f c8 a9 80 2c a9 00 73
cbf0 : 8d 83 03 20 fb ca 20 e4 64
cbf8 : ca 4c 3b ca ff 00 ff 00 10

```

Listing 8. MSE-Listing der Assembler-Routinen aus Listing 1 bis 5 (Schluß)

```

programm : sort/write          cf59 cfff
-----
cf59 : a0 ff c8 98 91 fb c5 8c 67
cf61 : d0 f8 85 ad a5 ad 85 ac fc
cf69 : a2 00 86 61 86 ae 86 64 9a
cf71 : 8a a8 b1 fb 0a 90 04 c6 fe
cf79 : 61 e6 fe a8 b1 fd 48 c8 e0
cf81 : b1 fd 24 61 10 04 e6 61 e6
cf89 : c6 fe e4 64 d0 08 85 62 bc
cf91 : 68 85 63 e8 d0 da c5 62 71
cf99 : 68 e5 63 50 02 49 80 10 63
cfa1 : 13 8a a8 86 ad b1 fb 48 dd
cfa9 : 88 b1 fb c8 91 fb 68 88 cd
cfb1 : 91 fb e6 ae e4 ac d0 b6 34
cfb9 : a5 ae d0 a8 a6 8c e8 ca 06
cfc1 : 8a a8 b1 fb 48 8a 0a 09 9e
cfc9 : 01 90 04 e6 61 e6 fc a8 83
cfd1 : 68 91 fb a9 00 88 91 fb b8
cfd9 : a5 61 f0 04 c6 61 c6 fc 78
cfe1 : 8a d0 dc 60 a5 01 29 fe 1b
cfe9 : 85 01 a0 07 b1 fd 31 fb c0
cff1 : 91 fb 88 10 f7 a5 01 09 67
cff9 : 01 85 01 60 00 ff 00 00 09

```

Listing 9. MSE-Listing der Assembler-Routinen aus Listing 6 und 7. Bitte mit dem MSE (Seite 8) eingeben.

```

10 REM          SHAPES DEMO          <146>
20 :                                <252>
30 REM          RICHARD L. RYLANDER   11/23/84
   (REVISED 1/20/85 TO ADD LABELING) <130>
40 :                                <016>
50 GR=49378      :REM GRAPHIK MODUS  <165>
60 TX=49411      :REM TEXT MODUS     <103>
70 :                                <046>
80 LB=893        :REM LINKE GRENZE   <252>
90 RB=894        :REM RECHTE GRENZE  <206>
100 UB=895       :REM OBERE GRENZE   <090>
110 DB=896       :REM UNTERE GRENZE  <046>
120 :                                <096>
130 REM          FLAGS FUER VERSCHIEDENE GRAFIKMO
   DI                                <107>
140 :                                <116>
150 SH=838       :REM SCHATTIERUNG: 0=RANDOM
   , 1=HALBTON   <252>
160 SC=839       :REM SKALIERUNG:0=NORMAL (1
   :1), 1=SKAL. (4:3) FUER DARSTELLUNG <024>

```

```

170 LT=898      :REM LICHTEIFALL: 0=NORMAL
   , 1=VON HINTEN BELEUCHTET <033>
180 :                                <156>
190 BO=53280     :REM RAHMENFARBE    <186>
200 :                                <176>
210 REM          FUNKTIONSADRESSEN   <104>
220 :                                <196>
230 CL=51979     :REM BITMAP BEREICH LOESCHE
   N                                <221>
240 CO=52001     :REM COLORBEREICH FUELLEN <013>
250 :                                <226>
260 SP=52119     :REM SPHERE          <160>
270 TR=52141     :REM RING VON OBEN   <108>
280 VC=52150     :REM ZYLINDER (VERTIKALE A
   CHSE)                                <011>
290 HC=52153     :REM ZYLINDER (HORIZONTALE
   ACHSE)                                <167>

```

Listing 10. Demo-Programm »Shapes«

