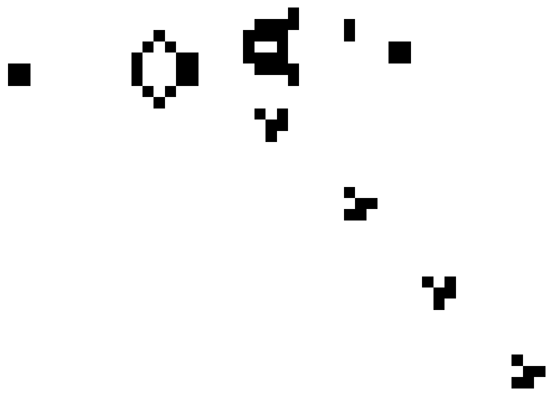
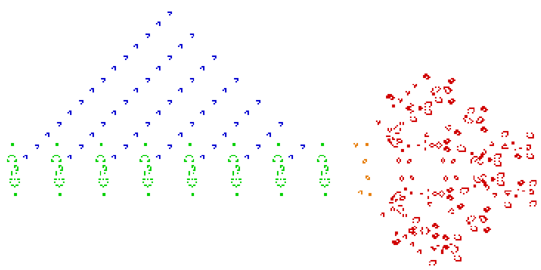
Conway's Game of Life

"Game of Life" redirects here. For other uses, see Game of Life (disambiguation).
"Conway game" redirects here. For Conway's surreal number game theory, see surreal number.

The **Game of Life**, also known simply as **Life**, is a



A single Gosper's Glider Gun creating "gliders"



A screenshot of a puffer-type breeder (red) that leaves glider guns (green) in its wake, which in turn create gliders (blue). (animation)

cellular automaton devised by the British mathematician John Horton Conway in 1970.[1]

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, or, for advanced "players", by creating patterns with particular properties.

1 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*, or "populated" or "unpopulated" (the difference may seem minor, except when viewing it as an early model of human/urban behavior simulation or how one views a blank space on a grid). 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 over-population.
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.

2 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.[2][3] 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. Scholars in various fields, such as computer science, physics, biology, biochemistry, economics, mathematics, philosophy, and generative sciences have made use of the way that complex patterns can emerge from the implementation of the game's 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, which might govern our universe.[4][5][6]

The popularity of Conway's Game of Life was helped by its coming into being just in time for a new generation of inexpensive computer access 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.[7][8]

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

1. There should be no explosive growth.
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.[9]

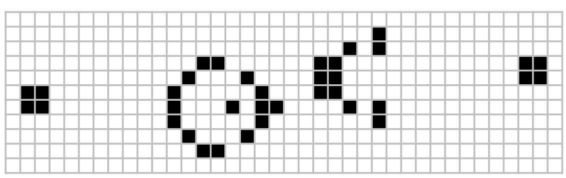
3 Examples of patterns

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 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[10] (these were the first gliders ever discovered).[11]

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[12] [13] examples of these three classes are shown below, with live cells shown in black, and dead cells shown in white.

The "pulsar"[14] 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 many periods are known to exist,[15] and oscillators of periods 4, 8, 14, 15, 30 and a few others have been seen to arise from random initial conditions.[16] Patterns called "Methuselahs" can evolve for long periods before stabilizing, the first-discovered of which was the R-pentomino. "Diehard" is a pattern that eventually disappears (rather than merely stabilizing) after 130 generations, which is conjectured to be maximal for patterns with seven or fewer cells.[17] "Acorn" takes 5206 generations to generate 633 cells including 13 escaped gliders.[18]

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. 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" produces its first glider on the 15th generation, and another glider every 30th generation from then on. For many years this glider gun was the smallest one known.[19] In 2015 a period-120 gun was discovered that has fewer live cells but a larger bounding box.[20]



Gosper glider gun

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"[21] each, while the third creates two. The first has only 10 live cells (which has been proven to be minimal).[22] The second fits in a 5×5 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.[23] 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.[2][3]

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.[3]

4 Undecidability

Many patterns in the game of life eventually become a combination of still lives, oscillators and spaceships; other patterns may be called chaotic. A pattern may stay chaotic for a very long time until it eventually settles to such a combination.

