An invitation to Conway’s Game of Life

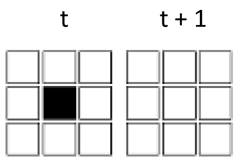
Group 43 08630404

Life was invented by John H. Conway in 1970. It is a zero-player game, played on an infinite two-dimensional grid of cells, each of which is either ‘alive’ (occupied) or ‘dead’ (empty). The state of each cell at discrete time t + 1 is determined by the states of its eight neighbours at time t. As time goes, each cell switches its state by a set of rules. The rules by which individual cell interacts are simple, but the population as a whole often exhibit complex and unexpected behaviour.

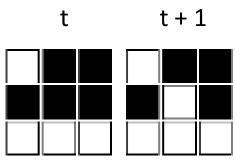
Rules of the space of *Life*

All cells die, survive or come back to life according to four rules:

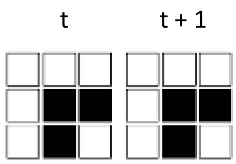
1. Loneliness: any live cell dies with fewer than 2 neighbours alive



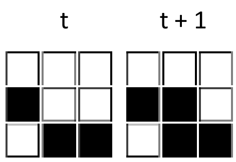
2. Overcrowding: any live cell dies with 4 or more neighbours alive



3. Stasis: any live cell, with 2 or 3 neighbours alive, stays alive



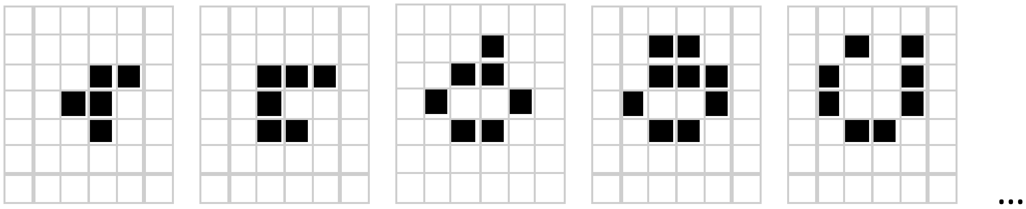
4. Reproduction: any dead cell, with exactly 3 neighbours alive, comes alive



An initial pattern of live cells constitutes the first generation of cells. Afterwards, successive generations are created by simultaneously applying the four rules to each cell at discrete time steps repeatedly. For example, if we start out the game with an R-Pentomino, we will observe the population evolve as illustrated:

R-Pentomino

Generation 1 Generation 2 Generation 3 Generation 4 Generation 5



When Conway first developed *Life*, he ran the game all by pen and paper. He found that every pattern he started out with eventually stabilised into a few recognisable patterns. This led Conway to conjecture that pattern can grow without limit. But then he came to R-Pentomino, which seemed a pattern that grows forever.

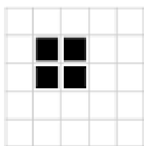
Study *Life* Patterns

There are three basic classes of patterns in *Life*: still life, oscillator and spaceship.

Still life

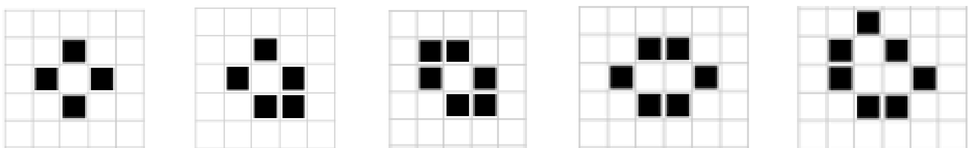
A stills life is a pattern which does not change from one generation to the next. The most common one is named Block. In Block, every live cell has 3 neighbours, and every dead cell has no more than 2 neighbours alive.

