

# Conway's Game of Life

The **Game of Life**, also known simply as **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970.<sup>[1]</sup>

The "game" is a zero-player game, meaning that its evolution is determined by its initial state, requiring no further input. One interacts with the Game of Life by creating an initial configuration and observing how it evolves.

## Rules

The universe of the Game of Life is an infinite two-dimensional orthogonal grid of square *cells*, each of which is in one of two possible states, *alive* or *dead*. Every cell interacts with its eight *neighbours*, which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

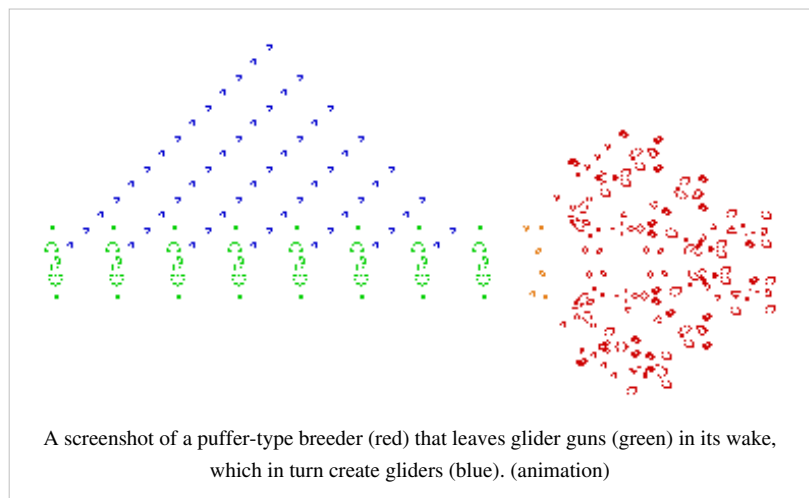
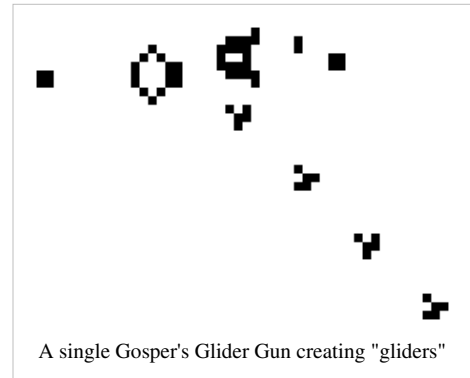
1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

The initial pattern constitutes the *seed* of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick* (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

## Origins

Conway was interested in a problem presented in the 1940s by mathematician John von Neumann, who attempted to find a hypothetical machine that could build copies of itself and succeeded when he found a mathematical model for such a machine with very complicated rules on a rectangular grid. The Game of Life emerged as Conway's successful attempt to drastically simplify von Neumann's ideas. The game made its first public appearance in the October 1970 issue of *Scientific American*, in Martin Gardner's "Mathematical Games" column. From a theoretical point of view, it is interesting because it has the power of a universal Turing machine: that is, anything that can be computed algorithmically can be computed within Conway's Game of Life.<sup>[2][3]</sup> Gardner wrote:

The game made Conway instantly famous, but it also opened up a whole new field of mathematical research, the field of cellular automata ... Because of Life's analogies with the rise, fall and alterations of a society of living organisms, it belongs to a growing class of what are called "simulation games"



(games that resemble real life processes).

Ever since its publication, Conway's Game of Life has attracted much interest, because of the surprising ways in which the patterns can evolve. Life provides an example of emergence and self-organization. It is interesting for computer scientists, physicists, biologists, biochemists, economists, mathematicians, philosophers, generative scientists and others to observe the way that complex patterns can emerge from the implementation of very simple rules. The game can also serve as a didactic analogy, used to convey the somewhat counter-intuitive notion that "design" and "organization" can spontaneously emerge in the absence of a designer. For example, philosopher and cognitive scientist Daniel Dennett has used the analogue of Conway's Life "universe" extensively to illustrate the possible evolution of complex philosophical constructs, such as consciousness and free will, from the relatively simple set of deterministic physical laws governing our own universe.<sup>[4][5][6]</sup>

The popularity of Conway's Game of Life was helped by its coming into being just in time for a new generation of inexpensive minicomputers which were being released into the market. The game could be run for hours on these machines, which would otherwise have remained unused at night. In this respect, it foreshadowed the later popularity of computer-generated fractals. For many, Life was simply a programming challenge; a fun way to use otherwise wasted CPU cycles. For some, however, Life had more philosophical connotations. It developed a cult following through the 1970s and beyond; current developments have gone so far as to create theoretic emulations of computer systems within the confines of a Life board.<sup>[7][8]</sup>

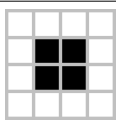
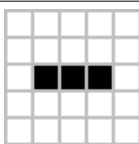
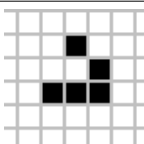
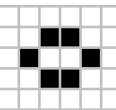
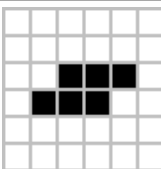
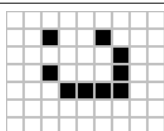
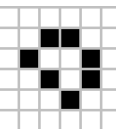
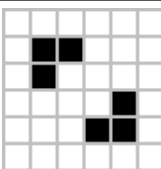
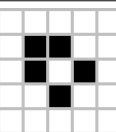
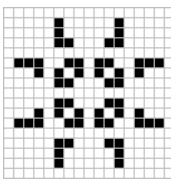
Conway chose his rules carefully, after considerable experimentation, to meet these criteria:

