

Der VC 20 als Laterna Magica

Laterna Magica macht Spiele beweglicher und professioneller. Man kann Zeichen und zusammengesetzte Bilder nicht nur entwickeln, sondern auch bewegen, pulsieren, rotieren oder explodieren lassen und in eigene Programme einbauen.

»Laterna Magica« ist für jeden VC 20 gedacht, sei er erweitert oder nicht. Als Peripherie wird ein Floppy-Disk-Laufwerk VC-1541 vorausgesetzt. Das Programm kann aber für die Datensette umgeschrieben werden, indem die Gerätenummer 8 in Zeile 60 des DATA-Laders in eine 1 umgewandelt wird. Außerdem muß es dann im Hauptprogramm heißen:

```
490 PRINT"SAVE"CHR$(34) "CHR$(34)",1"
```

Das Programm (Listing 1) ist nicht mehr zu erweitern, es können also auch keine REM-Zeilen mehr eingefügt werden. Es sieht zwar als Listing nicht sehr umfangreich aus, jedoch mußte das Basic-Ende heruntergesetzt werden, um das Maschinenprogramm, den neuen Zeichengenerator und den Speicherplatz, in dem wir später unsere Ersatzbilder speichern, vor Überschreibung zu schützen. Die Erzeugung von Grafik beim VC 20 beruht ja darauf, die Zeichen auf dem Bildschirm, die aus 88 Punkten aufgebaut sind, in ihrem Aufbau zu ändern. Bei einem Zeichen hieße es, 8 Byte gegen 8 andere auszutauschen. Bei 9 Zeichen (das ist die Zeichenanzahl, mit der »Laterna Magica« arbeitet) werden es 72 Byte. Versuchte man das Problem mit Basic zu lösen, müßte man bei jedem Tauschvorgang eine Zählschleife von 1 bis 72 mit einigen POKEs dazwischen in Kauf nehmen. So etwas kann Action-Spiele ganz schön langsam machen. Ich habe diese Routine daher in Maschinensprache geschrieben. So genügt ein SYS-Aufruf, und 9 Zeichen auf dem Bildschirm ändern sofort ihre Form.

Ausgeführt werden kann der Tausch nur, wenn der Zeichengenerator nicht wie üblich im ROM liegt. Das ist also die erste Aufgabe des Maschinenprogramms. Es legt den Zeichengenerator ins RAM ab \$1C00 (entspricht dezimal 7168). Danach holt es sich aus dem Reserve-Bildschirmspeicher (\$1A00 bis \$1BFF, dezimal 6656 bis 7167) die ersten 72 Byte und legt sie im Bildschirmspeicher ab. Das ist zunächst alles. Beim nächsten Aufruf verfährt es mit den folgenden 72 Byte in gleicher Weise. Das geht siebenmal so, danach werden wieder die ersten 72 Byte geholt und so weiter.

Wichtig ist, daß vor jedem Aufruf in die Speicher \$FB und \$FC (dezimal 251 und 252) ein Wert von maximal 7 gePOKEt wird. An diesem Wert erkennt das Maschinenprogramm, wievielmals es die Bilder austauschen soll, bis es wieder beim ersten beginnt. Das bedeutet, daß man auch »Filme« mit nur drei Bildern laufen lassen kann, indem man den Wert 3 in diese Speicher POKet.

Der Bildeditor

Um diese besprochenen Bilder zu entwickeln, wären langwierige Berechnungen und viel Tastaturarbeit zum Eingeben der bis zu 504 Werte notwendig. Deswegen habe ich ein Programm geschrieben, das diese Arbeit übernimmt. (Listing 2).

Der Umgang mit dem Bildeditor ist recht einfach. Nach RUN erscheint in der oberen Hälfte des Bildschirms eine Maske, in der die Zeichen aufgebaut werden und in der unteren Hälfte das Menü (siehe Bild).

Das Feld rechts oben zeigt das erste Bild. Man sieht zwar zunächst nur ein abstraktes Gebilde, das liegt aber nur daran, daß noch keine Zeichen entwickelt wurden. Links daneben stehen die gleichen Zeichen in Klarschrift. In beiden Feldern sind die gleichen Zeichen in der linken oberen Ecke schwarz, um herauszustellen, daß dieses Zeichen bearbeitet werden kann.