Block

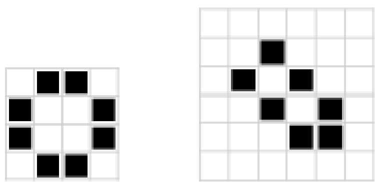


Still lives are equilibrium patterns. Frequently occurred still lives in *Life* also include:

Tub Boat Ship Beehive Loaf



Pond Long Boat

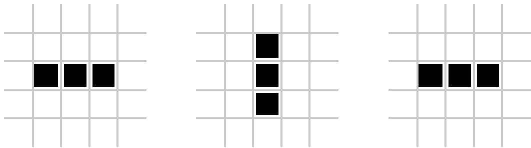


Oscillator

An oscillator is a pattern that does change but repeat itself in the same location after certain generations. Common oscillators include Blinker, Toad, Beacon, and even CP-Pulsar.

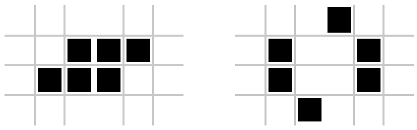
Blinker (period-2)

Generation 1 Generation 2 Generation 3



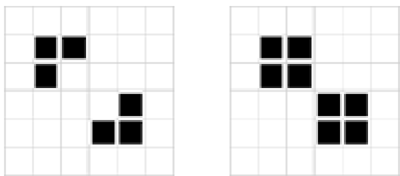
Toad

Period 1 Period 2



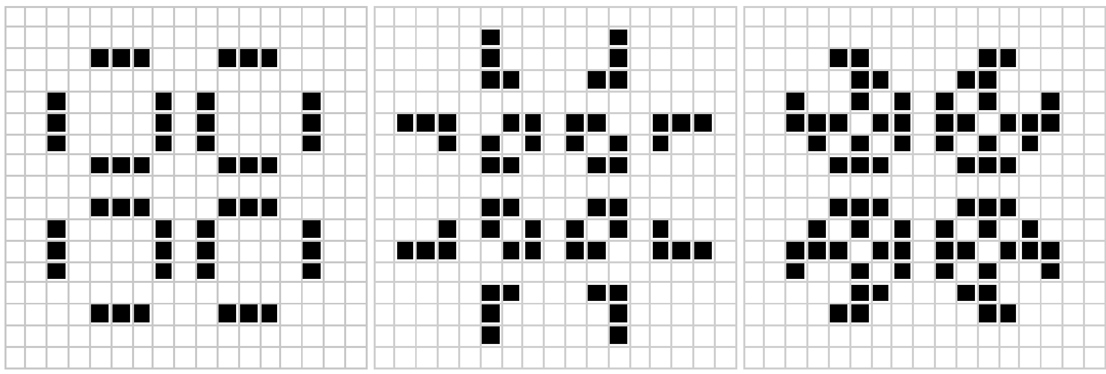
Beacon

Period 1 Period 2



CP-Pulsar

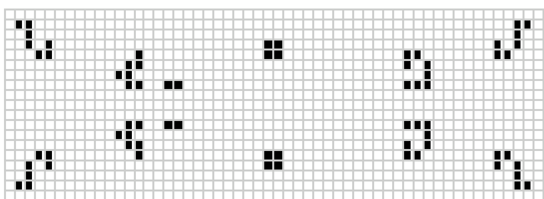
Period 1 Period 2 Period 3



Some of the oscillators discovered are exotic, such as the period-100 Centinal:

Centinal

Period 1

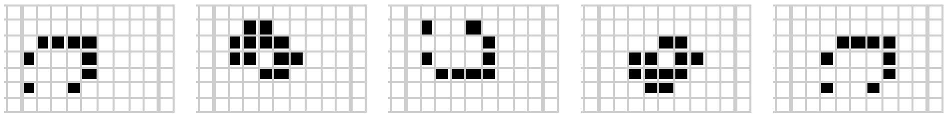


Spaceship

A spaceship is a pattern that repeats itself after certain generations, but translated in space. The first spaceship was discovered by Conway himself in 1970. It has the name Lightweight Spaceship (LWSS) and a period of 4.