1. There should be no explosive growth, which is why 34 Life was rejected.
2. There should exist small initial patterns with chaotic, unpredictable outcomes.
3. There should be potential for von Neumann universal constructors.
4. The rules should be as simple as possible, whilst adhering to the above constraints.<sup>[9]</sup>

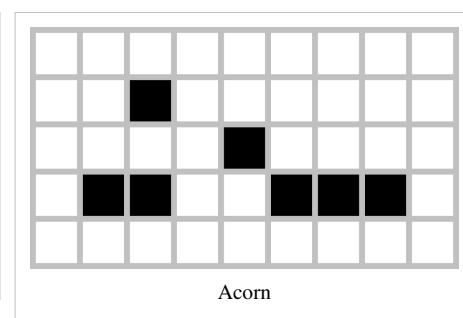
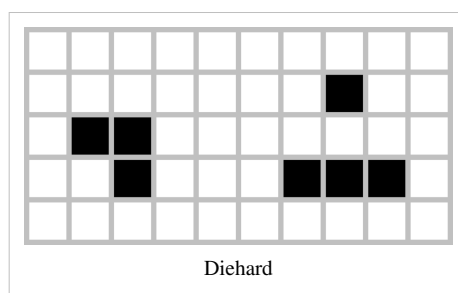
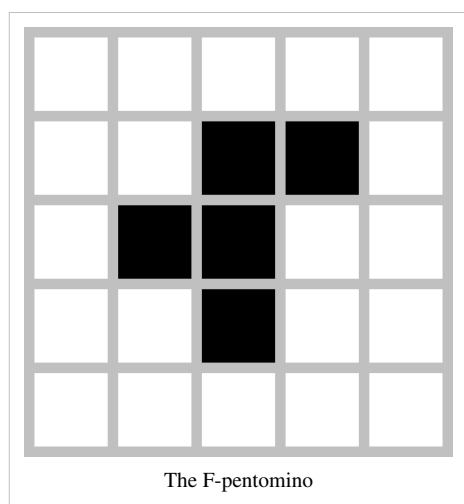
## Examples of patterns

The earliest interesting patterns in the Game of Life were discovered without the use of computers. The simplest static patterns ("still lifes") and repeating patterns ("oscillators"—a superset of still lifes) were discovered while tracking the fates of various small starting configurations using graph paper, blackboards, physical game boards (such as Go) and the like. During this early research, Conway discovered that the F-pentomino (which he called the "R-pentomino") failed to stabilize in a small number of generations. In fact, it takes 1103 generations to stabilize, by which time it has a population of 116 and has fired six escaping gliders<sup>[10]</sup> (these were the first gliders ever discovered).<sup>[11]</sup>

Many different types of patterns occur in the Game of Life, including still lifes, oscillators, and patterns that translate themselves across the board ("spaceships"). Some frequently occurring<sup>[12]</sup> examples of these three classes are shown below, with live cells shown in black, and dead cells shown in white.

Still lifes		Oscillators		Spaceships	
Block		Blinker (period 2)		Glider	
Beehive		Toad (period 2)		Lightweight spaceship (LWSS)	
Loaf		Beacon (period 2)			
Boat		Pulsar (period 3)			

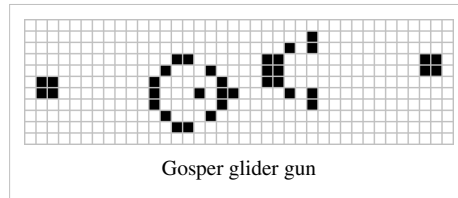
The "pulsar"<sup>[13]</sup> is the most common period 3 oscillator. The great majority of naturally occurring oscillators are period 2, like the blinker and the toad, but oscillators of all but finitely many periods are known to exist,<sup>[14]</sup> and oscillators of periods 4, 8, 14, 15, 30 and a few others have been seen to arise from random initial conditions.<sup>[15]</sup> Patterns called "Methuselahs" can evolve for long periods before stabilizing, the first-discovered of which was the F-pentomino. "Diehard" is a pattern that eventually disappears (rather than merely stabilize) after 130 generations, which is conjectured to be maximal for patterns with seven or fewer cells.<sup>[16]</sup> "Acorn" takes 5206 generations to generate 633 cells including 13 escaped gliders.<sup>[17]</sup>



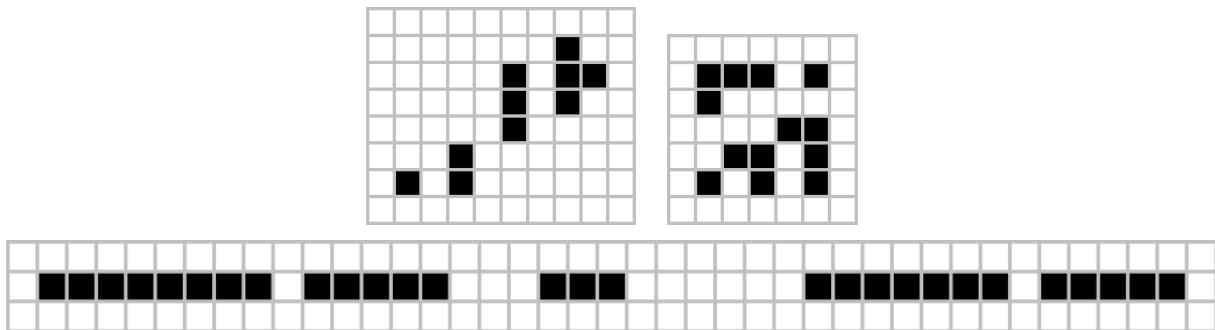
Conway originally conjectured that no pattern can grow indefinitely—i.e., that for any initial configuration with a finite number of living cells, the population cannot grow beyond some finite upper limit. In the game's original appearance in "Mathematical Games", Conway offered a \$50 prize to the first person who could prove or disprove the conjecture before the end of 1970. One way to disprove it would be to discover patterns that keep adding counters to the field: a "gun", which would be a configuration that repeatedly shoots out moving objects such as the