```

300 VT=52186 :REM RINGSCHNITT (VERTIKAL
    E ACHSE) <216>
310 HT=52189 :REM RINGSCHNITT (AXIS HOR
    IZONTALE ACHSE) <199>
320 VS=52203 :REM SPULE (VERTIKALE ACHS
    E) <241>
330 HS=52206 :REM SPULE (HORIZONTALER AC
    HSE) <064>
340 : <062>
350 REM EINZELNE SHAPES ZEICHNEN
360 : <082>
370 POKE SH,1 :REM HALBTONSCHATTIERUNG
380 POKE SC,1 :REM SKALIERUNG <121>
390 POKE LT,0 :REM NORMALBELEUCHTUNG
400 SYS(CL) :REM BILDSCHIRM LOESCHEN <077>
410 SYS(CO),17*11+1 :REM FARBKOMBINATION
    - DUNKELGRAU (11) WEISSE PUNKTE (1) <245>
414 REM AUF DEN MEISTEN FARBMONITOREN <150>
420 POKE BO,1 :REM WEISSER RAHMEN <041>
430 SYS(GR) :REM GRAFIK EINSCHALTEN <023>
432 RW=12:CM=14:MD=1:A$="SHAPES DEMO":GOSUB
    B 1900:REM TITEL <207>
434 X1=110:Y1=120:X2=210:Y2=120:BC=1:DC=7:
    GOSUB 1700:REM COLOR TITEL <190>
440 POKE LB,38:POKE RB,38:POKE UB,38:POKE
    DB,38 :REM SHAPE FENSTER <107>
450 SYS(SP),40,199,38 <129>
455 RW=9:CM=2:MD=1:A$="KUGEL":GOSUB 1900 <102>
460 SYS(HC),120,199,38,38 <152>
465 CM=11:A$="H-ZYLIND.":GOSUB 1900 <250>
470 SYS(VC),200,199 :REM VOREINGESTELLTE P
    ARAMETER WERDEN BENUTZT <229>
475 CM=21:A$="V-ZYLIND.":GOSUB 1900 <117>
480 SYS(TR),280,199,15,38 <084>
485 CM=32:A$="(2SPACE)RING":GOSUB 1900 <122>
490 SYS(VT),40,64 <046>
495 RW=23:CM=1:A$="H-RING":GOSUB 1900 <205>
500 SYS(HT),120,64 <190>
505 CM=11:A$="V-RING":GOSUB 1900 <155>
510 SYS(HS),200,64,5,100 <025>
515 CM=22:A$="H-SPULE":GOSUB 1900 <000>
520 SYS(VS),280,64 <162>
525 CM=32:A$="V-SPULE":GOSUB 1900 <011>
530 POKE 198,0:WAIT 198,1:POKE 198,0 <125>
540 REM AUF TASTE WARTEN <103>
550 : <018>
560 REM ZWEI "POKALE", EINER MIT HALBTON-
    , DER ANDERE MIT "RANDOM"-SCHATTIERUNG <170>
570 : <038>
580 SYS(CL):SYS(CO),16*11+1 <084>
582 RW=14:CM=14:A$="VERGLEICH":GOSUB 1900:
    RW=15:CM=16:A$="ZWEIER":GOSUB 1900 <228>
584 RW=16:CM=12:A$="SCHATTIERUNGEN":GOSUB
    1900:RW=18:CM=14:A$="HALBTON" <162>
586 GOSUB 1900:RW=20:CM=15:A$="RANDOM -->"
    :GOSUB 1900 <217>
590 POKE LB,255:POKE RB,255:POKE UB,49:POK
    E DB,255:REM AN DEN OBEREN RAND <229>
600 SYS(SP),80,190,80 <119>
610 POKE UB,51:POKE DB,51:REM WEITERN SPR
    ITE-TEIL ANHAENGEN <217>
620 SYS(VS),80,69,10,130 <170>
630 POKE DB,9:POKE UB,8 <060>
640 SYS(VT),80,9,25,45 <168>
650 POKE SH,0 :REM AUF RANDOM SCHATTIERUN
    G UMSCHALTEN <228>
660 POKE LB,255:POKE RB,255:POKE UB,49:POK
    E DB,255 <013>
670 SYS(SP),240,190,80 <171>
680 POKE UB,51:POKE DB,51 <130>
690 SYS(VS),240,69,10,130 <112>
700 POKE DB,9:POKE UB,8 <130>
710 SYS(VT),240,9,25,45 <053>
720 POKE 198,0:WAIT 198,1:POKE 198,0 <059>
740 : <208>
750 REM ZEICHNE "WEIN" SZENE <144>
760 POKE LT,1 :REM HINTERGRUNDBELEUCHTUNG
770 POKE SH,1 :REM HALBTONSCHATTIERUNG FUE
    R FLASCHENETIKETT <070>
780 SYS(CO):SYS(CL),255:REM BITMAP FUELLE
    N <239>
790 POKE BO,0 :REM SCHWARZER HINTERGRUND <021>
792 RW=0:CM=0:MD=2:A$="ZEICHNEN MIT":GOSUB
    B 1900:RW=1:CM=0:A$="RUECKBELEUCHTUNG" <066>
794 GOSUB 1900:RW=2:CM=0:A$="GEGEN GESETZT
    EN":GOSUB 1900 <050>
796 RW=3:CM=0:A$="HINTERGRUND":GOSUB 1900 <034>
798 RW=1:CM=26:A$="FARBEN WERDEN":GOSUB 19
    00 <169>
    <210>

```