LWSS

Period 1 Period 2 Period 3 Period 4 Period 5

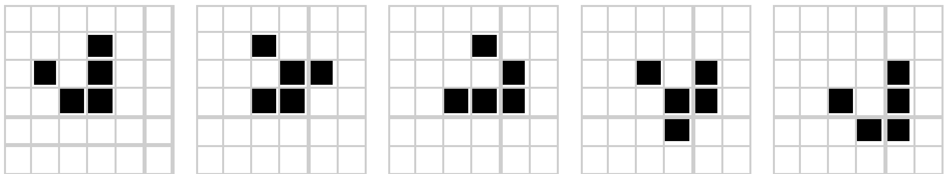


By the rule of reproduction, it is clear that the maximum translation speed attainable by any *Life* form is one cell per generation. This is defined to be the speed of light c. We can express the speed of a spaceship as a fraction of c. For LWSS, its speed is c /2.

When Conway tracked the evolution of R-Pentomino, he found it to shoot off 'Gliders' away from the central pattern. They are spaceships that travel at c /4 diagonally.

Gilder

Period 1 Period 2 Period 3 Period 4 Period 1



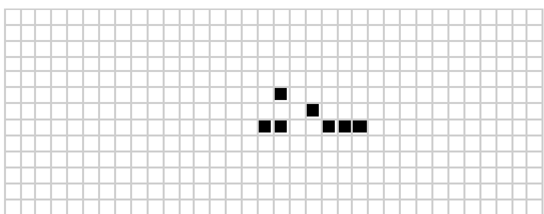
***Life* forms sophistication**

Methuselah

It turned out that R-Pentomino was neither a still life, nor an oscillator, nor a spaceship. For now R-Pentomino is classified as a Methuselah. Methuselah is any small seed pattern in *Life* that does not stabilise for a very long time. In fact, R-Pentomino runs 1103 steps before settling down into 6 Gliders, 8 Blocks, 4 Blinkers, 4 Beehives, 1 Boat, 1 Ship, and 1 Loaf. Acorn is another remarkable Methuselah. It is only 7-cell, but it takes 5206 generations to settle down to 633 lives cells and 13 escaping Gliders.

Acorn

Generation 1

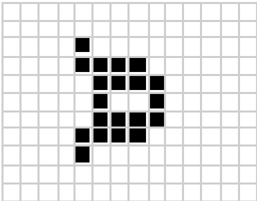


Glider Gun

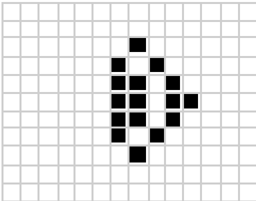
Glider gun is any pattern that grows infinitely by emitting Gliders. Gosper Gun was the first glider gun discovered. To construct a Gosper Gun, we start with Queen Bee:

Queen Bee

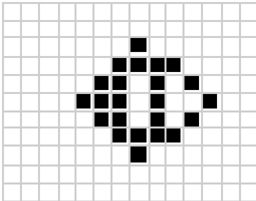
Generation 1



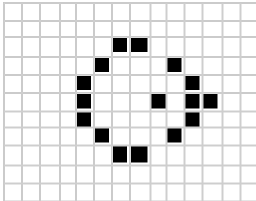
Generation 5



Generation 7

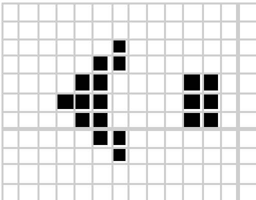


Generation 8

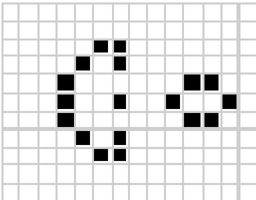


Queen Bee moves to the right until generation 8, it splits into two part and starts travelling to the opposite direction.