"glider". Another way of disproving the conjecture would be to construct a "puffer train", which would be a configuration that moves but leaves behind a trail of persistent "smoke".<sup>[18]</sup>

The prize was won in November of the same year by a team from the Massachusetts Institute of Technology, led by Bill Gosper; the "Gosper glider gun" shown below produces its first glider on the 15th generation, and another glider every 30th generation from then on. This first glider gun is still the smallest one known.<sup>[19]</sup>



Smaller patterns were later found that also exhibit infinite growth. All three of the following patterns grow indefinitely: the first two create one "block-laying switch engine"<sup>[20]</sup> each, while the third creates two. The first has only 10 live cells (which has been proven to be minimal).<sup>[21]</sup> The second fits in a  $5 \times 5$  square. The third is only one cell high:



Later discoveries included other "guns", which are stationary, and which shoot out gliders or other spaceships; "puffers", which move along leaving behind a trail of debris; and "rakes", which move and emit spaceships.<sup>[22]</sup> Gosper also constructed the first pattern with an asymptotically optimal quadratic growth rate, called a "breeder", or "lobster", which worked by leaving behind a trail of guns.

It is possible for gliders to interact with other objects in interesting ways. For example, if two gliders are shot at a block in just the right way, the block will move closer to the source of the gliders. If three gliders are shot in just the right way, the block will move farther away. This "sliding block memory" can be used to simulate a counter. It is possible to construct logic gates such as *AND*, *OR* and *NOT* using gliders. It is possible to build a pattern that acts like a finite state machine connected to two counters. This has the same computational power as a universal Turing machine, so the Game of Life is theoretically as powerful as any computer with unlimited memory and no time constraints: it is Turing complete.<sup>[23]</sup>

Furthermore, a pattern can contain a collection of guns that fire gliders in such a way as to construct new objects, including copies of the original pattern. A "universal constructor" can be built which contains a Turing complete computer, and which can build many types of complex objects, including more copies of itself.<sup>[23]</sup>

## Self-replication

On May 18, 2010, Andrew J. Wade announced a self-constructing pattern dubbed Gemini which creates a copy of itself while destroying its parent.<sup>[24][25]</sup> This pattern replicates in 34 million generations, and uses an instruction tape made of gliders which oscillate between two stable configurations made of Chapman-Greene construction arms. These, in turn, create new copies of the pattern, and destroy the previous copy. Gemini is a spaceship.<sup>[26]</sup>

## Iteration

From a random initial pattern of living cells on the grid, observers will find the population constantly changing as the generations tick by. The patterns that emerge from the simple rules may be considered a form of beauty. Small isolated subpatterns with no initial symmetry tend to become symmetrical. Once this happens, the symmetry may increase in richness, but it cannot be lost unless a nearby subpattern comes close enough to disturb it. In a very few cases the society eventually dies out, with all living cells vanishing, though this may not happen for a great many generations. Most initial patterns eventually "burn out", producing either stable figures or patterns that oscillate forever between two or more states;<sup>[27][28]</sup> many also produce one or more gliders or spaceships that travel indefinitely away from the initial location. Spaceships can travel at most one cell per unit time: so this velocity is set to be the cellular automaton speed of light and denoted  $c$ .

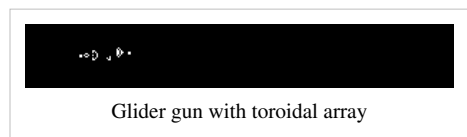
## Algorithms

Early patterns with unknown futures, such as the R-pentomino, led computer programmers across the world to write programs to track the evolution of Life patterns. Most of the early algorithms were similar; they represented Life patterns as two-dimensional arrays in computer memory. Typically two arrays are used, one to hold the current generation, and one in which to calculate its successor. Often 0 and 1 represent dead and live cells respectively. A nested for-loop considers each element of the current array in turn, counting the live neighbours of each cell to decide whether the corresponding element of the successor array should be 0 or 1. The successor array is displayed. For the next iteration the arrays swap roles so that the successor array in the last iteration becomes the current array in the next iteration.

A variety of minor enhancements to this basic scheme are possible, and there are many ways to save unnecessary computation. A cell that did not change at the last time step, and none of whose neighbours changed, is guaranteed not to change at the current time step as well. So, a program that keeps track of which areas are active can save time by not updating the inactive zones.<sup>[29]</sup>

In principle, the Life field is infinite, but computers have finite memory. This leads to problems when the active area encroaches on the border of the array. Programmers have used several strategies to address these problems. The simplest strategy is simply to assume that every cell outside the array is dead. This is easy to program, but leads to inaccurate results when the active area crosses the boundary. A more sophisticated trick is to consider the left and right edges of the field to be stitched together, and the top and bottom edges also, yielding a toroidal array. The result is that active areas that move across a field edge reappear at the opposite edge. Inaccuracy can still result if the pattern grows too large, but at least there are no pathological edge effects. Techniques of dynamic storage allocation may also be used, creating ever-larger arrays to hold growing patterns.

Alternatively, the programmer may abandon the notion of representing the Life field with a 2-dimensional array, and use a different data structure, like a vector of coordinate pairs representing live cells. This approach allows the pattern to move about the field unhindered, as long as the population does not exceed the size of the live-coordinate array. The drawback is that counting live neighbours becomes a search operation, slowing down simulation speed. With more sophisticated data structures this problem can also be largely solved.