It can be asked whether the game of life is decidable: whether an algorithm exists, so that given an "initial" pattern and a "later" pattern, the algorithm can tell whether, starting with the initial pattern, the later pattern is ever going to appear. This turns out to be impossible: no such algorithm exists. This is in fact a corollary of the halting problem.[24]

Indeed, since the game of life includes a pattern that is equivalent to a UTM (universal Turing machine), this "deciding" algorithm, if it existed, could have been used to solve the halting problem, by taking the initial pattern as the one corresponding to a UTM+input and the later pattern as the one corresponding to a halting state of the machine with an empty tape (as one can modify the Turing machine to always erase the tape before halting). However the halting problem is provably undecidable and so such an algorithm does not exist.

It also follows that some patterns exist that remain chaotic forever: otherwise one could just progress the game of life sequentially until a non-chaotic pattern emerges, and then easily compute whether the later pattern is going to appear.

5 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.[25][26] 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 also a spaceship, and is in fact the first spaceship constructed in the Game of Life which is neither orthogonal nor purely diagonal (these are called knightships).[27][28]

On November 23, 2013, Dave Greene built the first replicator in Conway's Game of Life that creates a complete copy of itself, including the instruction tape.[29]

In December 2015, diagonal versions of the Gemini were built.[30]

6 Iteration

From any 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;[31][32] many also produce one or more gliders or spaceships that travel indefinitely away from the initial location. Because of the nearest-neighbor based rules, no "information" can travel through the grid at a greater rate than one cell per unit time, so this velocity is said to be the cellular automaton speed of light and denoted c .

7 Algorithms

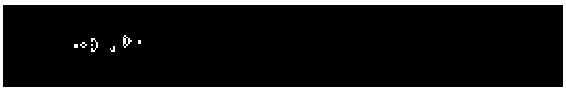
Early patterns with unknown futures, such as the R-pentomino, led computer programmers across the world to write programs to track the evolution of Life patterns.

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.[33]

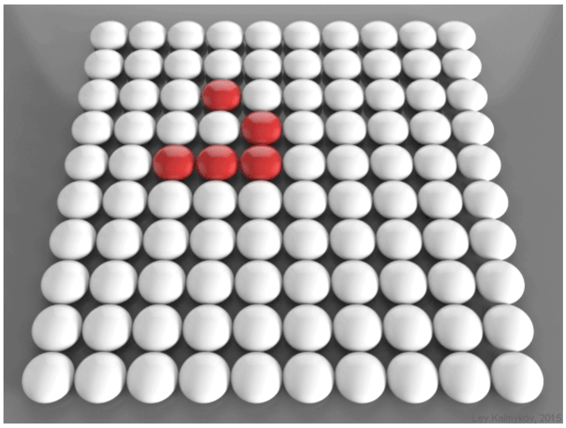
To avoid decisions and branches in the counting loop, the rules can be rearranged from an egocentric approach of the inner field regarding its neighbors to a scientific observer's viewpoint: if the sum of all nine fields is 3, the inner field state for the next generation will be life (no matter of its previous contents); if the all-field sum is 4, the inner field retains its current state and every other sum sets the inner field to death.

If it is desired to save memory, the storage can be reduced to one array plus 3 line buffers. One line buffer is used to calculate the successor state for a line, then the second line buffer is used to calculate the successor state for the next line. The first buffer is then written to its line and freed to hold the successor state for the third line. If a toroidal array is used, a third buffer is needed so that the original state of the first line in the array can be saved until the last line is computed.



Glider gun within a toroidal array. The stream of gliders eventually wraps round and destroys the gun.

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.



Red glider on the square lattice with periodic boundary conditions.

