

Leben und sterben lassen

»Life« ist wohl eine der faszinierendsten Simulationen von biologischen Vorgängen auf Computern. Mit »Life« lassen sich evolutionäre Abläufe spielerisch verstehen lernen.

Obwohl den Lesern der Inhalt des »Spiels« geläufig sein mag, hier die »Regeln«:

Es existiert ein zweidimensionales Feld und an jedem Punkt ist eine lebende oder eine tote Zelle. Jede Zelle hat acht Nachbarn.

An einem Punkt wird eine Zelle dann und nur dann lebendig, falls sie genau drei lebende Nachbarn hat.

Fast wie im richtigen Leben

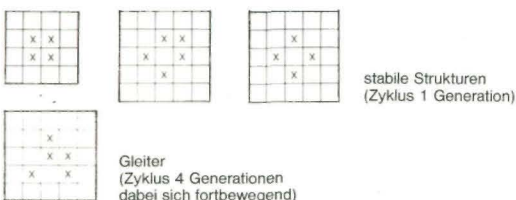
Eine Zelle stirbt, wenn sie weniger als zwei (Vereinsamung) oder mehr als drei (Überbevölkerung) lebende Nachbarn hat.

So ändert sich das Feld von Generation zu Generation, mit einer meist faszinierenden Entwicklung. Das Spiel, erfunden von dem Mathematiker John Horton Conway (Cambridge), stellt sich als höchst komplex und reizvoll heraus.

Um es gleich zu sagen: Life ist kein Spiel, das man selber spielt. Life spielt sich selbst. Man kann die »Anfangswelt« fest-



Bild 1. Ein einfaches Beispiel erläutert die Funktionsweise von »Life«



Gleiterkanone (Zyklus ca. 30 Generationen)

Kaleidoskop (3er Zyklus nach ca. 20 Generationen) Evolution (Endstadium nach ca. 100 Generationen)

Bild 2. Anhand einfacher Strukturen wird die Faszination von »Life« deutlich

legen und dann beobachten, wie sie sich entwickelt. Das Spiel wird sicher dem einen oder anderen Computer-Fan bekannt sein, da es ein ideales Problem für den Computer ist: Sehr einfache Regeln, aber viel Rechnerei.

In dem Beispiel (Rotor, siehe Bild 1) sind die Verhältnisse noch sehr einfach, weil die kleine Life-Welt nur zwischen zwei Zuständen hin und her pendelt. Dieser Dreier ist eine der Strukturen, die stabil sind. Meistens dauert es viele Generationen, bis eine Population ausstirbt oder in stabile Strukturen übergeht. Am besten, man experimentiert einfach etwas. Bild 2 zeigt noch einige interessante Strukturen.

Zum Programm:

Wie würde man das Problem normalerweise angehen? Nun, man legt sich ein zweidimensionales Feld an und speichert für jede lebende Zelle eine Eins, für jedes unbesetzte Kästchen eine Null. Dann geht man alle Felder durch, prüft die acht Nach-

Routinen des Maschinenprogramms

\$C000-C023	16-Bit-Multiplikation (Listing 2)
\$C024-C067	Rechnung: $6000 + 3(X + 40Y) \rightarrow SP(16B)$
\$C068-C079	Rechnung: $4S(16B) \rightarrow SP1(16B)$
\$C080-C097	Rechnung: $4M(16B) \rightarrow SP1(16B)$
\$C098-C0A5	Inkrement X Verändert SP(16B) entsprechend,
\$C0A6-C0B3	Inkrement Y so daß es dem Ergebnis der
\$C0B4-C0C1	Dekrement X Routine \$C024 entspricht.
\$C0C2-C0CF	Dekrement Y
\$C0D0-C0DC	erhöht M, falls Zelle bei X,Y lebt
\$C0DD-C104	relevanter Bereich, speichert Koordinaten
\$C105-C15C	relevanter Bereich, geht neun Umgebungszellen durch, ruft \$C0DD auf
\$C15D-C2A4	nächste Generation, läuft zweimal durch alle Koordinaten des relevanten Bereichs
\$C2A5-C332	Plot: setzt, löscht und prüft Punkte im 80 x 50-Bild
\$C333-C3A6	Initialisierungsroutine
\$C3A7-C3C4	Aufruf relevanter Bereich, setzt $A\%(X,Y)=1$, trägt Koordinaten der Umgebungsfelder ein (SYS C3A7,X,Y)
\$C3C5-C3F6	Aufruf Plot, SYS C3C5,X,Y,Z, löscht, setzt (Z30 = oder 1) oder kontrolliert (Z=2) Punkte
\$C3F7-C3FC	Aufruf nächste Generation, ruft \$C15D auf, setzt zwecks Beschleunigung Interrupt-Flag
\$C3FD-C403	Aufruf maximale Geschwindigkeit, setzt I-Flag, ruft in Endlosschleife \$C15D