```

800 RW=2:CM=26:A$="NACHGEZEICHNET":GOSUB 1
    900 <060>
810 REM ZEICHNE FLASCHE <187>
820 POKE UB,0:POKE DB,255:POKE LB,255:POKE
    RB,255 <087>
830 SYS(VT),150,10,30,50 <164>
840 POKE UB,255:SYS(VC),150,70,50,60 <113>
850 POKE DB,0:SYS(VT),150,130,6,50 <053>
860 POKE DB,55:POKE UB,0:SYS(VS),150,204,1
    5,181 <063>
870 POKE UB,255:SYS(VC),150,221,16,17 <164>
880 : <094>
890 REM ZEICHNE WEINGLAS <127>
900 POKE UB,20:SYS(SP),80,120,60 <235>
910 POKE UB,35:POKE DB,34:SYS(VS),80,34,10
    ,110 <183>
920 : <134>
930 REM ZEICHNE EIN PAAR TRAUBEN <156>
940 SYS(SP),8,8,8: <167>
950 SYS(SP),20,8:SYS(SP),40,8:SYS(SP),12,2
    0:SYS(SP),30,20:SYS(SP),25,16 <139>
960 : <174>
970 REM ZEICHNET APFEL BESTEHEND AUS 2 RIN
    GEN UND EINEM KUGELFRAGMENT <133>
980 POKE UB,255:POKE DB,255:POKE LB,255:PO
    KE RB,59 <012>
990 SYS(VT),260,29,0,50:SYS(VT),260,79 <217>
1000 POKE UB,43:POKE DB,43:SYS(SP),260,54,
    60 <014>
1010 REM APFELSTIEL ZEICHNEN <236>
1020 POKE RB,0:POKE DB,0:SYS(TR),272,104,1
    0,15 <173>
1030 REM BLATT ZEICHNEN <113>
1040 POKE DB,255:POKE RB,0:SYS(SP),256,119
    ,15 <165>
1050 REM "RANDOM"-SCHATTIERTES ETIKETT H
    INZUFUEGEN ! <191>
1060 POKE UB,255:POKE RB,255:POKE LB,6 <010>
1070 POKE SH,0:SYS(VC),150,72,50,48 <225>
1080 : <040>
1090 REM BILD NACHCOLORIEREN <148>
1100 SYS(CO),12 <033>
1110 X1=200:Y1=1:X2=315:Y2=100:DC=0:BC=2:G
    OSUB 1700 <145>
1120 X1=240:Y1=110:X2=255:Y2=150:BC=5:GOSU
    B 1700 <229>
1130 X1=260:Y1=110:X2=270:Y2=135:BC=9:GOSU
    B 1700 <055>
1140 X1=1:Y1=1:X2=48:Y2=30:BC=4:GOSUB 1700
    <007>
1150 X1=140:Y1=205:X2=180:Y2=235:BC=7:GOSU
    B 1700 <228>
1160 X1=145:Y1=25:X2=195:Y2=115:BC=6:GOSUB
    1700 <234>
1170 POKE 198,0:WAIT 198,1:POKE 198,0 <001>
1190 : <150>
1200 REM "KAFFEE UND PLAETZCHEN" <110>
1210 POKE SH,0 :REM RANDOM-SCHATTIERUNG FU
    ER PLAETZCHEN <230>
1220 SYS(CO),16*11+1:SYS(CL):POKE BO,1 <084>
1230 POKE LB,255:POKE RB,255:POKE UB,255:P
    OKE DB,255:REM KEIN SHAPE-FENSTER <040>
1240 SYS(VT),60,20,20,60 <069>
1250 POKE RB,29:SYS(VT),99,60:POKE RB,255 <198>
1260 SYS(TR),188,180 <166>
1270 REM HALBTONSCHATTIERTE KAFFEETASSE HI
    NZUFUEGEN. <239>
1280 POKE SH,1:POKE UB,0:SYS(VT),188,20:PO
    KE UB,255 <251>
1290 POKE DB,0:SYS(TR),278,110,20,40 <210>
1300 POKE RB,255:POKE UB,0:POKE LB,0 <140>
1310 SYS(TR),248,90,50,70:POKE LB,255:SYS(
    SP),248,110,10:POKE UB,255 <160>
1320 SYS(VC),308,100,10,10 <246>
1330 SYS(VC),188,77,60,57 <154>
1340 POKE DB,0:SYS(VT),188,134,40,60 <050>
1360 SYS(CO),1+16*9 :REM 1=WEISSER HINTERG
    RUND, 9=BRAUNE PUNKTE <067>
1370 X1=130:Y1=1:X2=319:Y2=136:BC=1:DC=5:G
    OSUB 1700 <009>
1380 X1=250:Y1=144:X2=319:Y2=144:GOSUB 170
    0 <161>
1390 POKE 198,0:WAIT 198,1:POKE 198,0 <223>
1410 : <116>
1420 REM ZEICHNE "VERBUNDENE" RINGE DURCH
    UEBERLAPPUNG DER FENSTER <138>
1430 POKE LT,0 :REM BLAUE PUNKTE AUF WEISS
    , KEINE HINTERGRUNDBELEUCHTUNG <041>
1440 SYS(CL):SYS(CO),1+16*6:REM 1=WEISSER

```

Listing 10. Demo-Programm »Shapes« (Fortsetzung)

