

Von der Ordnung zum Chaos - Apfelmännchen auf dem Atari

Faszinierende Grafiken von nahezu unendlicher Vielfalt erzeugt das Programm »Apfelmännchen« auf dem Atari 800XL in Turbo-Basic.

Grundlage dieses Programms ist ein Teilgebiet der Mathematik von großer Komplexität. Es handelt sich um die Berechnung und Betrachtung der sogenannten Mandelbrot-Menge und speziell deren Grenzsichten. Diese Menge, benannt nach dem Wissenschaftler Benoit B. Mandelbrot, besteht aus einer Folge von komplexen Zahlen, die sich recht merkwürdig verhalten. Trägt man die beiden Komponenten jeder komplexen Zahl, die sich aus diesem Algorithmus ergibt, auf die Abszisse und die Ordinate eines Koordinatensystems auf, so erhält man Grafiken, die kuriose Eigenschaften aufweisen.

Um den grundlegenden Algorithmus zu verstehen, muß man sich etwas näher mit den Eigenschaften komplexer Zahlen auseinandersetzen. Eine komplexe Zahl besteht immer aus zwei Komponenten, dem Realteil und dem Imaginärteil. Die Schreibweise erfolgt in der Form: $C = a + bi$. Die komplexe Zahl C ergibt sich also aus dem Realteil a und dem Imaginärteil b . Der Buchstabe i steht für eine Konstante und unterliegt folgender Definition: $i^2 = -1$. Die Rechenregeln der komplexen Zahlen weichen jedoch von denen der reellen Zahlen ab. Geht man bei der Addition noch in gewohnter Weise vor und addiert Real- und Imaginärteil einfach separat, ist das Multiplizieren schon etwas komplizierter. So ergibt sich $(a+bi) \cdot (c+di)$ zu $(a \cdot c - b \cdot d) + (a \cdot d + b \cdot c) \cdot i$. Wer diese Formel genauer betrachtet, wird feststellen, daß sich i^2 aufgrund seiner Definition zu -1 aufgelöst hat.

Beim »Apfelmännchen« geht man von der komplexen Zahl Null aus, zieht eine komplexe Zahl C ab und quadriert das Ergebnis. Nun zieht man erneut C ab und quadriert das Ganze wiederum. Dieser Vorgang wiederholt sich so lange, bis ein vorher festgelegtes Grenzkriterium überschritten wird. Alle Zahlen, die hierbei gegen Unendlich streben, gehören zur Mandelbrot-Menge. Am Rand dieser Menge stellt sich jedoch ein seltsamer Effekt ein. Hier fallen oder steigen die Zahlen bei wechselndem Vorzeichen scheinbar völlig regellos. Dies ist der für die Grafik interessante Teil der Mandelbrot-Menge. Zwei Grenzkriterien legen nun fest, ob man sich innerhalb oder außerhalb beziehungsweise am Rand der Mandelbrot-Ebene befindet. Das erste Kriterium bricht den Algorithmus nach einer vorher festgelegten Zahl von Durchgängen ab. Im Listing geschieht dies durch die Variable »TMAX«. Wird dieser Wert erreicht, nimmt man an, daß die Folge gegen Unendlich strebt. Der Punkt auf dem Bildschirm erhält die Farbe Schwarz. Im anderen Fall wurde

das zweite Grenzkriterium zuerst erreicht. Dies ist dann der Fall, wenn die komplexe Zahl einen bestimmten Wert überschreitet. Im Listing wurde der Grenzwert 8 angenommen, der jedoch keine feste Größe darstellt, und fast beliebig geändert werden kann. Die zuzuordnende Farbe wird nun von der Anzahl der Iterationen abhängig gemacht. Im Listing geschieht dies mit der Zuordnung »TIEFE MOD 3 + 1«. Diese Berechnung kann zu den Farben Eins, Zwei und Drei, niemals jedoch zu der Farbe Null führen.

Für jeden Punkt, der berechnet werden soll, wird natürlich eine andere komplexe Zahl C gewählt. Die Umgebung der Mandelbrot-Ebene ergibt sich für folgende Werte: Realteil von -2 bis 1 und Imaginärteil von -1.5 bis 1.5 . Innerhalb dieser Grenzen kann man nun beliebig kleine Ausschnitte auf dem Bildschirm darstellen, also Teilbereiche der Menge gewissermaßen vergrößern. Dabei stellt man fest, daß immer neue Grafiken oder Figuren entstehen, die zum Teil zu anderen Ausschnitten ähnlich, aber nie genau gleich sind. Die einzige Einschränkung, die hierbei vom Computer auferlegt wird, liegt in der begrenzten Rechengenauigkeit. Bei theoretisch unendlicher Rechengenauigkeit ließe sich die Grafik unendlich oft vergrößern. Geht man also beispielsweise von den Grenzen $-1.1, -1$ und 1 aus, so wären als nächste Vergrößerungsstufe die Werte $-0.5, 0.5, -0.5$ und 0.5 denkbar. Halbiert man diese Werte immer wieder, so ist klar ersichtlich, daß auch eine zehnstellige Rechengenauigkeit bald nicht mehr zur Darstellung der Zahlen genügt.