For exploring large patterns at great time-depths, sophisticated algorithms such as Hashlife may be useful. There is also a method, applicable to other cellular automata too, for implementation of the Game of Life using arbitrary asynchronous updates whilst still exactly emulating the behaviour of the synchronous game.<sup>[30]</sup>

## Variations on Life

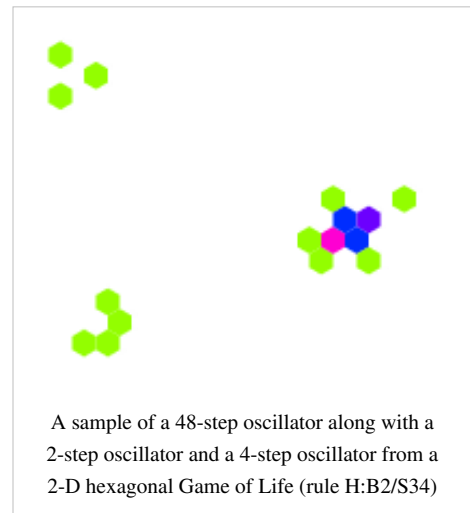
Since Life's inception, new similar cellular automata have been developed. The standard Game of Life, in which a cell is "born" if it has exactly 3 neighbours, stays alive if it has 2 or 3 living neighbours, and dies otherwise, is symbolised as "B3/S23". The first number, or list of numbers, is what is required for a dead cell to be born. The second set is the requirement for a live cell to survive to the next generation. Hence "B6/S16" means "a cell is born if there are 6 neighbours, and lives on if there are either 1 or 6 neighbours". Cellular automata on a two-dimensional grid that can be described in this way are known as Life-like cellular automata. Another common Life-like automaton, *HighLife*, is described by the rule B36/S23, because having 6 neighbours, in addition to the original game's B3/S23 rule, causes a birth. HighLife is best known for its frequently occurring replicators.<sup>[31][32]</sup> Additional Life-like cellular automata exist, although the vast majority of them produce universes that are either too chaotic or too desolate to be of interest.

Some variations on Life modify the geometry of the universe as well as the rule. The above variations can be thought of as 2-D square, because the world is two-dimensional and laid out in a square grid. 1-D square variations (known as elementary cellular automata)<sup>[33]</sup> and 3-D square variations have been developed, as have 2-D hexagonal and 2-D triangular variations.

Conway's rules may also be generalized such that instead of two states (*live* and *dead*) there are three or more. State transitions are then determined either by a weighting system or by a table specifying separate transition rules for each state; for example, Mirek's Celebration's multi-coloured "Rules Table" and "Weighted Life" rule families each include sample rules equivalent to Conway's Life.

Patterns relating to fractals and fractal systems may also be observed in certain Life-like variations. For example, the automaton B1/S12 generates four very close approximations to the Sierpiński triangle when applied to a single live cell. The Sierpiński triangle can also be observed in Conway's Game of Life by examining the long-term growth of a long single-cell-thick line of live cells,<sup>[34]</sup> as well as in HighLife, Seeds (B2/S), and Wolfram's Rule 90.<sup>[35]</sup>

*Immigration* is a variation that is very similar to Conway's Game of Life, except that there are two ON states (often expressed as two different colours). Whenever a new cell is born, it takes on the ON state that is the majority in the three cells that gave it birth. This feature can be used to examine interactions between spaceships and other "objects" within the game.<sup>[36]</sup> Another similar variation, called *QuadLife*, involves four different ON states. When a new cell is born from three different ON neighbours, it takes on the fourth value, and otherwise, like Immigration, it takes the majority value.<sup>[37]</sup> Except for the variation among ON cells, both of these variations act identically to Life.



A sample of a 48-step oscillator along with a 2-step oscillator and a 4-step oscillator from a 2-D hexagonal Game of Life (rule H:B2/S34)

## Entertainment

### Video Games

Some games for entertainment purposes have been developed from the Game of Life. One such game, for two players who each interact with the "game" once per tick, is based directly upon Conway's Game of Life. Live cells have one of two colours, and a player wins when all cells of the opponent's colour are eliminated. When a dead cell has three live neighbours, it becomes live in the same colour as the majority of its neighbours, as in the aforementioned *Immigration*. The game starts with a random or pre-chosen starting pattern with half the live cells of each colour. After one iteration, the first player may add one cell of his or her own colour and remove one cell of his or her opponent's colour. After the next iteration, the other player can do the same, and so forth. Other two-player variants of the Game of Life have also been developed.<sup>[38]</sup>

In *Populous II*, one of the 'divine intervention' effects is a fungus that grows according to the rules of the Game of Life.

Acclaimed game developer Jonathan Blow notes the game of life as a big influence on his design philosophy<sup>[39]</sup>

### Music

Various musical composition techniques use Conway's Game of Life, especially in MIDI sequencing.<sup>[40]</sup> A variety of programs exist for creating sound from patterns generated in the Game of Life. (see footnotes for links to examples.)<sup>[41][42][43]</sup>