Mit F1 kann man nun ein Zeichen auswählen, das verändert werden soll. Dabei wandert das schwarze Feld jeweils eine Ziffer weiter.

Weiterhin kann man über den beiden Feldern die Bildnummer ablesen. Mit F3 können wir zwischen sieben Bildern wählen. Die Bilder werden jeweils im rechten Feld gezeigt. Es entsteht so schon ein Zeichentrickeffekt.

Hat man nun seine Wahl getroffen, kann man mit F7 in den Entwicklungsmodus wechseln. Der Cursor steht nun in der linken oberen Ecke des großen Entwicklungsfeldes. Er läßt sich

```
10 REM * LATERNA-MAGICA-DATA-LOADER *
12 REM
15 REM =====
16 REM
20 POKE52,25:POKE56,25:CLR
30 FORI=6400TO6640:READA:POKEI,A:S=S+A:N
EXT
40 IFS<>26746THENPRINT"DATA-FEHLER!":END
50 PRINT"OK!"
60 A$="LOAD"+CHR$(34)+"LATERNA MAGICA"+CHR$(34)+"",8"
70 PRINTCHR$(147);A$:PRINTCHR$(19):PRINT
:PRINT:PRINT:PRINT:PRINT"RUN"
75 POKE631,19:POKE632,13:POKE633,13:POKE
198,3:NEW
80 DATA173,5,144,41,240,9,15,141,5,144,1
98,251,162,72,189,71
90 DATA26,157,127,29,202,208,247,173,15,
25,24,105,72,141,15,25
100 DATA173,16,25,105,0,141,16,25,165,25
1,208,14,169,255,141,15
110 DATA25,169,25,141,16,25,165,252,133,
251,96,169,0,133,204,32
120 DATA245,241,240,251,76,211,25,234,72
,120,165,207,240,12,165,206
130 DATA174,135,2,160,0,132,207,32,161,2
34,88,104,32,122,242,166
140 DATA214,224,8,48,2,162,0,164,211,192
,8,48,2,160,0,32
150 DATA12,229,76,175,25,32,129,229,162,
0,142,60,3,160,0,177
160 DATA209,201,160,240,9,169,1,24,109,6
0,3,141,60,3,192,7
170 DATA240,3,14,60,3,200,192,8,48,229,3
2,19,229,232,173,60
180 DATA3,32,199,25,224,8,240,6,32,12,22
9,76,120,25,96,32
190 DATA19,229,152,72,138,72,169,1,133,2
04,32,117,25,104,170,104
200 DATA168,32,12,229,76,59,25,157,7,26,
157,135,29,96,104,168
210 DATA104,170,96,201,13,240,3,76,72,25
,120,165,207,240,12,165
220 DATA206,174,135,2,160,0,132,207,32,1
61,234,88,169,1,133,204
230 DATA96
```