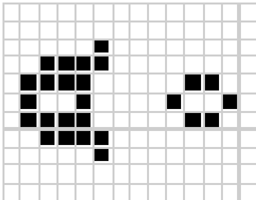
Generation 9



Generation 10

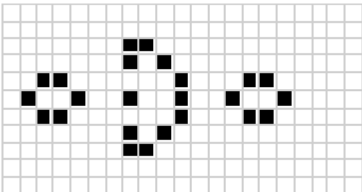


Generation 16

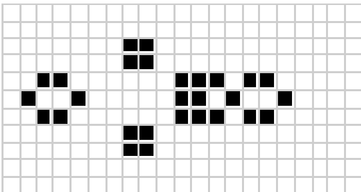


Queen Bee has left a Beehive behind. At generation 16, the original Bee returns. It should repeat the process, travel to the left, leave another Beehive, and then turn back.

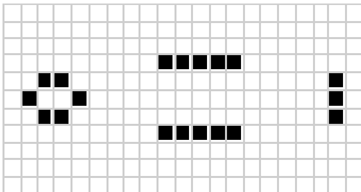
Generation 25



Generation 28



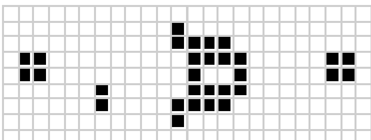
Generation 46



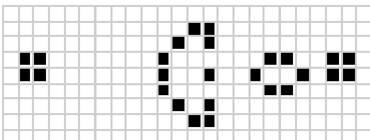
The Queen Bee crashes with Beehive and patterns collapsed. The solution is Queen Bee Shuttle. Blocks are added to both sides. They remove Beehives without permanent damage.

Queen Bee Shuttle

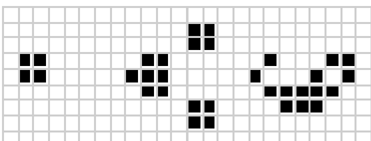
Generation 1



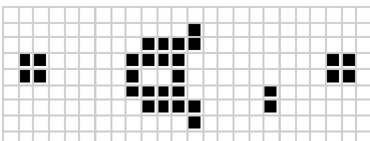
Generation 10



Generation 13



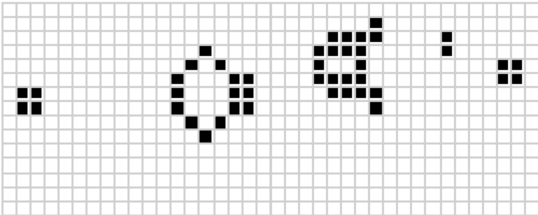
Generation 16



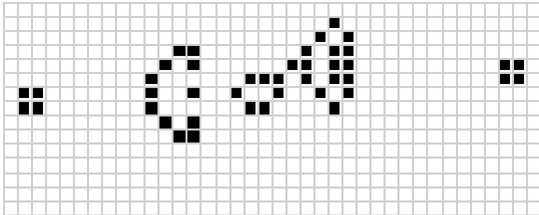
Queen Bee Shuttle is a period-30 oscillator that moves back and forth two Blocks. It forms the basis of the Gosper Gun:

Gosper Gun

Generation 1

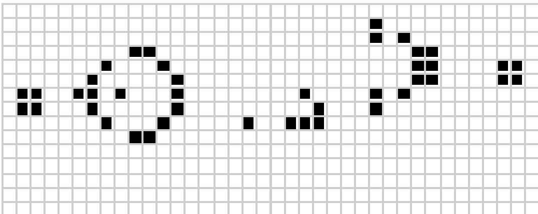


Generation 10

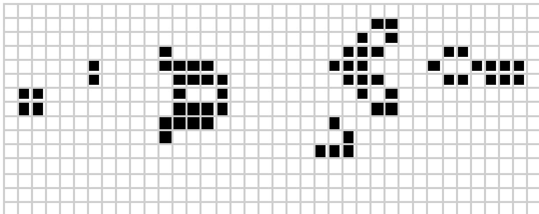


Two Queen Bees Shuttle bounce back and forth. At generation 10, the left one leaves a Beehive at the centre and is moving off. The right Queen Bee collides with that Beehive.

Generation 18



Generation 26



The collision is messy. But surprisingly after 8 generations, a Glider is produced. At generation 26, the Queen Bees are moving to centre again. It is clear that one is going to leave a Beehive, and the other to collide with it and shoot off the next Gilder. Fortunately every Glider escapes just in time.