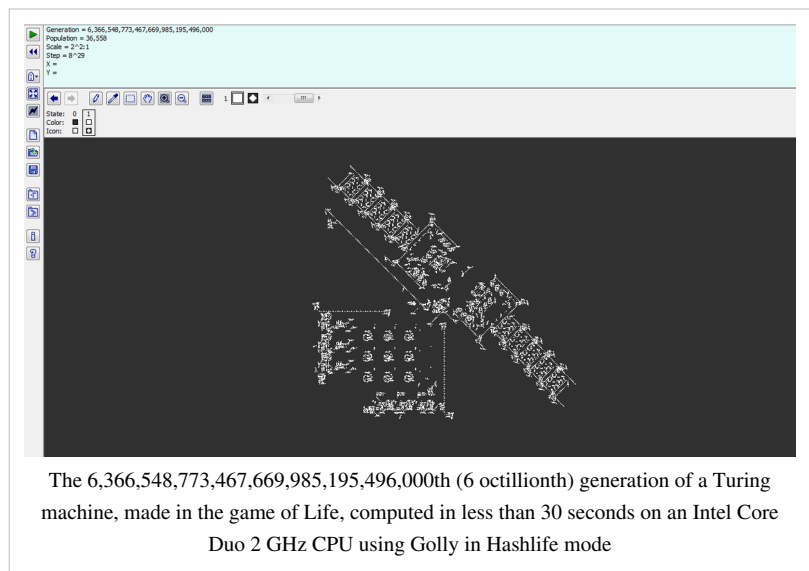
The Native Instruments modular sound generation/processing software package Reaktor features a drum machine with an integrated sequencer which implements the Game of Life.

## Notable Life programs

Computers have been used to follow Life configurations from the earliest days. When John Conway was first investigating how various starting configurations developed, he tracked them by hand using a Go board with its black and white stones. This was tedious, and prone to errors. The first-ever Life program was written by John Francis (an undergraduate student at Cambridge) on an IBM 360, and was used to automate this process and track the fate of the "R" pentomino for 1000 generations. The first interactive Life program was written in ALGOL

68 for the PDP-7 by M. J. T. Guy and S. R. Bourne. Without its help, some discoveries about the game would have been difficult to make. The results were published in the October 1970 issue of *Scientific American*.<sup>[44]</sup>

There are now thousands of Life programs on line, so a full list will not be provided here. The following is a selection of a small number of programs with some special claim to notability, such as popularity or unusual features. Most of these programs incorporate a graphical user interface for pattern editing and simulation, the capability for simulating multiple rules including Life, and a large library of interesting patterns in Life and other CA rules.



The 6,366,548,773,467,669,985,195,496,000th (6 octillionth) generation of a Turing machine, made in the game of Life, computed in less than 30 seconds on an Intel Core Duo 2 GHz CPU using Golly in Hashlife mode

- Conway's Game of Life <sup>[45]</sup>, by Alan Hensel. A pop-up Java web applet with fast simulation algorithms and a large library of interesting Life patterns.
- Golly <sup>[46]</sup>. A cross-platform (Windows, Macintosh and Linux) open-source simulation system for Life and other cellular automata, by Andrew Trevorrow and Tomas Rokicki. It includes the hashlife algorithm for extremely fast generation, and Perl or Python scriptability for both editing and simulation.
- Life32 <sup>[47]</sup>. Freeware for Windows machines, it includes powerful and scriptable pattern editing features.
- Mirek's Cellebration <sup>[48]</sup>. Free 1-D and 2-D cellular automata viewer, explorer and editor for Windows. Includes powerful facilities for simulating and viewing a wide variety of CA rules including Life, and a scriptable editor.
- Xlife <sup>[49]</sup>. A cellular-automaton laboratory by Jon Bennett. The standard Linux Life simulation application for a long time, it has also been ported to Windows. Can handle cellular automaton rules with the same neighbourhood as Life, and up to eight possible states per cell.