READY. Listing 1. »Laterna Magica« Basic-Lader

```

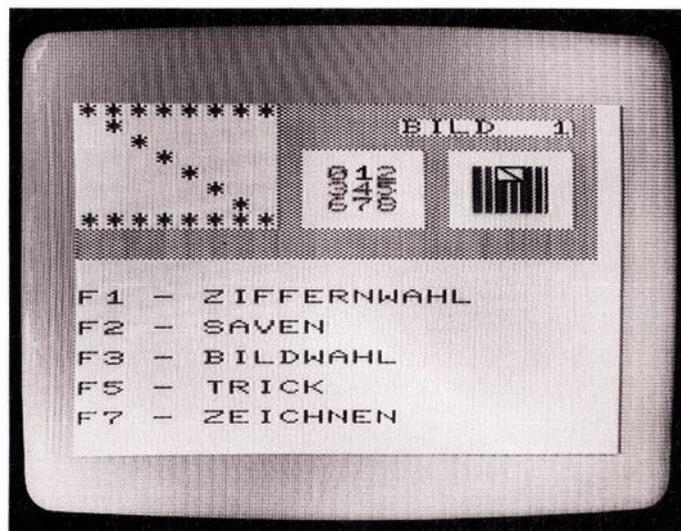
10 POKE36869,PEEK(36869)AND240OR15
20 IF(PEEK(36866)OR127)=255THENAZ=38400
30 IF(PEEK(36866)OR127)<>255THENAZ=37888
40 BI=0:POKE251,7:POKE252,7:POKE6415,255
:POKE6416,25:SYS6400
50 PRINT" ";:FORI=1TO11:PRINT"
":NEXT
60 PRINT" ";:FORI=1TO8:PRINT"
":NEXT
70 PRINT"TAB(9)"
TAB(9)"012"
80 PRINT TAB(9)"345"
90 PRINT TAB(9)"678":PRINTTAB
B(9)"
":GOTO200
100 PRINT" ";:SYS6459:POKE204,1
110 PRINT" ";:WV=11:GOSUB510
120 PRINT" F1 - ZIFFERNWAHL"
130 PRINT" F2 - SAVEN"
140 PRINT" F3 - BILDWAHL"
150 PRINT" F5 - TRICK"
160 PRINT" F7 - ZEICHNEN"
170 GETA$:IFA$=""THEN170
180 IFASC(A$)<133ORASC(A$)>137THEN170
190 ONASC(A$)-132GOTO200,280,360,250,450
200 ZI=ZI+1:IFZI>9THENZI=1
210 FORJ=1TO3:FORI=0TO2:HH=AZ+104+I+K:HI
=HH-6:TE=TE+1:IFTE=ZITHEM230
220 POKEHH,5:POKEHI,5:GOTO240
230 POKEHH,0:POKEHI,0
240 NEXTI:K=K+22:NEXTJ:K=0:TE=0:PR=1:GOT
0290
250 PRINTCHR$(19)::WV=10:GOSUB510
260 FORI=1TO11:PRINT"
N ":NEXT:PRINT" ";:IFLE=1THENLE=0:RETUR
270 GOTO100
280 BI=BI+1:IFBI>6THENBI=0
290 PRINT"TAB(13)"BILD "BI+1
300 X=72*BI+8*(ZI-1)+6655:Q=INT(X/256):R
=X-256*Q
310 POKE6600,R:POKE6601,Q
320 HI=7551+8*(ZI-1):H=INT(HI/256):L=HI-
256*H
330 POKE6603,L:POKE6604,H
340 IFPR=1THENPR=0:GOTO110
350 SYS6400:GOTO110
360 LE=1:GOSUB250:PRINT" ";:WV=11:GOSUB5
10:PRINT"TEMPO 0=MAX."
370 PRINT:PRINTTAB(7)"1000=KLEIN"
380 PRINT:PRINT"1100";:INPUT" ";:T
E
390 PRINT" ";:WV=8:GOSUB510:PRINTTAB(9)"0
12"
400 PRINTTAB(9)"345"
410 PRINTTAB(9)"678"
420 PRINT" > STOP << MIT SPACE!"
430 SYS6400:FORI=0TOTE:NEXT:IFPEEK(197)=
32THENLE=1:GOSUB250:CLR:GOTO10
440 GOTO430
450 POKE680,PEEK(43):POKE681,PEEK(44):PO
KE682,PEEK(45):POKE683,PEEK(46)
460 POKE36869,PEEK(36869)AND240
470 PRINT"BITTE NAMEN EINSETZEN:"
480 PRINT"PF43,0:PF44,26:PF45,0:PF46,28:
PF198,1:PF631,13:"
490 PRINT"SAVE"CHR$(34)"@:
"CHR$(34)",8,1"
500 PRINT"PF43,P(680):PF44,P(681):P
(682):PF46,P(683):RUN":END
510 FORI=1TOWV:PRINTCHR$(17)::NEXT:RETUR
N
READY.

```

Listing 2.
»Laterna Magica«
Grafik-Editor

voll steuern, jedoch nicht über den Feldrand hinaus. Tippt man nun, mehr oder weniger willkürlich, Buchstaben oder Zeichen in das Feld, sieht man gleichzeitig an der vorgewählten Ziffer im rechten Feld das Ergebnis. Man sollte nur nicht den Fehler machen, CLR/HOME oder INST/DEL zu drücken. Das kann den Bildschirmaufbau durcheinander bringen.

Mit RETURN kommt man ins Menü zurück.