```

INTERGRUND, 6=BLAUE PUNKTE <255>
1450 POKE UB,255:POKE DB,255:POKE LB,255:P <058>
    OKE RB,255: REM KEINE FENSTER
1460 POKE SH,0 :REM ZUFÄLLIGE SCHATTIERU <096>
    NG
1470 SYS(TR),244,84,48,70 <088>
1480 SYS(TR),160,84:SYS(TR),76,84 <109>
1490 SYS(TR),118,156:SYS(TR),202,156 <163>
1500 REM UEBERLAPPENDE ZONEN HINZUFUEGEN <129>
1510 POKE RB,0:POKE DB,0:SYS(TR),160,84:PO <127>
    KE RB,255:POKE LB,0
1520 SYS(TR),76,84:POKE DB,255:POKE UB,0:S <198>
    YS(TR),118,156
1530 POKE LB,255:POKE RB,0:SYS(TR),202,156 <081>
    :POKE LB,27
1540 POKE DB,0:POKE UB,255:SYS(TR),160,84 <152>
1550 POKE DB,0:POKE UB,255:SYS(TR),160,84 <162>
1560 POKE LB,255:POKE UB,27:SYS(TR),244,84 <171>
1570 POKE LB,0:POKE RB,27:POKE UB,255:SYS( <180>
    TR),244,84
1580 POKE 198,0 <220>
1590 GET A$:IF A$= "" THEN 1590 <184>
1600 SYS(TX):POKE BO,14:REM RUECKKEHR ZUM <239>
    TEXTMODUS
1610 END <088>
1620 : <072>
1630 REM UNTERPROGRAMM ZU NACHCOLORIEREN V <233>
    ERSCHIEDENER ZONEN
1640 REM ACHTUNG: ES KANN JEWEILS NUR EINE <232>
    RECHTECKIGE' FLAECHE
1650 REM VON 8X8 PUNKTEN EINGEFAERBT WERDE <088>
    N !!
1660 REM (X1,Y1)=UNTERER LINKER PUNKT, (X2 <235>
    ,Y2)=OBERER RECHTER PUNKT
1690 REM DIE Y-KOORDINATEN MUESSEN "UNSKAL <223>
    LIERT" SEIN. FALLS SCALE-FLAG GESETZT
1700 IF PEEK(SC) THEN Y1=(Y1+1)*213/256:Y2= <168>
    (Y2+1)*213/256
1710 REM UEBERGABEVARIABLE IST CC. <236>
1720 REM CC=16*DC + BC [DC=PUNKT FARBE, BC <037>
    = HINTERGRUNDFARBE]
1730 CC=16*DC+BC <227>
1740 FOR IX=INT(X1/8) TO INT(X2/8) <033>
1750 FOR IY=INT(Y1/8) TO INT(Y2/8) <061>