## References

- [1] Gardner, Martin (1970-10). *Mathematical Games - The fantastic combinations of John Conway's new solitaire game "life"* ([http://web.archive.org/web/20090603015231/http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis\\_projekt/proj\\_gamelif/ConwayScientificAmerican.htm](http://web.archive.org/web/20090603015231/http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelif/ConwayScientificAmerican.htm)). **223**. pp. 120–123. ISBN 0-89454-001-7. Archived from the original ([http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis\\_projekt/proj\\_gamelif/ConwayScientificAmerican.htm](http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelif/ConwayScientificAmerican.htm)) on 2009-06-03. . Retrieved 2011-06-26.
- [2] Paul Chapman (November 11, 2002). "Life Universal Computer" (<http://www.igblan.free-online.co.uk/igblan/ca/>). . Retrieved July 12, 2009.
- [3] Berlekamp, E. R.; ; Guy, R.K. (2001 2004). *Winning Ways for your Mathematical Plays* (2nd ed.). A K Peters Ltd. ISBN 978-1-56881-130-7; ISBN 1-56881-142-X; ISBN 1-56881-143-8; ISBN 1-56881-144-6
- [4] Dennett, D.C. (1991). *Consciousness Explained*. Boston: Back Bay Books. ISBN 0-316-18066-1
- [5] Dennett, D.C. (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life*. New York: Simon & Schuster. ISBN 0-684-82471-X
- [6] Dennett, D.C. (2003). *Freedom Evolves*. New York: Penguin Books. ISBN 0-14-200384-0
- [7] Paul Rendell (January 12, 2005). "A Turing Machine in Conway's Game of Life" (<http://rendell-attic.org/gol/tm.htm>). . Retrieved July 12, 2009.
- [8] Adam P. Goucher. "Spartan universal computer-constructor" ([http://www.conwaylife.com/wiki/index.php?title=Spartan\\_universal\\_computer-constructor](http://www.conwaylife.com/wiki/index.php?title=Spartan_universal_computer-constructor)). LifeWiki. . Retrieved July 12, 2009.
- [9] Conway, private communication to the 'Life list', 14 April 1999.
- [10] "R-pentomino" (<http://www.conwaylife.com/wiki/index.php?title=R-pentomino>). LifeWiki. . Retrieved July 12, 2009.
- [11] Stephen A. Silver. "Glider" ([http://www.argentum.freemove.co.uk/lex\\_g.htm#glider](http://www.argentum.freemove.co.uk/lex_g.htm#glider)). The Life Lexicon. . Retrieved July 12, 2009.
- [12] "Census Results in Conway's Game of Life" (<http://www.conwaylife.com/soup/census.asp?rule=B3/S23&sl=1&os=1&ss=1>). The Online Life-Like CA Soup Search. . Retrieved July 12, 2009.
- [13] "Pulsar" (<http://www.ericweisstein.com/encyclopedias/life/Pulsar.html>). Eric Weisstein's Treasure Trove of Life. . Retrieved 2008-09-16.
- [14] Game of Life Status page (<http://entropymine.com/jason/life/status.html>), Jason Summers, retrieved 2012-02-23.
- [15] Achim Flammenkamp (2004-09-07). "Most seen natural occurring ash objects in Game of Life" ([http://www.homes.uni-bielefeld.de/achim/freq\\_top\\_life.html](http://www.homes.uni-bielefeld.de/achim/freq_top_life.html)). . Retrieved 2008-09-16.
- [16] Stephen A. Silver. "Diehard" ([http://www.argentum.freemove.co.uk/lex\\_d.htm#diehard](http://www.argentum.freemove.co.uk/lex_d.htm#diehard)). The Life Lexicon. . Retrieved July 12, 2009.
- [17] Koenig, H. (February 21, 2005). "New Methuselah Records" ([http://pentadecathlon.com/lifeNews/2005/02/new\\_methuselah\\_records.html](http://pentadecathlon.com/lifeNews/2005/02/new_methuselah_records.html)). . Retrieved January 24, 2009.
- [18] Stephen A. Silver. "Puffer train" ([http://www.argentum.freemove.co.uk/lex\\_p.htm#puffertrain](http://www.argentum.freemove.co.uk/lex_p.htm#puffertrain)). The Life Lexicon. . Retrieved July 12, 2009.
- [19] Stephen A. Silver. "Gosper glider gun" ([http://www.argentum.freemove.co.uk/lex\\_g.htm#gosperglidergun](http://www.argentum.freemove.co.uk/lex_g.htm#gosperglidergun)). The Life Lexicon. . Retrieved July 12, 2009.
- [20] "Block-laying switch engine" ([http://www.conwaylife.com/wiki/index.php?title=Block-laying\\_switch\\_engine](http://www.conwaylife.com/wiki/index.php?title=Block-laying_switch_engine)). LifeWiki. . Retrieved July 12, 2009.
- [21] "Infinite Growth" (<http://www.ericweisstein.com/encyclopedias/life/InfiniteGrowth.html>). Eric Weisstein's Treasure Trove of Life. . Retrieved 2008-09-16.
- [22] Stephen A. Silver. "Rake" ([http://www.argentum.freemove.co.uk/lex\\_r.htm#rake](http://www.argentum.freemove.co.uk/lex_r.htm#rake)). The Life Lexicon. . Retrieved July 12, 2009.
- [23] Descriptions of these constructions are given in Berlekamp, E. R.; ; Guy, R.K. (2001 2004). *Winning Ways for your Mathematical Plays* (2nd ed.). A K Peters Ltd. ISBN 978-1-56881-130-7; ISBN 1-56881-142-X; ISBN 1-56881-143-8; ISBN 1-56881-144-6
- [24] <http://conwaylife.com/forums/viewtopic.php?f=2&t=399&p=2327#p2327>
- [25] <http://conwaylife.com/wiki/index.php?title=Gemini>
- [26] Aron, Jacob (16 June 2010). "First replicating creature spawned in life simulator". *New Scientist*



- [27] Andrzej Okrasinski. "Game of Life Object Statistics" (<http://web.archive.org/web/20090727010353/http://geocities.com/conwaylife/>). Archived from the original (<http://www.geocities.com/conwaylife/>) on 2009-07-27. . Retrieved July 12, 2009.
- [28] Nathaniel Johnston. "The Online Life-Like CA Soup Search" (<http://www.conwaylife.com/soup/>). . Retrieved July 12, 2009.
- [29] Alan Hensel. "About my Conway's Game of Life Applet" (<http://www.ibiblio.org/lifepatterns/lifeapplet.html>). . Retrieved July 12, 2009.
- [30] Nehaniv, Chrystopher L. (2002). "Self-Reproduction in Asynchronous Cellular Automata". *2002 NASA/DoD Conference on Evolvable Hardware (15–18 July 2002, Alexandria, Virginia, USA)*. IEEE Computer Society Press. pp. 201–209
- [31] HighLife - An Interesting Variant of Life (<http://www.tip.net.au/~dbell/articles/HighLife.zip>) by David Bell (.zip file)
- [32] Stephen A. Silver. "Replicator" ([http://www.argentum.freemove.co.uk/lex\\_r.htm#replicator](http://www.argentum.freemove.co.uk/lex_r.htm#replicator)). The Life Lexicon. . Retrieved July 12, 2009.
- [33] "Elementary Cellular Automaton" (<http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>). Wolfram Mathworld. . Retrieved July 12, 2009.
- [34] "One cell thick pattern" ([http://www.conwaylife.com/wiki/index.php?title=One\\_cell\\_thick\\_pattern](http://www.conwaylife.com/wiki/index.php?title=One_cell_thick_pattern)). LifeWiki. . Retrieved July 12, 2009.
- [35] "Life Imitates Sierpinski" (<http://www.conwaylife.com/forums/viewtopic.php?f=7&t=90>). ConwayLife.com forums. . Retrieved July 12, 2009.
- [36] "Immigration" (<http://www.ericweisstein.com/encyclopedias/life/Immigration.html>). Eric Weisstein's Treasure Trove of Life. . Retrieved 2008-09-16.
- [37] "QuadLife" (<http://www.ericweisstein.com/encyclopedias/life/QuadLife.html>). Eric Weisstein's Treasure Trove of Life. . Retrieved 2008-09-16.
- [38] Mark Levene; George Roussos (2002). "A Two-Player Game of Life". *International Journal of Modern Physics C [Computational Physics and Physical Computation]* **14** (2): 195. arXiv:cond-mat/0207679. doi:10.1142/S0129183103004346.
- [39] <http://braid-game.com/news/2010/02/a-new-short-speech-about-game-design/>
- [40] Burraston, Dave; Edmonds, Ernest (2004). "Cellular Automata in MIDI based Computer Music" (<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.6.3882&rep=rep1&type=pdf>). *Proceedings of the 2004 International Computer Music Conference*. doi:10.1.1.6.3882.
- [41] glitchDS – Cellular Automaton Sequencer For The Nintendo DS (<http://www.synthtopia.com/content/2008/05/29/glitchds-cellular-automaton-sequencer-for-the-nintendo-ds/>)
- [42] Game Of Life Music Sequencer (<http://www.synthtopia.com/content/2009/04/29/game-of-life-music-sequencer/>)
- [43] Game Of Life Music Sequencer For iOS, Runxt Life (<http://www.synthtopia.com/content/2011/01/12/game-of-life-music-sequencer-for-ios-runxt-life/>)
- [44] Gardner, Martin (October 1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"". *Scientific American* **223**: 120–123
- [45] <http://www.ibiblio.org/lifepatterns/>
- [46] <http://golly.sourceforge.net/>
- [47] <http://www.brothersoft.com/life32-download-298213.html>
- [48] <http://www.mirekw.com/ca/index.html>
- [49] <http://packages.debian.org/lenny/xlife>