Komplexe Formel, schöne Grafik

Zur Realisierung dieses Programms auf dem Atari wurde als Programmiersprache Turbo-Basic XL gewählt. Die zirka viermal so hohe Rechengeschwindigkeit, die diese Sprache gegenüber dem normalen Basic aufweist, ist für eine akzeptable Rechendauer unbedingt notwendig. Die Grafik wird in Grafikstufe 15 erzeugt. Dies bedeutet, daß genau 30720 Bildpunkte berechnet werden müssen. Abhängig von der maximalen Iterationszahl, die man mit »TMAX« bestimmt hat, kann jeder Punkt zudem bis zu 250mal berechnet werden. Selbst unter Turbo-Basic XL führt dies in Extremfällen zu Rechenzeiten von acht Stunden und mehr. Die Programmierung in Atari-Basic erscheint also wenig sinnvoll, will man nicht ein bis zwei Tage auf die Vollendung einer Grafik warten.

Handhabung des Programms

Nach dem Starten des Programms kommt man in das zentrale Menü, von dem aus alle Funktionen direkt ansprechbar sind. Punkt eins des Menüs betrifft den Start eines Apfelmännchens. Man muß zuerst die obere und untere Grenze des Real- und Imaginärbereichs eingeben. Auf die Frage

PROGRAMM-STECKBRIEF	
Programmname	Apfelmännchen
Programmtyp	Grafik
Programmiersprache	Turbo-Basic XL
Programmlänge	7155 Byte
für Computer	800 XL/130 XE
zusätzliche Hardware	Diskettenlaufwerk
Eingabehilfe	Prüfsummer
Bemerkung	Erzeugt und speichert Grafiken, die mittels einer speziellen mathematischen Formel berechnet werden. Auf der Leserservice-Diskette sind einige fertige Grafiken bereits gespeichert.
Leserservice	Diskette (APFELTUR)

»TIEFE?« antwortet man mit der maximalen Anzahl der Iterationen, die man zur Berechnung jedes Punktes zulassen will. Je höher man diesen Wert wählt, desto feiner werden die Grenzbereiche zur Mandelbrot-Ebene berechnet. Dies bedeutet allerdings auch eine Erhöhung der Rechenzeit, so daß man hier meist Kompromisse eingehen wird. Die Berechnung der Grafik kann dann auf dem Bildschirm mitverfolgt werden. Nach der Fertigstellung springt das Programm zurück ins Hauptmenü. Das fertige Bild kann man nun mit Option 2 auf dem Bildschirm betrachten. Mit den Tasten 0, 1, 2, und 3 werden die vier Grundfarben nach Belieben nachträglich geändert. Mit Hilfe des dritten Menüpunktes speichert man das fertige Bild. Wie es auch im Micropainterformat üblich ist, werden am Ende des Bildes vier Byte angehängt, die Auskunft über die verwendeten Farben geben. Zusätzlich speichert das Programm allerdings noch die eingegebenen Parameter. So kann man mit Punkt vier des Menüs also auch nur solche Bilder von Diskette laden, die das eben beschriebene Format besitzen.

Apfelmännchen schön und bunt

Das Inhaltsverzeichnis einer Diskette erhält man mit Punkt fünf des Menüs. Der sechste Punkt, nämlich »Effekte«, läßt die drei Vordergrundfarben eines Apfelmännchens über den Bildschirm »fließen«. Dies gibt der Grafik je nach Aussehen interessante, teilweise räumlich wirkende Effekte.

Um das Vergrößern einzelner Bildteile zu erleichtern, wurde Menüpunkt sieben aufgenommen. Mit Hilfe eines Joysticks kann man einen Bildausschnitt eingrenzen und erneut berechnen lassen. Dies geschieht, indem man zwei Eckpunkte eines gedachten Rechtecks entsprechend auf dem Bildschirm platziert. Welchen der Eckpunkte man bewegen will, wird mit der Feuertaste gewählt. Die Berechnung der neuen Grafik beginnt nach dem Drücken der START-Taste. Will man sich die neuen Parameter vorher noch ansehen, so wählt man einfach OPTION. Funktion acht des Menüs zeigt nun die Parameter, und mit Punkt neun startet die Berechnung des neuen Apfelmännchens.