Variablen des Maschinenprogramms

\$C404-C413	Tabelle Binärzahlen \rightarrow ASCII-Code (der Grafikzeichen für Plot)
\$C414	Z (siehe \$C3C5), enthält wegen logischen Verknüpfungen 00 für setzen und 0F für löschen
\$C415	L gibt an, welches Bit des Grafikzeichens angesprochen wird
\$C416	QQ liefert Ergebnis des Tests
\$C417/8	X/Y
\$C419/B	Zwischenspeicher 1/2
\$C41A	E gibt an, welcher $A\%$ Speicherbedarf angesprochen wird
\$C41C/D	M(16B) Zahl der Koordinaten be-
\$C41E/F	S(16B) ziehungsweise Schleifenindex
\$C420	M Zahl der lebenden Zellen in jeweiliger Umgebung
\$C421	J gibt an, welcher $X\%,Y\%$ Speicherbetrieb angesprochen wird
\$C422-C521	Tabelle ASCII-Code \rightarrow Binärzahlen (wird erzeugt)
\$57-5B	für 16-Bit-Multiplikation $(5B) \times (59/5A) = (57/58)$
\$57/58	SP(16B) (lokal)
\$59/5A	SP1(16B) (lokal)

S(16B) bedeutet Speicher S mit der Länge von 16 Bit. X und Y haben nichts mit den Registern zu tun, die Namen der Variablen beziehen sich auf das Basic-Programm (Listing 1)

Tabelle 1. Der Aufbau des Maschinenprogramms (Listing 2)

barfelder und speichert das Ergebnis, tot oder lebendig in der nächsten Generation, in einem zweiten Feld. Der Computer hat dann (bei 80 mal 50) 32 000 Felder zu überprüfen, und das bedeutet: warten ...

Aber Moment! Ist es denn nötig, auch die Bereiche zu bearbeiten, in denen es ohnehin kein Leben gibt?

Besser, man bearbeitet jeweils nur die relevanten (belebten) Bereiche: Es kann nur in dem Bereich etwas passieren, der direkt an lebende Zellen angrenzt. Man speichert die Koordinaten aller lebenden Zellen und ihrer Nachbarn, sorgt dafür, daß keine Felder doppelt gezählt werden und braucht dann jeweils nur noch die Liste der Koordinaten durchzugehen. Dafür benötigt man ein weiteres Variablen-Feld, in dem bereits vermerkte Zellen gekennzeichnet werden. Nach dieser Vorbereitung werden zur Berechnung der nächsten Generation die Koordinaten des relevanten Bereichs durchgegangen. Ein zweites Mal durchläuft man die Koordinaten, um den neuen relevanten Bereich festzustellen. Wenn also eine Struktur auf dem Bildschirm wandert, so wandert auch der relevante Bereich mit.

Bedienungshinweise:

Man hat eine 78 x 48-Auflösung (der äußerste Rand ist eine Todeszone; hier ist grundsätzlich kein Leben möglich) zur Verfügung. Startet man das Programm, so befindet man sich in einem Editor.

Eine Zelle wird mit der SPACE-Taste lebendig.

Eine Zelle wird mit der DEL-Taste getötet.

Der Cursor schreitet nach SPACE und DEL in die Richtung fort, die die letzte gedrückte Cursortaste angab.

Mit »L« wird Life dann gestartet. In dem jetzigen Zustand gibt es folgende Befehle:

S Stop
F1-F8 Verzögerung
N Neue Eingabe
E Ergänzung, Veränderung

Q Quit, Programmende
P Power, maximale Geschwindigkeit
(nur mit RUN-STOP/RESTORE zu lösen)

Erweiterungsmöglichkeiten:

Man könnte natürlich die Auflösung weiter erhöhen. Da böte sich dann die Bitmap-Auflösung an (HiRes-Grafik). Hier müßte man allerdings mit dem Speicherplatz sehr vorsichtig umgehen, da der Bildschirm 64000 Bildpunkte hat. Die zusätzlichen Berechnungen würden das Programm auf jeden Fall langsamer machen. Außerdem wird, wenn die relevanten Bereiche zu groß werden, ein Speichern derselben unmöglich (Speicherplatz) und der Generationswechsel dauert zu lange. Eine andere interessante Veränderung wäre mit dem Rand durchzuführen. Mit dem »Todesrand« wurde das Problem der Bildschirm-Bereichsüberschreitung zwar recht elegant gelöst, denn es sind durch das Prinzip der relevanten Bereiche keine Abfragen notwendig. Interessanter jedoch wäre eine offene Wand oder gar eine ganz in sich geschlossene Welt. Unten (rechts) heraustretende Formen würden dann oben (links) wieder auftauchen. Wegen der Abfragen würde jedoch auch dies die Geschwindigkeit verringern.

Hinweise zum Abtippen:

Als erstes geben Sie die Maschinenroutine (Listing 2) mit dem MSE ein (Speichern nicht vergessen). Zur Eingabe des Hauptprogramms (Listing 1) verwenden Sie bitte den Checksummer.