1760 POKE 34752+IX-40*IY,CC <092>
1770 NEXT:NEXT:RETURN <227>
1780 : <232>
1790 REM "TEXT"-UNTERROUTINE ERMOEGLICHT E <076>
    INFUEGEN VON TEXT IN DIE BILDER.
1800 REM "RW" UND "CM" SIND DIE ZEILEN(0-2 <040>
    4) UND SPALTEN(0-39) KOORDINATEN DES
1810 REM ERSTEN BUCHSTABEN DES ZU DRUCKEND <249>
    EN TEXTSTRINGS.
1820 REM UEBERGABEVARIABLE FUER TEXT IST A <073>
    $
1830 REM "MD" ZEIGT DEN DRUCKMODUS AN: <133>
1840 REM 1 - NORMAL ("SCHWARZE" BUCHSTABE <206>
    N AUF "WEISSEM" HINTERGRUND)
1850 REM 2 - INVERSE ("WEISSE" BUCHSTABEN <103>
    AUF "SCHWARZEM" HINTERGRUND)
1860 REM 3 - BUCHSTABEN WERDEN MIT LOGISC <191>
    HEM ODER MIT HINTERGRUND VERKNUEPFT.
1870 REM 4 - BUCHSTABEN WERDEN MIT LOGISC <070>
    HEM UND MIT HINTERGRUND VERKNUEPFT.
1880 REM 5 - BUCHSTABEN WERDEN MIT LOGISC <111>
    HEM EXOR MIT HINTERGRUND VERKNUEPFT.
1890 : <088>
1900 SB=40952:TB=54272:IF (MD AND 1) THEN TB <044>
    =53248:REM BASISADRESSEN
1910 OS=320*RW+8*CM:REM ADRESSE DES ZEICH <111>
    ENGEGENERATORS
1920 POKE 56334,PEEK(56334)AND 254:REM DIS <136>
    ABLE IRQ TIMER
1930 POKE 1,PEEK(1)AND 251:REM CHARACTER R <066>
    OM EINSCHALTEN
1940 L=LEN(A$):FOR N=1 TO L:N8=N*8+OS+SB <068>
1950 X=ASC(MID$(A$,N,1)):IF X>63 THEN X=X- <107>
    64
1960 TC=TB+8*X <173>
1970 ON MD GOTO 1980,1980,1990,2000,2010 <120>
1980 POKE 53231,36:GOTO 2020 <028>
1990 POKE 53231,17:GOTO 2020 <038>
2000 POKE 53231,49:GOTO 2020 <062>
2010 POKE 53231,81 <044>
2020 POKE 252,N8/256:POKE 251,N8-256*INT(N <137>
    8/256)
2030 POKE 254,TC/256:POKE 253,TC-256*INT(T <136>
    C/256)

```