Zum Programmaufbau

Das Programm »Apfelmännchen« wurde in Turbo-Basic XL geschrieben und konnte daher nicht nur wesentlich schneller, sondern auch strukturierter als im normalen Basic angelegt werden. So wird es auf den ersten Blick auffallen, daß beispielsweise auf Zeilennummern-orientierte Sprunganweisungen mit Ausnahme der »TRAP«-Befehle vollständig verzichtet wurde. Alle Hauptprogrammteile und Unterprogramme sind als Prozeduren angelegt und sinngemäß benannt.

Die Hauptschleife besteht aus einer DO-LOOP-Anweisung von Zeile 95 bis Zeile 240. Hier wird das Hauptmenü auf den Bildschirm gebracht, und von hier aus erfolgt der Sprung zu allen wesentlichen Prozeduren. Die einzelnen Prozeduren:

DECLARE

Es erfolgt die Dimensionierung aller Stringvariablen sowie die Erstbelegung der Variablen für die Änderung der Farbre-gister. Das Aussehen der beiden Player, die für die Ausschnittberechnung benötigt werden, legen die Variablen LO\$ und RU\$ fest.

RECHNEN

Hier findet man den Berechnungsalgorithmus, wie er im wesentlichen bereits erklärt wurde. Verwendung findet die Grafikstufe 15; man verfügt also über vier Farben bei einer Auflösung von 160 mal 192 Bildpunkten. Es wäre auch eine andere Grafikstufe denkbar. In diesem Fall müßten folgende Änderungen vorgenommen werden: Anpassung der Variablen DX, DY sowie der Schleifen ZEILE und SPALTE an die

neue Auflösung. In Zeile 1230 muß anstatt der drei die Anzahl der Farben minus eins eingesetzt werden.

EFFEKTE

Die drei Farbre-gister für die Vordergrundfarben werden mit verschiedenen Farbwerten belegt. Durch die Verschiebung der Farben entsteht der Eindruck des Fließens.

PARAMETER

Hier gibt man die Randwerte für ein neues Apfelmännchen ein. Ist dies geschehen, wird die Prozedur RECHNEN aufgerufen.

SEHEN

Über die Prozedur BILDHOLEN wird das momentan im Speicher befindliche Apfelmännchen auf dem Bildschirm dargestellt. Die Farbre-gister 708, 709, 710 und 712 werden durch Drücken der Tasten 0 bis 3 jeweils um den Wert eins inkrementiert. Die REPEAT-Anweisung wird bis zum Betätigen der Leertaste ausgeführt.

SPEICHERN

Zuerst erfolgt der Aufruf der Prozedur »FILEGET«. Dabei wird der Filename des Bildes gelesen, auf Fehler überprüft und gegebenenfalls ergänzt. Mit dem BPUT-Befehl wird der Inhalt des Bildschirmspeichers auf Diskette abgelegt. Es folgen die vier Farbytes und die Parameter. Da es sich bei letzteren um Gleitkommazahlen handelt, werden sie als Strings gespeichert.

LADEN

Diese Prozedur entspricht fast vollständig der Prozedur »SPEICHERN«. Alle Schreibbefehle sind sinngemäß in Lesebefehle umgewandelt. Eventuell auftretende Fehler bei der Diskettenhandhabung werden von TRAP-Routinen abgefangen.

DIRECTORY

Der linke Bildrand wird auf Position 7 gesetzt und das Directory mit dem DIR-Befehl aufgerufen. Nach Betätigung einer Taste erfolgt der Rücksprung ins Hauptmenü.

Ausschnitt

Diese Prozedur setzt sich aus mehreren Unterprogrammen zusammen. PMINIT initialisiert die grundsätzlichen Voreinstellungen für den Player/Missile-Aufruf. Verwendung finden die Player null und eins. Sie werden in Einzelzeilen-Auflösung dargestellt. Ihre Farbe ist Weiß und die erste Position wird durch die Variablen XO, YO, X1 und Y1 definiert. PMBUILD positioniert die beiden Player auf dem Bildschirm. Der Aufbau erfolgt dabei durch den MOVE-Befehl. Die folgende REPEAT-Anweisung fragt den Joystick und die Sondertasten ab. Abhängig von der Auslenkrichtung des Joysticks werden die Prozeduren »PMLEFT«, »PMRIGHT«, »PMUP« und »PMDOWN« angesprochen. Sie versetzen den jeweiligen Player um einen Pixel in die gewünschte Richtung. Ist die Positionierung abgeschlossen, werden die alten Parameter durch die neuen Werte ersetzt. Die Prozedur »PMEXIT« schaltet zuletzt die beiden Player wieder aus.