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 hash-table lookup or search operation, slowing down simulation speed. With more sophisticated data structures this problem can also be largely solved.

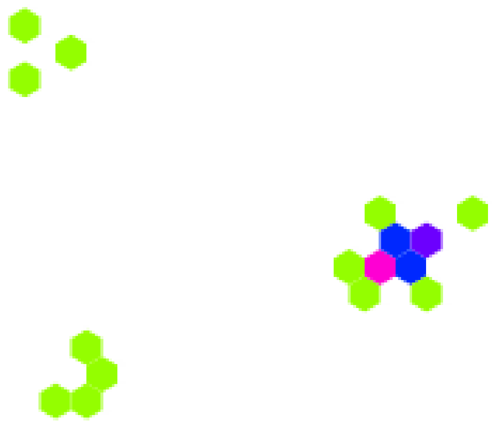
For exploring large patterns at great time-depths, sophisticated algorithms such as Hashlife may be useful. There is also a method, applicable to other cellular automata too, for implementation of the Game of Life using arbitrary asynchronous updates whilst still exactly emulating the behaviour of the synchronous game.[34]

Source code examples that implements the basic Game of Life scenario in various programming languages, including C, C++, Java and Python can be found here[35]

8 Variations on Life

Main article: Life-like cellular automaton

Since Life's inception, new similar cellular automata have been developed. The standard Game of Life is symbolised as "B3/S23": A cell is "**B**orn" if it has exactly 3 neighbours, "**S**urvives" if it has 2 or 3 living neighbours; it dies otherwise. 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.[36][37] 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.



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)

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)[38] and 3-D square variations have been developed, as have 2-D hexagonal and 2-D triangular variations. A variant using non-periodic tile grids has also been made.[39]

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 Cellebration's multi-coloured "Rules Table" and "Weighted Life" rule families each include sample rules equivalent to Conway's Life.

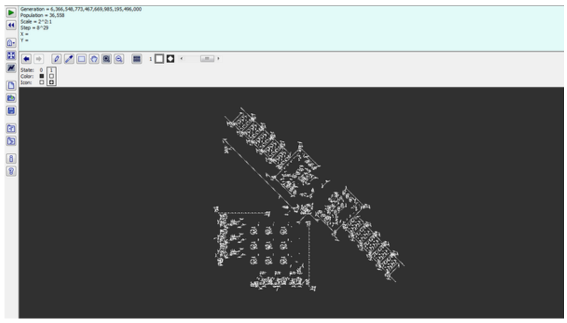
Patterns relating to fractals and fractal systems may also be observed in certain Life-like variations. For example, the automaton B1/S12 generates four very close approximations to the Sierpiński triangle when applied to a single live cell. The Sierpiński triangle can also be observed in Conway's Game of Life by examining the long-term growth of a long single-cell-thick line of live cells,[40] as well as in Highlife, Seeds (B2/S), and Wolfram's Rule 90.[41]

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.[42] 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.[43] Except for the variation among ON cells, both of these variations act identically to Life.

9 Music

Various musical composition techniques use Conway's Life, especially in MIDI sequencing.[44] A variety of programs exist for creating sound from patterns generated in Life (see footnotes for links to examples).[45][46][47]

10 Notable Life programs



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

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 interactive Life program was written in ALGOL 68 for the PDP-7 by M. J. T. Guy and S. R. Bourne. The results were published in the October 1970 issue of *Scientific American*[48] and—regarding the use of the program—reports "Without its help, some discoveries about the game would have been difficult to make."

There are now thousands of Life programs online, so a full list will not be provided here. The following is a small selection 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.

- Golly. A cross-platform (Windows, Macintosh, Linux and also iOS and Android) 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.
- Mirek's Cellebration. 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. A cellular-automaton laboratory by Jon Bennett. The standard UNIX X11 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.

Google implemented an easter egg of Conway's Game of Life in 2012. Users who search for the term are shown an implementation of the game in the search results page.

