Life

The Game of Life was originally described in *Scientific American*, October 1970, in an article by Martin Gardner. The game itself was originated by John Conway of Gonville and Caius College, University of Cambridge, England.

In the "manual" game, organisms exist in the form of counters (chips or checkers) on a large checkerboard and die or reproduce according to some simple genetic rules. Conway's criteria for choosing his genetic laws were carefully delineated as follows:

1. There should be no initial pattern for which there is a simple proof that the population can grow without limit.
2. There should be initial patterns that apparently do grow without limit.
3. There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in three possible ways: fading away completely (from overcrowding or from becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering an oscillating phase in which they repeat an endless cycle of two or more periods.

In brief, the rules should be such as to make the behavior of the population relatively unpredictable. Conway's genetic laws are delightfully simple. First note that each cell of the checkerboard (assumed to be an infinite plane) has eight neighboring cells, four adjacent orthogonally, four adjacent diagonally. The rules are:

1. Survivals. Every counter with two or three neighboring counters survives for the next generation.
2. Deaths. Each counter with four or more neighbors dies (is removed) from overpopulation. Every counter with one neighbor or none dies from isolation.
3. Births. Each empty cell adjacent to exactly three neighbors — no more, no fewer — is a birth cell. A counter is placed on it at the next move.

It is important to understand that all births and deaths occur simultaneously. Together they constitute a single generation or, as we shall call it, a "move" in the complete "life history" of the initial configuration.

You will find the population constantly undergoing unusual, sometimes beautiful and always unexpected change. In a few cases the society eventually dies out (all counters vanishing), although this may not happen until after a great many generations. Most starting patterns either reach stable figures — Conway calls them "still lifes" — that cannot change or patterns that oscillate forever. Patterns with no initial symmetry tend to become symmetrical. Once this happens the symmetry cannot be lost, although it may increase in richness.

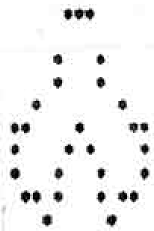
Conway used a DEC PDP-7 with a graphic display to observe long-lived populations. You'll probably find this more enjoyable to watch on a CRT than a hard-copy terminal.

Since MITS 8K BASIC does not have LINE INPUT, to enter leading blanks in the pattern, type a "." at the start of the line. This will be converted to a space by BASIC, but it permits you to type leading spaces. Typing DONE indicates that you are finished entering the pattern. See sample run.

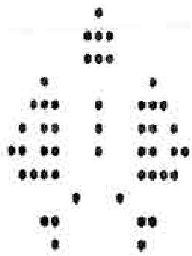
Clark Baker of Project DELTA originally wrote this version of LIFE which was further modified by Steve North of Creative Computing.

```
                     LIFE
CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY

ENTER YOUR PATTERN:
? .  ***
? . *  *
? .*   *
? DONE

GENERATION: 0                 POPULATION: 7

 ***
 *  *
*   *

GENERATION: 1                 POPULATION: 8

  *
 ***
** **

GENERATION: 2                 POPULATION: 9

 ***
*   *
** **

GENERATION: 3                 POPULATION: 10

  *
 ***
*   *
** **

GENERATION: 4                 POPULATION: 12

 ***
 ***
*   *
** **
```