PARAMANZ

Hier werden lediglich die aktuellen Parameter auf dem Bildschirm angezeigt. Nach dem Betätigen einer beliebigen Taste gelangt man wieder ins Hauptmenü.

(Wolfgang Czerny/wb)

```

10 -- <YV>
15 REM - Apfelmaennchen in - <DT>
17 REM - Turbo-Basic XL fuer - <FU>
19 REM - Atari-Computer mit - <CE>
20 REM - mindestens 48 KByte-RAM- <BK>
25 REM - - <KM>
27 REM - Programmiert von - <RA>
29 REM - Wolfgang Czerny - <AR>
30 REM - (c) Happy-Computer - <LL>
35 -- <ZI>
90 EXEC DECLARE <YX>
95 DO <RA>
100 GRAPHICS 0:POKE 752,1 <RU>
110 POKE 709,0:POKE 710,10:POKE 712,10 <BY>
115 POKE 82,0 <HV>
120 POSITION 9,1:?"HAUPTMENUE" <XC>
130 POSITION 9,2:?"-----" <BC>
140 POSITION 3,5:?"(1)▲▲▲Parameter▲eing
eben" <SM>
150 POSITION 3,7:?"(2)▲▲▲Bild▲ansehen" <AW>
160 POSITION 3,9:?"(3)▲▲▲Bild▲speichern
" <DV>
170 POSITION 3,11:?"(4)▲▲▲Bild▲laden" <WQ>
180 POSITION 3,13:?"(5)▲▲▲Directory" <IU>
190 POSITION 3,15:?"(6)▲▲▲Effekte" <GC>
195 POSITION 3,17:?"(7)▲▲▲Ausschnitt" <JF>
197 POSITION 3,19:?"(8)▲▲▲Parameter▲anz
eigen" <WK>
199 POSITION 3,21:?"(9)▲▲▲Rechnen" <RF>
200 POSITION 3,23:?"(0)▲▲▲Ende"; <EA>
205 REPEAT <DM>
210 GET EIN <HE>
220 UNTIL EIN>47 AND EIN<58 <OH>
225 IF EIN=48 THEN EXIT <EF>
230 ON EIN-48 EXEC PARAMETER,SEHEN,SPEIC
HERN,LADEN,DIRECTORY,EFFEKTE,AUSSCHNITT,
PARAMANZ,RECHNEN <XD>
240 LOOP <CF>
250 END <XX>
1000 -- <VE>
1002 REM - Rechenroutine - <IU>
1003 -- <VQ>
1009 PROC RECHNEN <NR>
1010 GRAPHICS 15+16:POKE 764,255 <QS>
1030 DX=(XMAX-XMIN)/159 <MN>
1040 DY=(YMAX-YMIN)/191 <KE>
1050 CX=XMIN:CY=YMAX <SU>
1080 FOR ZEILE=0 TO 191 <FH>
1090 FOR SPALTE=0 TO 159 <WH>
1110 TI=0:XW=0:YW=0:X2=0:Y2=0 <JY>
1120 WHILE TI<TMAX AND (X2+Y2)<8 <PY>
1130 YW=2*XW*YW-CY:XW=X2-Y2-CX <TU>
1150 X2=XW^2:Y2=YW^2:TI=TI+1 <XN>
1180 WEND <UE>
1190 IF TI=TMAX <FI>
1200 COLOR %0 <QL>
1220 ELSE <TE>
1230 COLOR (TI MOD 3)+%1 <ID>
1240 ENDIF <SW>
1250 CX=CX+DX:CY=CY+DY <XA>
1260 IF PEEK(764)=28 THEN POP :GOTO 1295 <FH>
1270 NEXT SPALTE <DB>
1280 CX=XMIN:CY=CY-DY <DC>
1290 NEXT ZEILE <GB>
1295 EXEC BILDSPEICHERN <WL>
1300 ENDPROC <OI>
1301 -- <VO>
1302 REM - Bildeffekte - <DV>
1303 -- <VW>
1305 PROC EFFEKTE <JT>
1310 EXEC BILDHOLEN <IF>
1370 POKE 764,255 <GS>
1375 REPEAT <FU>
1380 POKE 708,COL0:POKE 709,COL1 <QE>
1390 POKE 710,COL2:PAUSE 5 <HI>
1400 POKE 708,COL1:POKE 709,COL2 <RG>