Der Bildeditor

Ist eine Bild- oder Zeichenfolge fertiggestellt, kann man den Trickfilm mit F5 ausprobieren. Im Menüfeld wird die Geschwindigkeit des Ablaufes erfragt. Eine gute Geschwindigkeit von 100 ist vorgegeben. Ein Wert von Null ergibt die größte Geschwindigkeit. In der anderen Richtung wird man selbst merken, wenn es langweilig wird. Mit RETURN wird gestartet, und mit der SPACE-Taste kann der Vorgang abgebrochen werden.

Es bleibt noch die F2-Funktion zu erklären. Bei Druck auf F2 erscheinen ein paar im ersten Augenblick wirr anmutende Befehle und die Bitte, den Namen der Bildfolge einzutragen. Das bedeutet, daß in der ersten Zeile (mit den vielen POKES und dem SAVE) in die Lücke ein Name eingetragen werden soll. Es ist auch angebracht, das zweite Anführungszeichen an den Namen heranzuziehen. Dann drückt man RETURN, und alles geht wieder automatisch. Die Bildfolge wird als Maschinenprogramm geSAVET und das Programm startet von neuem.

Zur Eingabe des Programmes: Bei Verwendung der Datensette muß zunächst der DATA-Lader (Listing 1) abgetippt und abgespeichert werden und danach Listing 2, weil der DATA-Lader selbständig das nächste Programm nachlädt. Verwendet man ein Floppy-Laufwerk, ist die Reihenfolge unerheblich.

Will man mit dem Programm arbeiten, muß erst der DATA-Lader geladen werden. »RUN« startet den Lader, der das Hauptprogramm lädt.

Bildfolgen in eigenen Programmen

Will man die Trickfolge im selbstgemachten Programm anwenden, muß natürlich das Maschinenprogramm im Speicher vorhanden sein. Dazu lädt man den DATA-Lader, löscht die Zeilen 60-75, setzt in Zeile 60 ein NEW und startet den Lader. Damit ist das Maschinenprogramm im Bereich dezimal 6400 bis 6640 abgelegt und vor Überschreibung geschützt. Der Lader löscht sich selbst. Außerdem ist auch der Bereich geschützt, in den nun die selbstgemachte Bildfolge mit »LOAD "Name",8,1« oder »LOAD "Name",1,1« geladen werden kann.

Damit keine Fehlermeldungen auftreten, gibt man jetzt noch den Befehl NEW ein und drückt die Return-Taste. Jetzt kann man das eigene Programm laden oder eintippen.

Programm-Übersicht

- 10 — 40:** Hier wird die Anfangsadresse des Zeichenspeichers nach Adresse 7168 verlegt, der Beginn des Farb-RAM ermittelt (unterschiedliche Werte bei GV und Erweiterungen) und das Maschinenprogramm »Bildtausch« initialisiert.
- 50 — 90:** Aufbau des Bildschirms
- 100:** Einsprung in das Maschinenprogramm »Zeichen-Entwicklung«
- 110 — 190:** Menü
- 200 — 240:** Ziffernwahl. Bei Betätigung von F1 wird der Ziffernzähler erhöht und diese Änderung auf dem Bildschirm kenntlich gemacht (Unterlegung der gewählten Ziffer mit Schwarz, der Rest Blau).
- 260 — 270:** Löschen des Menüs
- 280 — 350:** Bildwahl. Berechnung der Anfangsadresse des nächsten Zeichens im Reserve-Bildspeicher und im Zeichenspeicher aus den Werten BI (Bildnr.) und ZI (Ziffernr.). Die Adressen werden in das LB-HB-Format umgewandelt und an das MP »Zeichen-Entwicklung« übergeben.
- 360 — 440:** Trick. Der Bildschirm wird neu aufgebaut. Die Hauptsache ist jedoch die Zeile 430. Dort wird immer wieder das Maschinenprogramm »Bildtausch« aufgerufen, das einen Mini-Trickfilm auf dem Bildschirm erzeugt.
- 450 — 500:** Basic-Anfang und -Ende werden in den Bytes 680-683 abgelegt. Der Zeichenspeicher wird normalisiert. Es wird eine Maske zum Abspeichern des Reserve-Bildspeichers ausgegeben. Maske: Basic-Anfang und -Ende werden gleich Reserve-Bildspeicher-Anfang und -Ende gesetzt. In den Tastaturpuffer wird ein RETURN = CHR\$(13) gePOKEt, damit nach dem Abspeichern die zweite Zeile der Maske abgearbeitet wird, die Basic-Anfang und -Ende wieder in den Ausgangszustand bringt. Mit dem »RUN« am Ende der Zeile startet das Programm wieder.
- 510:** Unterprogramm zur Ausgabe von mehreren »Cursor Down«