Life is a Universal Computer

Actually, Conway intended to make *Life* an abstract machine capable of any computation, at the same time based on a set of simple states and states transition rules. Conway expected *Life* to be system to behave somewhere between steadiness and chaos. He chose the rules of *Life* with care and a long period of experimentation, to meet three criteria:

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.

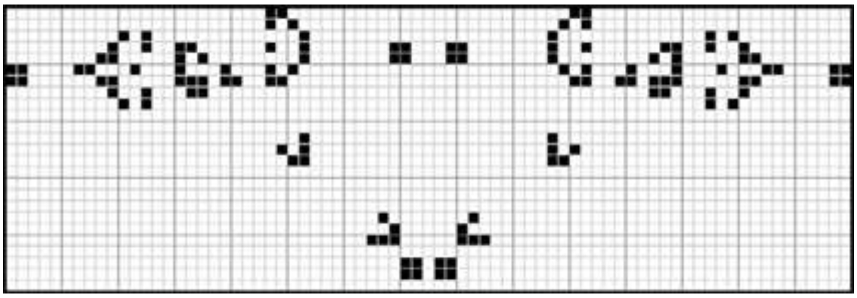
3. There should be simple initial patterns that grow and change for a considerable period of time before coming to end in three possible ways: fading away completely (from overcrowding or becoming too sparse), settling into a stable configuration that remains unchanged thereafter, or entering an oscillating phase in which they repeat an endless cycle of two or more periods.

Tentative logic implementation

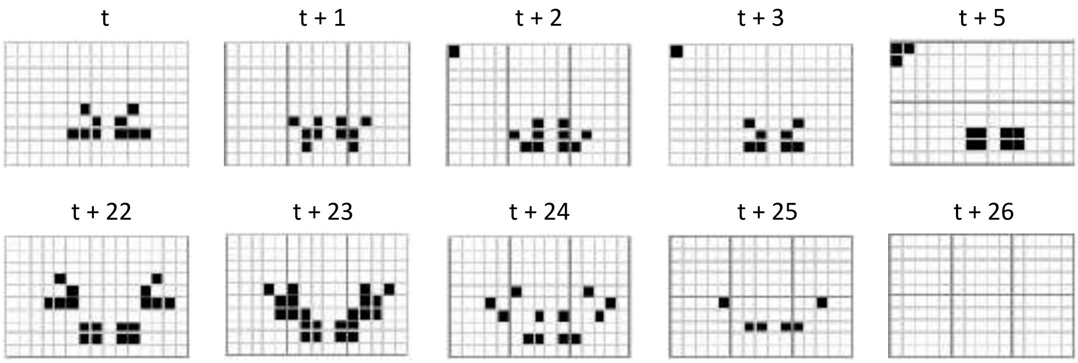
Of course, *Life* is not an advisable platform to implement a general purpose computer. But it is a revelation to prove such possibility.

To construct an abstract machine, the least we need is a scheme of information representation that serves as input and output, and a NOT gate plus an AND gate for processing. Binary data is simply represented by Gliders stream. A Glider represents 1; absence of a Glider represents 0. For instance, “Glider – no Glider” is binary 10.

Gate design is based on the fact that two Gliders can collide and annihilate each other. Gliders travel at diagonals. So we can position two Glider streams such that they collide at right angle.



a 90° binary Glider collision

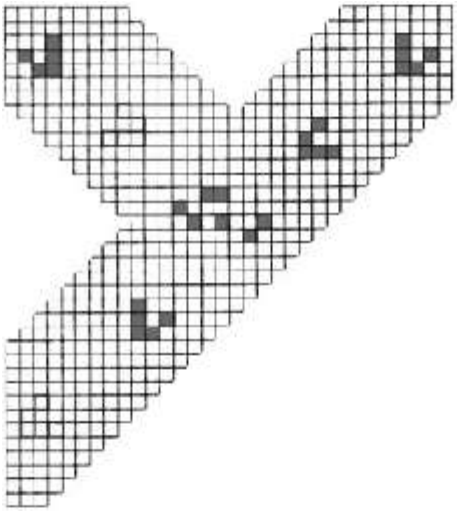


In this head-on collision, Gliders vanish within 27 generations. Suppose we use two Gosper Gun as Glider source. Since Gosper Gun is period-30, later Gliders arrive to the collision site only after the previous Gliders are completely annihiliated. In other words, it is ensured that Glider streams of period 30 are not interfered with debris of a Glider collision.