```

```

1410 POKE 710,COL0:PAUSE 5 <FC>
1420 POKE 708,COL2:POKE 709,COL0 <PS>
1430 POKE 710,COL1:PAUSE 5 <GA>
1455 UNTIL PEEK(764)<>255 <XF>
1460 ENDPROC <PC>
1470 -- <WH>
1480 REM - Parametereingabe - <MK>
1490 -- <WN>
2000 PROC PARAMETER <UK>
2005 REPEAT <EU>
2010 TRAP 2010 <LY>
2020 CLS :POKE 752,0 <NK>
2030 POSITION 9,3:?"PARAMETER▲EINGEBEN" <WB>
2040 POSITION 9,4:?"-----" <YI>
2045 REPEAT <FG>
2050 POSITION 3,7:INPUT "Linker▲Rand▲="
: XMIN <SZ>
2060 POSITION 3,9:INPUT "Rechter▲Rand▲="
: XMAX <HT>
2065 UNTIL XMIN<XMAX <PB>
2067 REPEAT <FU>
2070 POSITION 3,11:INPUT "Unterer▲Rand▲="
: YMIN <SS>
2080 POSITION 3,13:INPUT "Oberer▲Rand▲="
: YMAX <WL>
2082 UNTIL YMAX<YMIN <PO>
2083 REPEAT <FK>
2085 POSITION 3,15:INPUT "Tiefe▲(5▲-▲254
)▲=":TMAX <DJ>
2087 UNTIL TMAX>4 AND TMAX<255 <WJ>
2090 POSITION 3,20:?"ALLES▲IN▲ORDNUNG▲?
▲(J/N)▲": <DO>
2091 REPEAT <FF>
2092 GET EIN <JJ>
2093 UNTIL EIN=74 OR EIN=78 <ZR>
2094 UNTIL EIN=74 <KB>
2095 EXEC RECHNEN <KD>
2100 ENDPROC <OF>
2110 -- <VK>
2120 REM - Bild ansehen - <DP>
2130 -- <VQ>
3000 PROC SEHEN <YF>
3010 EXEC BILDHOLEN <IB>
3015 REPEAT <EY>
3017 REPEAT <FG>
3020 GET EIN <IH>
3040 UNTIL (EIN>47 AND EIN<52) OR EIN=32 <UQ>
3050 IF EIN=48=0 THEN COL0=COL0+1:IF COL
0>255 THEN COL0=COL0-256 <ZQ>
3060 IF EIN=48=1 THEN COL1=COL1+1:IF COL
1>255 THEN COL1=COL1-256 <IO>
3070 IF EIN=48=2 THEN COL2=COL2+1:IF COL
2>255 THEN COL2=COL2-256 <RM>
3080 IF EIN=48=3 THEN COL3=COL3+1:IF COL
3>255 THEN COL3=COL3-256 <AK>
3090 POKE 708,COL0:POKE 709,COL1:POKE 71
0,COL2:POKE 712,COL3 <XJ>
3100 UNTIL EIN=32 <EY>
3110 EXEC BILDSPEICHERN <UT>
3900 ENDPROC <OW>
3990 -- <WZ>
3992 REM - Bild auf Diskette - <KC>
3993 REM - speichern - <GS>
3994 -- <XP>
4000 PROC SPEICHERN <SV>
4010 CLS <TL>
4020 POSITION 9,3:?"BILD▲SPEICHERN" <BX>
4030 POSITION 9,4:?"-----" <TL>
4050 EXEC FILEGET <IN>
4100 EXEC BILDHOLEN <IB>
4110 OPEN #1,8,0,DAT$ <SV>
4120 #PUT #1,DPEEK(88),7680 <EU>
4130 #PUT #1,COL0,COL1,COL2,COL3 <PH>
4140 PRINT #1,STR$(XMIN) <DJ>
4150 PRINT #1,STR$(XMAX) <FO>
4160 PRINT #1,STR$(YMIN) <EJ>