```

2040 SYS(53221):NEXT <158>
2050 POKE 1,PEEK(1)OR 4:POKE 56334,PEEK(56 <136>
    334)OR 1:REM NORMAL SCHALTEN
2060 RETURN <086>

```

© 64'er

Listing 10. Demo-Programm »Shapes« (Schluß)

```

10 REM "STELLATION" <061>
20 REM BY RICHARD L. RYLANDER 12/5/84 <185>
30 : <006>
40 GR=49378 :REM GRAPHIK MODUS <155>
50 TX=49411 :REM TEXT MODUS <093>
60 BO=53280 :REM RAHMENFARBE <056>
70 : <046>
80 REM STYLE PARAMETERS <157>
90 POKE 839,1 :REM SKALIERUNG (3:4) <047>
100 POKE 871,0 :REM FACETTENRAENDER (0=NAC <086>
    HZEICHNEN, 1=LOESCHEN)
110 SH=838 :REM SCHATTIERUNG (0=RANDOM <162>
    , 1=HALBTON)
120 EG=868 :REM RANDFLAG (0=NORMAL, 1= <058>
    LINIEN AM RAND HINZUFUEGEN)
130 : <106>
140 REM FUNKTIONS ADRESSEN <034>
150 CL=51979 :REM BITMAP LOESCHEN <242>
160 CO=52001 :REM COLOR MAP FUELLEN <112>
170 FC=52052 :REM SCHATTIERTE FACETTEN Z <124>
    EICHNEN
180 KS=53081 :REM SORTIERROUTINE <231>
190 : <166>
200 XC=160:YC=120 :REM ZENTRIERUNGSKOORDI <164>
    NATEN
210 : <186>
220 PRINT "{CLR}*****" <253>
    *****
230 PRINT"* KLEINER STERNFOERMIGER DODEKAH <133>
    EDRON *"
240 PRINT"*****" <024>
    *****
250 PRINT "{DOWN,SPACE}SCHATTIERUNG:" <037>
260 PRINT" R=RANDOM, H=HALBTON" <001>
270 INPUT "{DOWN,SPACE}IHRE WAHL {5SPACE}H{3 <001>
    LEFT}";A$
280 POKE SH,0:IF A$="H" THEN POKE SH,1 <206>
290 PRINT "{DOWN,SPACE}RAND-STIL : " <230>
300 PRINT "{DOWN,SPACE}N - NORMAL":PRINT "{D <196>
    OWN,SPACE}E - ECKENBETONUNG"
310 PRINT "{DOWN,SPACE}W - DRAHTGITTER" <251>
320 INPUT "{DOWN,SPACE}IHRE WAHL {5SPACE}N{3 <243>
    LEFT}";A$
330 POKE EG,0:WI=0:IF A$="N" THEN 360 <101>
340 POKE EG,1:IF A$="W" THEN WI=-1 <252>
350 : <072>
360 PRINT "{DOWN,SPACE}LESE SCHEITEL DATEN" <003>
370 VN=32:DIM P%(VN-1,2) <088>
380 FOR N=0 TO VN-1:READ P%(N,0),P%(N,1),P <242>
    %(N,2):NEXT
390 : <112>
400 PRINT "{DOWN,SPACE}GEBEN SIE X, Y, UND <080>
    Z WINKEL EIN"
410 PRINT "{7SPACE}(WINKEL IN GRAD)" <102>
420 INPUT X,Y,Z <163>
430 J=3.14159265/180:X=X*J:Y=Y*J:Z=Z*J <104>
440 X0=COS(Y)*COS(Z)-SIN(X)*SIN(Y)*SIN(Z): <148>
    X1=COS(Y)*SIN(Z)+SIN(X)*SIN(Y)*COS(Z)
450 X2=-COS(X)*SIN(Y):Y0=-COS(X)*SIN(Z):Y1 <074>
    =COS(X)*COS(A):Y2=SIN(X)
460 Z0=SIN(Y)*COS(Z)+SIN(X)*COS(Y)*SIN(Z) <138>
470 Z1=SIN(Y)*SIN(Z)-SIN(X)*COS(Y)*COS(Z): <250>
    Z2=COS(X)*COS(Y)
480 PRINT "{DOWN,SPACE}ROTATIONSBERECHNUNG" <202>
490 FOR N=0 TO VN-1 <082>
500 X=P%(N,0):Y=P%(N,1):Z=P%(N,2) <027>
510 P%(N,0)=X0*X+X1*Y+X2*Z:P%(N,1)=Y0*X+Y1 <136>
    *Y+Y2*Z:P%(N,2)=Z0*X+Z1*Y+Z2*Z:NEXT
520 : <244>
530 FA=60:REM ANZAHL DER FACETTEN <201>
540 DIM F%(FA/2,2),SH(FA/2),Z%(FA/2),K%(FA <099>
    /2)
550 PRINT" LESE VERBINDUNGSDATEN{14SPACE}" <089>
560 VF=-1:REM VF = ANZAHL DER SICHTBAREN <057>
    FACETTEN
570 FOR N=1 TO FA <148>
580 VF=VF+1 <223>
590 FOR I=0 TO 2:READ F%(VF,I):NEXT <042>

```

Listing 11. Demo-Programm »Stellation«