Then *Life*'s NOT gate is constructed as follows. A Gosper Gun is set up to supply Gliders in a specified direction. Input to the NOT gate are encoded in a second periodic Glider stream, aimed at right angle to the first one. The Input stream intersects the Gosper-Gun stream. That is to say a Glider in the input stream punches a hole in the Gosper-Gun stream, converting a Glider (1) into a non-Glider (0) in the process of annihilation. The absence of a Glider in the input stream allows a Glider from Gosper Gun to pass through unmolested.

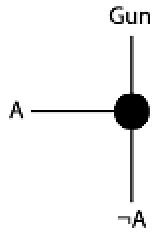
Life implementation of NOT gate

input Glider stream



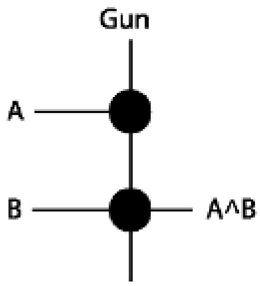
Gosper-Gun stream

NOT gate scheme



With the same principle, an AND gate is easily constructed.

NOT gate scheme



In principle, it has been proved that *Life* can solve any Boolean equation, so it is a universal computer. Certainly, actual logic implementation on *Life* is far more complex and optimized. Various glider guns, glider synthesis, ‘eater’, ‘relay’, ‘sliding block memory’, etc. are used to construct various gates and logic functions.

We have introduced two pictures of *Life*. In the first picture, we see *Life* as a complex system. Individual cells interact by a set of simple rules. *Life* forms arise. Then individual Life forms interact, the whole system evolves, and the final state is hardly predictable. *Life* forms so far presented is a pathetic deficit. New objects and interactions have been continually added to databases till now, 38 years from the genesis of Conway’s Game of Life. In the second picture, we see *Life* as a universal computer, that it is a mathematical construct that can be used to solve mathematical problems in return. There is a Halting Theorem which asserts that there is no algorithm for predicting when a computer will halt its execution of a given program. Given that *Life* is a universal computer, so that Halting Theorem applies, it means that in general no one can predict whether a particular initial pattern of live cells will eventually die out. There is no way to tell *Life*’s behaviour other than to actually play with it. Perhaps this is what makes *Life* so fun, rich and refreshing all the time.

References

Berlekamp, Elwyn R.; Conway, John H.; Guy, Richard. Winning Ways for your mathematical plays, Vol.2. Academic Press, 1982.

Calcagni, Carolyn; Chiu, David; Field, Kimen. The Game of Life. 15 November 2008 <http://cse.stanford.edu/class/sophomore-college/projects-01/cellular-automata/index.html>.

Dewdney, A.K. “Computer Recreations.” Scientific American 252 May 1985.

Flammenkamp, Achim. Most seen natural occuring ash objects in Game of Life. 7 September 2004. 15 November 2008 <http://wwwhomes.uni-bielefeld.de/achim/freq_top_life.html>.

Fornes, Marc. theverymany. 25 August 2006. 15 November 2008 <http://www.theverymany.net/2006_08_01_archive.html>.

Gardner, Martin. “Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life".” Scientific American 223 October 1970: 120-123.

Rennard, Jean-Philippe. “Implementation of logic functions in the Game of Life.”.

Weisstein, Eric W.. Eric Weisstein's Treasure Trove of the Life C.A. 15 November 2008 <http://www.ericweisstein.com/encyclopedias/life/Spaceship.html>.