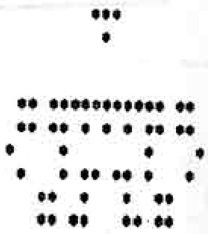
GENERATION: 13 POPULATION: 30



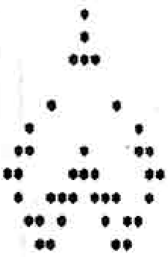
GENERATION: 16 POPULATION: 48



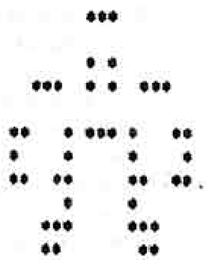
GENERATION: 19 POPULATION: 56



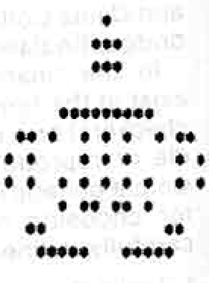
GENERATION: 14 POPULATION: 39



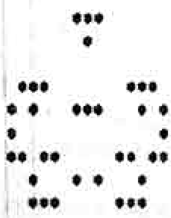
GENERATION: 17 POPULATION: 46



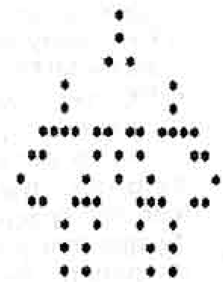
GENERATION: 20 POPULATION: 61



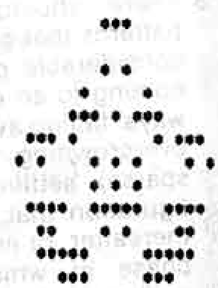
GENERATION: 15 POPULATION: 37



GENERATION: 18 POPULATION: 54



GENERATION: 21 POPULATION: 67



```
2 PRINT TAB(34);"LIFE"
4 PRINT TAB(15);"CREATIVE COMPUTING  MORRISTOWN, NEW JERSEY"
6 PRINT: PRINT: PRINT
8 PRINT "ENTER YOUR PATTERN:"
9 X1=1: Y1=1: X2=24: Y2=70
10 DIM A(24,70),B$(24)
20 C=1
30 INPUT B$(C)
40 IF B$(C)="DONE" THEN B$(C)="": GOTO 80
50 IF LEFT$(B$(C),1)="." THEN B$(C)=" "+RIGHT$(B$(C),LEN(B$(C))-1)
60 C=C+1
70 GOTO 30
80 C=C-1: L=0
90 FOR X=1 TO C-1
100 IF LEN(B$(X))>L THEN L=LEN(B$(X))
110 NEXT X
120 X1=11-C/2
130 Y1=33-L/2
140 FOR X=1 TO C
150 FOR Y=1 TO LEN(B$(X))
160 IF MID$(B$(X),Y,1)<>" " THEN A(X1+X,Y1+Y)=1:P=P+1
170 NEXT Y
180 NEXT X
200 PRINT:PRINT:PRINT
210 PRINT "GENERATION:";G,"POPULATION:";P;: IF I9 THEN PRINT "INVALID";
215 X3=24: Y3=70: X4=1: Y4=1: P=0
220 G=G+1
225 FOR X=1 TO X1-1: PRINT: NEXT X
230 FOR X=X1 TO X2
240 PRINT
250 FOR Y=Y1 TO Y2
253 IF A(X,Y)=2 THEN A(X,Y)=0: GOTO 270
256 IF A(X,Y)=3 THEN A(X,Y)=1: GOTO 261
260 IF A(X,Y)<>1 THEN 270
261 PRINT TAB(Y);"*";
262 IF X<X3 THEN X3=X
264 IF X>X4 THEN X4=X
266 IF Y<Y3 THEN Y3=Y
268 IF Y>Y4 THEN Y4=Y
270 NEXT Y
290 NEXT X
295 FOR X=X2+1 TO 24: PRINT: NEXT X
299 X1=X3: X2=X4: Y1=Y3: Y2=Y4
301 IF X1<3 THEN X1=3: I9=-1
303 IF X2>22 THEN X2=22: I9=-1
305 IF Y1<3 THEN Y1=3: I9=-1
307 IF Y2>68 THEN Y2=68: I9=-1
309 P=0
500 FOR X=X1-1 TO X2+1
510 FOR Y=Y1-1 TO Y2+1
520 C=0
530 FOR I=X-1 TO X+1
540 FOR J=Y-1 TO Y+1
550 IF A(I,J)=1 OR A(I,J)=2 THEN C=C+1
560 NEXT J
570 NEXT I
580 IF A(X,Y)=0 THEN 610
590 IF C<3 OR C>4 THEN A(X,Y)=2: GOTO 600
595 P=P+1
600 GOTO 620
610 IF C=3 THEN A(X,Y)=3: P=P+1
620 NEXT Y
630 NEXT X
635 X1=X1-1: Y1=Y1-1: X2=X2+1: Y2=Y2+1
640 GOTO 210
650 END
```