## External links

- Conway's Game of Life ([http://www.dmoz.org/Computers/Artificial\\_Life/Cellular\\_Automata/Conway's\\_Game\\_of\\_Life/](http://www.dmoz.org/Computers/Artificial_Life/Cellular_Automata/Conway's_Game_of_Life/)) at the Open Directory Project
- Game of Life News (<http://pentadecathlon.com/lifeNews/index.php>)
- LifeWiki (<http://www.conwaylife.com/wiki/>)
- Cellular Automata FAQ — Conway's Game of Life (<http://cafaq.com/lifefaq/index.php>)

# Article Sources and Contributors

**Conway's Game of Life** *Source:* <http://en.wikipedia.org/w/index.php?oldid=487422600> *Contributors:* 213.253.40.xxx, ABoerma, ACW, Abelson, Adijk, Ae-a, Aesir, Agorf, Alessio88, AlphaPyro, Amakuru, Amux, Andrew.brusentsov, Andy120290, Andyk, Anomalocaris, AnonMoos, AnthonyQBachler, Aplusb, Arbol01, Ariel., Arto B, Ashley Pomeroy, Atakdoug, AugPi, Awaterl, B1u SkR33N, Baccyak4H, Baltar, Gaius, Banno, Barticus88, Betacommand, Bevo, Bitbutter, Bkell, Bloodrage, BooHamster, Bookandcoffee, BookgirlST, Borisblue, Bryan.burgers, Bubba73, Bud Charles, Butros, Cabbers, Caekaert, Calcyman, CanisRufus, Capricorn42, Captain-n00dle, Carey Evans, Charles Matthews, Chris 73, Christopher.Madsen, Chuck Smith, Ciphers, Clarince63, Closedmouth, D6, DFRussia, DIG, DMacks, DNewhall, DV8 2XL, DaGizza, Daelin, Damian Yerrick, Darth Mike, Darth Panda, Darthrevan1789, David Eppstein, David van bruwaene, DavidWBrooks, Davipo, Dbabbitt, Dclayh, Deadlyfish, Delldot, Diego, Dimitrii, Dirac1933, Discospinster, Dispenser, Dmitry123456, Doradus, Dori, DragonflySixtyseven, Ds13, Dvgrn, Dvorak729, Dxco, Dylan Lake, Dysprosia, E-trail, ESKog, Ecomba, Emurphy42, Eric119, Esger, EverGreg, Evercat, Excirial, Faisal.akeel, Ferkel, Ffransoo, FiP, Flewis, Fpfect111, Frecklefoot, Fredrik, Froese, Gaius Cornelius, Gdr, Geneffects, Gerbrant, Gevil, Giftlite, Gioto, Gleishma, Gorank4, Graham87, Graue, GregorB, Grontesca, Gulliveig, Gwern, HairyWombat, Headbomb, Heliac, Henrygb, HexaChord, Hillgentleman, Home Row Keysplurge, Hughsk, Hydrargyrum, Hyperdeath, Ihope127, Immunize, IntrigueBlue, Ioscus, Ipso2, Ixfd64, JTN, JYUuyang, Jarl Friis, Jbragadeesh, Jean-claude perez, Jim.belk, Jleedev, Jlopez1967, Jmundo, Joe92, John.constantine, Johnf.at.home, Johnuniq, JokeySmurf, Jpo, Jrockley, Julescubtree, Justin W Smith, Jwz, Kaldari, Kappa, Katalaveno, KenWalker, Kenirwin, Kieff, Kim Bruning, Kosmar, Kubigula, Kurykh, KymFarnik, LC, LOL, La goutte de pluie, Lanthanum-138, LegendGamer, LeonardoRob0t, Leonxlin, Lesnail, Lesonyrra, Livajo, Luca Edd Fike, M.nelson, Markaci, Markhurd, Marktompsett, Martarius, Marudubshinki, Maury Markowitz, McGeddon, Meggar, Melesse, MeltBanana, Michael L. Kaufman, MisfitToys, MisterSanderson, Mitsukai, Mk5384, Mkill, Moondoll, Mtfr, Muriel Gottrop, Narumara, Nealmb, NevilleDNZ, Nick Levine, NickNyman, Nikevich, Noe, Notheruser, NuclearCarnage, NuclearWarfare, Obrienmi8, Ogxela, Oliver Pereira, Opelio, Owain, P0lyglut, Pakaran, Palpalpalpal, Panoptical, Paul A, Paul August, Paul G, PauloCalipari, Pavel Jelínek, Pcr, Perl, Pharaoh of the Wizards, Phil Boswell, Pichote, Ploock, PuzzletChung, Python eggs, Quentin X, Quiddity, Quuxplusone, RDBury, Racklever, Raekwon, Randomphantom45, Rangek, Rat at WikiFur, Rbtmdl, Redslazer, RekishiEJ, Reonic, Retired username, Richard001, Robin klein, RodrigoCamargo, RogerBarnett, Rossumcapek, Rspeer, SDS, SGBailey, Sabbut, Salix alba, Sam Tobar, Sandman303, Satbir15, Savant13, ScAvenger, Sciguy47, Scote, Scuzzi, Shadowcheets, Shanes, Shellreef, Sheridan, Shizhao, Shoemaker's Holiday, Simpsons contributor, Sir Isaac, Sir Paul, Sjorford, Snoyes, Solipsist, Spellmaster, Srliefler, Standardfact, Stay cool, Steven Watson, Sverdrup, Tarquin, Tedweird, Tenbaset, The Anome, The Earwig, TheAllSeeingEye, Theusernameiwantedisalreadynuse, Throwaway85, Thumperward, Tim1988, Tkorrovi, Tom harrison, Tomyungoong, Trovatore, Trubyte, Turidoth, Twri, UBERNESS, Updatehelper, Usb10, UtherSRG, Val42, Versus22, Voidxor, Wavelength, Wayfarer, Weyes, Wiggy04, Wknight94, Wody1025, Xhin, Ynhockey, ZICO, Zdim, Zibart, Zundark, Zvika, 460 anonymous edits