AZ	Beginn des Farb-RAM
BI	Bildzähler
I,J,K	Schleifenzähler
WV	Parameterübergabe für Unterprogramm in Zeile 510
A\$	Abfrageergebnis aus dem Menü
ZI	Ziffernzähler
HH,HI	Farbspeicher, die auf dem Bildschirm schwarz erscheinen sollen
TE	Zähler zum Vergleich mit aktueller Ziffer (200 bis 240) und Geschwindigkeit für »Trick« (360 bis 440)
PR	Verzweigungsargument für Unterprogramm (280 bis 350)
LE	Verzweigungsargument für Zeile 260
X	1. Speicherplatz für das nächste zu erstellende Zeichen (im Reserve-Bildspeicher)
Q	HB von X
R	LB von X
HI	Äquivalent zu X (für Zeichenspeicher)
H	HB von HI
L	LB von HI

Die wichtigsten Variablen

Das Maschinenprogramm wird übrigens mit SYS6400 aufgerufen.

Und noch einmal: Am Anfang des Programmes, vor dem ersten SYS6400 muß unbedingt eine Zahl zwischen 1 und 7 in 251 und 252 gePOKEt werden, eben die Anzahl der Bilder des Trickfilms. Alle kleineren Zahlen als 1 und alle größeren als 7 richten zwar keinen Schaden an, jedoch bewirken sie eine Einschränkung der Funktion.

Sollte man dies einmal vergessen haben, ist es angebracht, das Maschinenprogramm neu zu laden.

Zum Schluß noch zwei Tips.

1. Bei erweitertem VC 20 sollte man für eigene Programme nicht das Basic-Ende herunter-, sondern den Basic-Beginn heraufsetzen. Damit hat man mehr Speicherplatz für seine Programme. (Zeile 20) des DATA-Laders: POKE44,32:POKE8192,0:CLR).
2. Da bei der Arbeit mit diesem Programm nicht der Original-Zeichensatz vorhanden ist, kann man ihn, wenn gebraucht, über die RVS-ON-Taste darstellen. Dieses Verfahren habe ich selbst im Listing 1 verwendet. (Bernd Schrödter/ev)

Grafik leicht gemacht

Mit einigen neuen Befehlen macht dieses Programm das Plotten von Funktionen zum Kinderspiel. Das mühsame Arbeiten mit den Commodore-Grafikbefehlen ist vorbei, wenn man die neuen Befehle einsetzt.

Das Programm stellt eine Basic-Erweiterung dar, die im Bereich von \$0800-\$4000 angesiedelt ist. Dieser Bereich enthält das Maschinenspracheprogramm sowie den Grafikbildschirm. Die neuen Befehle und ihre Bedeutung:

HIRES: Dieser Befehl schaltet die hochauflösende Grafik ein.
SCNCLR: Hiermit wird der Grafikbildschirm gelöscht.

TEXT: Um die Grafik wieder auszuschalten wird dieser Befehl benutzt. Am Programmende schaltet die hochauflösende Grafik automatisch ab.

REGION a: Die Cursor- beziehungsweise Plotfarbe wird festgelegt. Der Parameter a darf Werte von 0 bis 15 annehmen.

COLOR a,b,c: Bestimmt die Hintergrundfarbe (a), die Rahmenfarbe (b) und die Zeichenfarbe (c).

PLOT x,y: Dieser Befehl setzt einzelne Punkte auf dem Grafikbildschirm. Die Parameter x und y geben die Position des Punktes an. Der Ursprung des Koordinatensystems liegt in der linken oberen Bildschirmcke. X darf Werte von 0 bis 319, y Werte von 0 bis 199 annehmen. Bei Überschreiten dieser Bereiche gibt der Computer jedoch keine Fehlermeldung aus, wie es bei anderen Programmen meist der Fall ist. Für die Darstellung von Funktionen ist diese Einrichtung jedoch sehr nützlich.

UNPLOT x,y: Löscht einen gesetzten Punkt. Für die Parameter gelten die gleichen Grenzen wie beim Befehl PLOT.