```

Turbo-Basic XL-Listing zu »Apfelmännchen«

4170 PRINT #1,STR\$(YMAX)	<GO>	8140 XDELTA=X1POS-X0POS	<CH>
4180 PRINT #1,STR\$(TMAX)	<CV>	8145 DSOUND	<JO>
4190 CLOSE #1	<NP>	8150 UNTIL (START=6 OR START=5) AND XDEL	
4900 ENDFPROC	<OX>	TA>1 AND YDELTA>1	<TJ>
4990 --	<XA>	8160 DEX=203-40:DEY=223-32	<DS>
4992 REM - Bild von Diskette -	<QJ>	8170 DXD=(XMAX-XMIN)/DEX	<CP>
4993 REM - laden -	<ZH>	8180 DYD=(YMAX-YMIN)/DEY	<EY>
4994 --	<XQ>	8190 XMINN=XMIN+DXD*(X0POS-44)	<VS>
5000 PROC LADEN	<RM>	8200 XMAXN=XMIN+DXD*(X1POS-35)	<WR>
5010 CLS	<TM>	8210 YMINN=YMIN+DYD*(Y0POS-30)	<RO>
5020 POSITION 9,3:?"BILD.LADEN"	<IG>	8220 YMAXN=YMIN+DYD*(Y1POS-28)	<BK>
5030 POSITION 9,4:?"-----"	<QI>	8230 TMAX=254	<DC>
5050 EXEC FILEGET	<IO>	8240 XMIN=XMINN:XMAX=XMAXN	<XZ>
5100 GRAPHICS 15+16	<FR>	8250 YMIN=YMINN:YMAX=YMAXN	<AB>
5110 OPEN #1,4,0,DAT\$	<QS>	8260 EXEC PMEXIT	<DF>
5120 BGOT #1,DPEEK(88),7680	<XM>	8270 IF START=6 THEN EXEC RECHNEN	<ZC>
5125 GET #1,COL0,COL1,COL2,COL3	<JP>	8900 ENDFPROC	<PB>
5130 INPUT #1,E\$:XMIN=VAL(E\$)	<GM>	8990 --	<XE>
5140 INPUT #1,E\$:XMAX=VAL(E\$)	<IN>	8991 REM - Player Missile-Grafik -	<JV>
5150 INPUT #1,E\$:YMIN=VAL(E\$)	<HK>	8992 REM - initialisieren -	<QJ>
5160 INPUT #1,E\$:YMAX=VAL(E\$)	<JL>	8993 --	<XQ>
5165 INPUT #1,E\$:TMAX=VAL(E\$)	<GT>	9000 PROC PMINIT	<DA>
5170 CLOSE #1	<NK>	9010 PMB=PEEK(106)-40:POKE 54279,PMB	<MV>
5180 EXEC BILDSPEICHERN	<VQ>	9020 PMB=PMB*256:POKE 559,62	<XL>
5190 ENDFPROC	<PJ>	9030 POKE 53256,0:POKE 53257,0	<YU>
5199 --	<XV>	9040 PMNR=0:POKE 704,255:POKE 705,255	<HP>
5200 REM - Directory auf dem -	<PR>	9050 MOVE ADR(LEER\$),PMB+1024,256	<BE>
5210 REM - Bildschirm ausgeben -	<PI>	9060 MOVE ADR(LEER\$),PMB+1280,256	<FJ>
5220 --	<VS>	9070 X0POS=44:Y0POS=30	<DF>
6000 PROC DIRECTORY	<JQ>	9080 X1POS=198:Y1POS=218	<GF>
6010 CLS :POKE 82,7:?	<XQ>	9090 ENDFPROC	<PL>
6020 DIR	<ST>	9092 --	<WV>
6030 GET EIN	<IN>	9093 REM - Player Missiles auf- -	<FG>
6035 POKE 82,0	<LP>	9094 REM - bauen und mani- -	<KO>
6040 ENDFPROC	<OT>	9095 REM - pulieren -	<GK>
6050 --	<VY>	9099 --	<XX>
6060 REM - Filename abfragen -	<VX>	9100 PROC PMBUILD	<QX>
6070 --	<WE>	9110 POKE 53277,3	<EX>
7000 PROC FILEGET	<MB>	9120 POKE 53248,X0POS:POKE 53249,X1POS	<RZ>