Interessierte Leser finden in der Tabelle 1 die Beschreibung des Maschinenprogramms.

Wer will, kann das Programm noch wesentlich erweitern: Zum Beispiel könnten Hungersnöte, Naturkatastrophen oder Seuchen in bestimmten Gebieten die Bevölkerung wesentlich dezimieren. Ich glaube nicht, daß man das Spiel »Life« auf dem C 64 noch wesentlich schneller programmieren kann, jedoch soll diese Behauptung ruhig dazu provozieren, das Gegenteil zu beweisen.
(Jürgen Engbring/tr)

10 REM-----	<165>	1220 GOTO 1450	<112>
20 REM** LIFE FUER C-64 **	<182>	1300 SYS PL,X,Y,Z:Y=Y-1:CH=145	<124>
25 REM-----	<180>	1310 IF Y<0 THEN Y=49	<244>
30 REM** JUERGEN ENGBRING **	<067>	1320 GOTO 1450	<214>
40 REM** HENKELSHOF 5-7 **	<166>	1400 SYS PL,X,Y,Z:X=X-1:CH=157	<209>
50 REM** 5630 REMSCHEID-11 **	<195>	1410 IF X<0 THEN X=79	<214>
60 REM** TEL. 02191/65533 **	<215>	1420 GOTO 1450	<058>
70 REM-----	<225>	1450 SYS PL,X,Y,2:Z=PEEK(QQ)	<197>
80 M=49152:FOR I=0 TO 9:READ A	<164>	1460 GOTO 1020	<194>
85 IF A<>PEEK(M) THEN LOAD"MASCHLIFE",8,1	<171>	1500 Z=0:A=CH:POKE BA+3*X+Y*240,0	<110>
90 M=M+100:NEXT	<095>	1510 GOTO 1060	<116>
95 DATA 169,88,87,206,2,152,0,192,45,0	<098>	1600 Z=1:A=CH	<032>
100 PRINT"CLR"	<088>	1610 SYS G,X,Y:GOTO 1060	<137>
110 SYS 49971: REM INITIALISIERUNG	<126>	1700 SYS PL,X,Y,Z	<059>
120 PL=50117: REM SYSPL,X,Y,0/1/2	<105>	2000 T=9999: REM MINIMALE VERZOEGERUNG	<014>
130 :N=50167: REM NAECHSTE GENERATION	<061>	2005 POKE 50209,2: REM VOR ERSTEM SYSN	<081>
140 :S=50173: REM MAXIMALE GESCHW.	<163>	2010 GET A\$: IF TI-Q>T THEN Q=TI: SYS N	<011>
150 :G=50087: REM SYSG,X,Y ZELLE LEBT	<179>	2020 IF A\$="" THEN 2010	<178>
155 BA=6*4096: REM BASIS ZELLSPEICHER	<247>	2030 IF A\$=" " THEN SYS N:T=9999:GOTO 2010	<068>
160 QQ=50198: REM TESTBYTE SYSPL,X,Y,2	<179>	2040 IF A\$="S" THEN T=9999	<059>
170 REM+++++	<049>	2050 IF A\$="N" THEN RUN	<044>
990 REM== 80 X 40 ZELLEDITOR ==	<080>	2055 IF A\$="E" GOTO 2300	<176>
1000 DIM G%(255):X=39:Y=24:Q=TI:CH=29	<129>	2060 IF A\$="Q" THEN END	<121>
1005 FOR I=0 TO 6:READ A:READ B:G%(A)=B:NE	<205>	2065 IF A\$="P" THEN SYS S	<038>
XT		2067 IF A\$=CHR\$(133) GOTO 2200	<086>
1010 DATA 17,1,29,2,145,3,157,4,20,5,32,6,	<170>	2070 A=ASC(A\$)	<188>
76,7		2080 IF A<134 OR A>140 GOTO 2010	<017>
1020 GET A\$: IF TI-Q>15 THEN Q=TI:W=1-W	<118>	2090 T=2*2+(A-134)	<159>
1030 SYS PL,X,Y,W	<141>	2100 GOTO 2010	<048>
1040 IF A\$="" THEN 1020	<222>	2200 GET A\$: IF A\$="" THEN SYS N:GOTO 2200	<155>
1050 A=ASC(A\$)	<184>	2210 GOTO 2020	<190>
1060 ON G%(A) GOTO 1100,1200,1300,1400,1500		2300 IF PEEK(50202)=1 THEN SYS N	<063>
,1600,1700	<079>	2310 POKE 50209,0:GOTO 1020	<141>
1070 GOTO 1020	<058>		
1100 SYS PL,X,Y,Z:Y=Y+1:CH=17	<088>		
1110 IF Y>49 THEN Y=0	<150>		
1120 GOTO 1450	<012>		
1200 SYS PL,X,Y,Z:X=X+1:CH=29	<013>		
1210 IF X>79 THEN X=0	<081>		

© 64'er

Listing 1. Der Basic-Teil von »Life« ist am besten mit dem Checksummer einzugeben