```

600 REM BERECHNUNG DER NORMALENVEKTOREN <118>
610 Z=(P%(F%(VF,2),0)-P%(F%(VF,1),0))*(P%(F%(VF,0),1)-P%(F%(VF,1),1)) <150>
620 Z=Z-(P%(F%(VF,0),0)-P%(F%(VF,1),0))*(P%(F%(VF,2),1)-P%(F%(VF,1),1)) <141>
630 IF Z=<0 THEN 720:REM FACETTE NICHT SICH BAR <009>
640 X=(P%(F%(VF,2),1)-P%(F%(VF,1),1))*(P%(F%(VF,0),2)-P%(F%(VF,1),2)) <178>
650 X=X-(P%(F%(VF,0),1)-P%(F%(VF,1),1))*(P%(F%(VF,2),2)-P%(F%(VF,1),2)) <155>
660 Y=(P%(F%(VF,2),2)-P%(F%(VF,1),2))*(P%(F%(VF,0),0)-P%(F%(VF,1),0)) <199>
670 Y=Y-(P%(F%(VF,0),2)-P%(F%(VF,1),2))*(P%(F%(VF,2),0)-P%(F%(VF,1),0)) <183>
680 NC=SQR(X*X+Y*Y+Z*Z):REM LAENGE DES NORMALENVEKTORES <068>
690 SH(VF)=26*(2*Z+X+Y)/NC <114>
700 SH(VF)=(SH(VF)+64)*(SH(VF)+64)/256:REM SCHATTIERUNG <177>
710 GOTO 730 <226>
720 VF=VF-1 <235>
730 NEXT <232>
740 : <208>
750 PRINT" BERECHNUNG DER DARSTELLUNGSGROESSE" <140>
760 Y=0:FOR N=0 TO VN-1:IF ABS(P%(N,1))>Y THEN Y=ABS(P%(N,1)) <190>
770 NEXT:S=119/Y <075>
780 FOR N=0 TO VN-1:P%(N,1)=S*P%(N,1)+YC:P%(N,0)=S*P%(N,0)+XC:NEXT <130>
790 : <004>
800 : <019>
810 FOR N=0 TO VF <115>
820 Z%(N)=(P%(F%(N,0),2)+P%(F%(N,1),2)+P%(F%(N,2),2))/3:NEXT <060>
830 : <044>
840 PRINT" SORTIERUNG DER FACETTEN" <010>
850 POKE 140,VF <004>
860 K%(0)=K%(0):POKE 251,PEEK(71):POKE 252,PEEK(72) <154>
870 Z%(0)=Z%(0):POKE 253,PEEK(71):POKE 254,PEEK(72) <019>
880 SYS(KS) <153>
890 : <104>
900 REM ZEICHNE FACETTEN <002>
910 SYS(GR):SYS(CD):SYS(CL):POKE 80,1 <168>
920 FOR N=0 TO VF:FA=K%(N) <009>
930 IF WI THEN SH(FA)=64 <016>
940 X0=P%(F%(FA,0),0):Y0=P%(F%(FA,0),1):X1=P%(F%(FA,1),0):Y1=P%(F%(FA,1),1) <170>
950 X2=P%(F%(FA,2),0):Y2=P%(F%(FA,2),1) <193>
960 SYS(FC),X0,Y0,X1,Y1,X2,Y2,SH(FA) <225>

```