7005 TRAP 7005	<RC>	9130 MOVE ADR(LO\$),PMB+1024+Y0POS,8	<NX>
7007 POKE 752,0	<PC>	9140 MOVE ADR(RU\$),PMB+1280+Y1POS,8	<AC>
7010 POSITION 3,7:INPUT "WELCHER FILE-NA		9190 ENDFPROC	<PN>
ME";FILE\$	<JL>	9200 PROC PMEXIT	<GO>
7020 DAT\$="-----"	<AO>	9210 POKE 53277,0	<DA>
7050 IF FILE\$(1,2)<>"D:"	<EN>	9220 POKE 559,34	<YW>
7060 DAT\$(3,LEN(FILE\$)+2)=FILE\$	<TB>	9230 MOVE ADR(LEER\$),PMB+1024,256	<BC>
7070 DAT\$(1,2)="D:"	<CH>	9240 MOVE ADR(LEER\$),PMB+1280,256	<FH>
7080 ELSE	<TY>	9250 ENDFPROC	<PD>
7090 DAT\$=FILE\$	<UD>	9290 --	<WR>
7095 ENDF	<UH>	9292 REM - Player Missiles nach -	<NS>
7110 TRAP 20000	<NK>	9293 REM - oben bewegen -	<MD>
7900 ENDFPROC	<PA>	9294 --	<XH>
7990 --	<XD>	9300 PROC PMUP	<UN>
7992 REM - Ausschnitt waehlen -	<MB>	9310 IF PMNR=0	<VJ>
7993 REM - fuer diesen Programm- -	<PA>	9312 Y0POS=Y0POS-1	<UY>
7994 REM - teil wird ein Joy- -	<YA>	9314 IF Y0POS<30 THEN Y0POS=30	<XU>
7995 REM - stick benoetigt -	<VL>	9317 MOVE ADR(LO\$),PMB+1024+Y0POS,8	<OX>
7996 --	<YB>	9318 ENDF	<UD>
8000 PROC AUSSCHNITT	<QS>	9320 IF PMNR=1	<WA>
8010 EXEC BILDHOLEN	<IG>	9322 Y1POS=Y1POS-1	<VV>
8015 EXEC PMINIT	<ZW>	9324 IF Y1POS<28 THEN Y1POS=28	<LR>
8020 EXEC PMBUILD	<NA>	9327 MOVE ADR(RU\$),PMB+1280+Y1POS,8	<BC>
8022 REPEAT	<EU>	9328 ENDF	<UG>
8024 REPEAT	<FC>	9390 ENDFPROC	<PR>
8030 JOY=STICK(0)	<CP>	9392 --	<XB>
8040 PRESS=STRIG(0)	<FG>	9393 REM - Player Missiles nach -	<NY>
8045 START=PEEK(53279)	<IL>	9394 REM - unten bewegen -	<RN>
8050 UNTIL JOY<>15 OR PRESS=0 OR START<>		9399 --	<YD>
7	<NQ>	9400 PROC PMDOWN	<FT>
8060 IF JOY=11 THEN EXEC PMLEFT	<AA>	9410 IF PMNR=0	<VL>
8070 IF JOY=7 THEN EXEC PMRIGHT	<EO>	9412 Y0POS=Y0POS+1	<TS>
8080 IF JOY=14 THEN EXEC PMUP	<YX>	9414 IF Y0POS>221 THEN Y0POS=221	<YA>
8090 IF JOY=13 THEN EXEC PMDOWN	<QL>	9416 MOVE ADR(LO\$),PMB+1024+Y0POS,8	<OV>
8100 IF PRESS=0 AND PMNR=0	<MY>	9418 ENDF	<UF>
8105 PMNR=1:SOUND 0,10,10,10:PAUSE 10	<PY>	9420 IF PMNR=1	<WC>
8107 ELSE	<UF>	9422 Y1POS=Y1POS+1	<UP>
8110 IF PRESS=0 AND PMNR=1 THEN PMNR=0:S		9424 IF Y1POS>218 THEN Y1POS=218	<KW>
OUND 0,100,10,10:PAUSE 10	<UU>	9426 MOVE ADR(RU\$),PMB+1280+Y1POS,8	<BA>
8120 ENDF	<SV>	9428 ENDF	<UI>
8130 YDELTA=Y1POS-Y0POS	<DQ>	9590 ENDFPROC	<PV>