# Image Sources, Licenses and Contributors

**File:Gospers glider gun.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Gospers\\_glider\\_gun.gif](http://en.wikipedia.org/w/index.php?title=File:Gospers_glider_gun.gif) *License:* GNU Free Documentation License *Contributors:* Kieff

**File:Conways game of life breeder.png** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Conways\\_game\\_of\\_life\\_breeder.png](http://en.wikipedia.org/w/index.php?title=File:Conways_game_of_life_breeder.png) *License:* Creative Commons Attribution-Sharealike 3.0 *Contributors:* Hyperdeath

**File:Game of life block with border.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_block\\_with\\_border.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_block_with_border.svg) *License:* Public Domain *Contributors:* None

**File:Game of life beehive.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_beehive.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_beehive.svg) *License:* Public Domain *Contributors:* None

**File:Game of life loaf.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_loaf.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_loaf.svg) *License:* Public Domain *Contributors:* None

**File:Game of life boat.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_boat.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_boat.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life blinker.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_blinker.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_blinker.gif) *License:* Public domain *Contributors:* JokeySmurf at en.wikipedia

**File:game of life toad.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_toad.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_toad.gif) *License:* Public Domain *Contributors:* JokeySmurf

**File:game of life beacon.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_beacon.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_beacon.gif) *License:* Public domain *Contributors:* JokeySmurf at en.wikipedia

**File:game of life pulsar.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_pulsar.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_pulsar.gif) *License:* Public domain *Contributors:* JokeySmurf at en.wikipedia

**File:Game of life animated glider.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_animated\\_glider.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_animated_glider.gif) *License:* Public domain *Contributors:* Rodrigo Silveira Camargo (RodrigoCamargo on commons or RodrigoCamargo on en.wikipedia

**File:Game of life animated LWSS.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_animated\\_LWSS.gif](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_animated_LWSS.gif) *License:* Public domain *Contributors:* Rodrigo Silveira Camargo a.k.a. RodrigoCamargo at en.wikipedia

**File:Game of life fpento.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_fpento.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_fpento.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life diehard.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_diehard.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_diehard.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life acorn.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_acorn.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_acorn.svg) *License:* Public Domain *Contributors:* None

**File:Game of life glider gun.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_glider\\_gun.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_glider_gun.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life infinite1.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_infinite1.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_infinite1.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life infinite2.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_infinite2.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_infinite2.svg) *License:* Public Domain *Contributors:* Bryan.burgers

**File:game of life infinite3.svg** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Game\\_of\\_life\\_infinite3.svg](http://en.wikipedia.org/w/index.php?title=File:Game_of_life_infinite3.svg) *License:* Public Domain *Contributors:* Original version by w:en>User:Kieff. SVG version by User:FedericoMP.

**File:Long gun.gif** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Long\\_gun.gif](http://en.wikipedia.org/w/index.php?title=File:Long_gun.gif) *License:* Public Domain *Contributors:* Simpsons contributor (talk)

**File:Oscillator.gif** *Source:* <http://en.wikipedia.org/w/index.php?title=File:Oscillator.gif> *License:* GNU Free Documentation License *Contributors:* Original uploader was Grontesca at en.wikipedia

**File:Turing Machine in Golly.png** *Source:* [http://en.wikipedia.org/w/index.php?title=File:Turing\\_Machine\\_in\\_Golly.png](http://en.wikipedia.org/w/index.php?title=File:Turing_Machine_in_Golly.png) *License:* GNU General Public License *Contributors:* Andrew Trevorrow and Tomas Rokicki

# License

Creative Commons Attribution-Share Alike 3.0 Unported  
//creativecommons.org/licenses/by-sa/3.0/