```

970 NEXT <218>
980 POKE 198,0 <126>
990 GET A$:IF A$="" THEN 990 <168>
1000 SYS(TX):POKE 80,14:END <128>
1010 : <224>
1020 REM SCHEITELDATEN <083>
1030 DATA 1000,618,0,1000,-618,0,-1000,618,0,-1000,-618,0 <092>
1040 DATA 0,1000,618,0,1000,-618,0,-1000,618,0,-1000,-618 <099>
1050 DATA 618,0,1000,-618,0,1000,618,0,-1000,-618,0,-1000 <183>
1060 DATA 618,0,236,618,0,-236,-618,0,236,-618,0,-236 <043>
1070 DATA 236,618,0,-236,618,0,236,-618,0,-236,-618,0 <243>
1080 DATA 0,236,618,0,-236,618,0,236,-618,0,-236,-618 <100>
1090 DATA 382,382,382,-382,382,-382,382,-382,382,-382,-382 <236>
1100 DATA -382,382,382,-382,382,-382,-382,382,382,-382,-382 <056>
1110 : <070>
1120 REM VERBINDUNGSDATEN <109>
1130 DATA 0,12,13,0,13,25,0,25,16,0,16,24,0,24,12 <057>
1140 DATA 1,12,26,1,26,18,1,18,27,1,27,13,1,13,12 <248>
1150 DATA 2,15,14,2,14,28,2,28,17,2,17,29,2,29,15 <251>
1160 DATA 3,14,15,3,15,31,3,31,19,3,19,30,3,30,14 <208>
1170 DATA 4,16,17,4,17,28,4,28,20,4,20,24,4,24,16 <110>
1180 DATA 5,17,16,5,16,25,5,25,22,5,22,29,5,29,17 <168>
1190 DATA 6,19,18,6,18,26,6,26,21,6,21,30,6,30,19 <177>
1200 DATA 7,18,19,7,19,31,7,31,23,7,23,27,7,27,18 <235>
1210 DATA 8,20,21,8,21,26,8,26,12,8,12,24,8,24,20 <060>
1220 DATA 9,21,20,9,20,28,9,28,14,9,14,30,9,30,21 <144>
1230 DATA 10,23,22,10,22,25,10,25,13,10,13,27,10,27,23 <074>
1240 DATA 11,22,23,11,23,31,11,31,15,11,15,29,11,29,22 <224>

```

© 64'er

Listing 11. Demo-Programm »Stellation« (Schluß)