Turbo-Basic XL-Listing zu »Apfelmännchen« (Fortsetzung auf Seite 66)

9592 --	<XF>	11040 ENDPROC	<UP>
9593 REM - Player Missiles nach -	<OC>	11100 --	<JH>
9594 REM - links bewegen -	<NZ>	11102 REM - Bild speichern und -	<HU>
9599 --	<YH>	11103 REM - Parameter anzeigen -	<UN>
9600 PROC PMLEFT	<YK>	11104 --	<KB>
9620 IF PMNR=0	<VS>	11110 PROC BILDSPEICHERN	<HD>
9622 X0POS=X0POS-1	<UP>	11120 MOVE DPEEK(88),ADR(SCREEN\$),7680	<AG>
9624 IF X0POS<44 THEN X0POS=44	<FR>	11130 ENDPROC	<UO>
9626 POKE 53248,X0POS	<OZ>	15000 PROC PARAMANZ	<PR>
9628 ENDIF	<UM>	15010 CLS	<OD>
9630 IF PMNR=1	<WJ>	15020 POSITION 9,3: ? "PARAMETER ANZEIGEN	
9632 X1POS=X1POS-1	<VM>	"	<TP>
9634 IF X1POS<43 THEN X1POS=43	<FI>	15030 POSITION 9,4: ? "-----	
9636 POKE 53249,X1POS	<QJ>	"	<CC>
9638 ENDIF	<UP>	15040 POSITION 3,7: ? "Linker Rand=";XM	
9790 ENDPROC	<PZ>	IN	<OZ>
9792 --	<XJ>	15050 POSITION 3,9: ? "Rechter Rand=";X	
9793 REM - Player Missiles nach -	<OG>	MAX	<RQ>
9794 REM - rechts bewegen	<GM>	15060 POSITION 3,11: ? "Unterer Rand=";Y	
9799 --	<YL>	YMAX	<OJ>
9800 PROC PMRIGHT	<ZV>	15070 POSITION 3,13: ? "Oberer Rand=";Y	
9820 IF PMNR=0	<VW>	MIN	<HX>
9822 X0POS=X0POS+1	<TL>	15080 POSITION 3,15: ? "Tiefe=";TMAX	<JY>
9824 IF X0POS>199 THEN X0POS=199	<XE>	15100 GET EIN	<LG>
9826 POKE 53248,X0POS	<PD>	15110 ENDPROC	<UO>
9828 ENDIF	<UO>	15190 --	<KZ>
9830 IF PMNR=1	<WN>	15200 REM - Fehlerbehandlung -	<ZG>
9832 X1POS=X1POS+1	<UI>	15210 --	<JW>
9834 IF X1POS>198 THEN X1POS=198	<WT>	20000 REM FEHLERROUTINE	<ND>
9836 POKE 53249,X1POS	<QN>	20010 ? "(ESC CTL 2)"	<DP>
9838 ENDIF	<UT>	20015 POSITION 3,10: ? "FEHLER NR.";ERR;	
9990 ENDPROC	<OD>	"_IST_AUFGETRETEN"	<OH>
10000 --	<JC>	20017 POSITION 3,12: ? "ZEILEN NR.";ERL	<ZA>
10002 REM - Variablen und Player -	<FN>	20020 IF ERR=170 THEN POSITION 3,20: ? "D	
10003 REM - Missiles definieren -	<KZ>	ATEI_NICHT_VORHANDEN"	<SZ>
10004 --	<JW>	20030 IF ERR=162 THEN POSITION 3,20: ? "D	
10009 PROC DECLARE	<VM>	ISKETTE_IST_VOLL"	<EJ>
10010 DIM FILE\$(20),DAT\$(20),SCREEN\$(768		20040 IF ERR=164 THEN POSITION 3,20: ? "D	
0),LEER\$(256),LO\$(10),RU\$(10),E\$(20)	<WZ>	ATEI_NICHT_IN_ORDNUNG"	<IG>
10020 COL0=40:COL1=202:COL2=148:COL3=0	<AJ>	20050 IF ERR=169 THEN POSITION 3,20: ? "D	
10030 LEER\$(1)="(CTL ,)":LEER\$(256)="(CTL		IRECTORY_IST_VOLL"	<QF>
L ,)":LEER\$(2)=LEER\$	<IW>	20060 IF ERR=160 THEN POSITION 3,20: ? "D	
10040 LO\$="(CTL ,){CTL O}{CTL H}{CTL H}{		ISKETTENLAUFWERK_NICHT_BEKANNT"	<SV>
CTL H}{CTL H}{CTL H}{CTL H}{CTL ,)":RU\$="(CTL		20070 IF ERR=128 THEN POSITION 3,20: ? "U	
,){CTL A}{CTL A}{CTL A}{CTL A}{CTL A}{CTL		EBERTRAGUNG_UNTERBROCHEN"	<VF>
O}{CTL ,)"	<EZ>	20080 IF ERR=167 THEN POSITION 3,20: ? "D	
10100 ENDPROC	<UA>	ATEI_IST_SCHREIBGESCHUETZT"	<ZX>
11000 --	<JE>	20090 IF ERR=144 THEN POSITION 3,20: ? "D	
11002 REM - Apfelmaennchen auf -	<GF>	ISKETTE_IST_SCHREIBGESCHUETZT"	<TI>
11003 REM - dem Bildschirm dar- -	<CJ>	20100 IF ERR=138 THEN POSITION 3,20: ? "G	
11004 REM - stellen -	<UC>	ERAET_NICHT_ANSPRECHBAR"	<JT>
11005 --	<KD>	20110 IF ERR=165 THEN POSITION 3,20: ? "F	
11010 PROC BILDHOLEN	<IX>	ALSCHER_DATEINAME"	<AB>
11015 GRAPHICS 15+16	<YR>	20115 POSITION 3,22: ? "WEITER MIT RETURN	
11020 MOVE ADR(SCREEN\$),DPEEK(88),7680	<IF>	!"	<WP>
11030 POKE 708,COL0:POKE 709,COL1:POKE 7		20120 GET EIN:TRAP 40000	<AU>
10,COL2:POKE 712,COL3	<IB>	20130 GOTO 95	<WZ>

Turbo-Basic XL-Listing zu »Apfelmännchen« (Schluß)