11 See also

- Artificial life
- Glider
- *Glory Season*, a novel by David Brin, is set in a future society where the Game of Life is played in a competitive 2-player mode
- Langton's ant, another rule-set that uses a rectangular grid and shows emergent patterns
- Poietic Generator, a "human" game of life.

12 References

[1] Gardner, Martin (October 1970). *Mathematical Games – The fantastic combinations of John Conway's new solitaire game "life"*. *Scientific American*. **223**. pp. 120–123. ISBN 0-89454-001-7. Archived from the original on 2009-06-03. Retrieved 2011-06-26.

[2] It is a model and simulation that is interesting to watch and can show that simple things can become complicated problems. Paul Chapman (11 November 2002). "Life Universal Computer". Retrieved 12 July 2009.

[3] Berlekamp, E. R.; Conway, John Horton; Guy, R. K. (2001–2004). *Winning Ways for your Mathematical Plays* (2nd ed.). A K Peters Ltd.

[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". Retrieved July 12, 2009.

[8] Adam P. Goucher. "Spartan universal computer-constructor". LifeWiki. Retrieved July 12, 2009.

[9] Conway, private communication to the 'Life list', 14 April 1999.

[10] "R-pentomino". LifeWiki. Retrieved July 12, 2009.

[11] Stephen A. Silver. "Glider". The Life Lexicon. Retrieved July 12, 2009.

[12] "Census Results in Conway's Game of Life". The Online Life-Like CA Soup Search. Archived from the original on 2009-09-10. Retrieved July 12, 2009.

[13] "Spontaneous appeared Spaceships out of Random Dust". Achim Flammenkamp (1995-12-09). Retrieved July 10, 2012.

[14] "Pulsar". Eric Weisstein's Treasure Trove of Life. Retrieved 2008-09-16.

[15] Game of Life Status page, Jason Summers, retrieved 2012-02-23.

[16] Achim Flammenkamp (2004-09-07). "Most seen natural occurring ash objects in Game of Life". Retrieved 2008-09-16.

[17] Stephen A. Silver. "Diehard". The Life Lexicon. Retrieved July 12, 2009.

[18] Koenig, H. (February 21, 2005). "New Methuselah Records". Retrieved January 24, 2009.

[19] Stephen A. Silver. "Gosper glider gun". The Life Lexicon. Retrieved July 12, 2009.

[20] The Hunting of the New Herschel Conduits, ConwayLife forums, April 28th, 2015, posts by Michael Simkin ("simsim314") and Dongook Lee ("Scorbie").

[21] "Block-laying switch engine". LifeWiki. Retrieved July 12, 2009.

[22] "Infinite Growth". Eric Weisstein's Treasure Trove of Life. Retrieved 2008-09-16.

[23] Stephen A. Silver. "Rake". The Life Lexicon. Retrieved July 12, 2009.

[24] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy, Winning Ways for your Mathematical Plays. Academic Press, 1982

[25] "Universal Constructor Based Spaceship". Conwaylife.com. Retrieved 2012-06-24.

[26] "Gemini – LifeWiki". Conwaylife.com. Retrieved 2012-06-24.

[27] Aron, Jacob (16 June 2010). "First replicating creature spawned in life simulator". *New Scientist*. Retrieved 12 October 2013.

[28] "Gemini – LifeWiki". Conwaylife.com. Retrieved 2013-10-16.

[29] "Geminoid Challenge". Conwaylife.com. Retrieved 2015-06-25.

[30] "Demonoid". LifeWiki. Retrieved 18 June 2016.

[31] Andrzej Okrasinski. "Game of Life Object Statistics". Archived from the original on 2009-07-27. Retrieved July 12, 2009.

[32] Nathaniel Johnston. "The Online Life-Like CA Soup Search". Archived from the original on 2009-09-10. Retrieved July 12, 2009.

[33] Alan Hensel. "About my Conway's Game of Life Applet". Retrieved July 12, 2009.

[34] Nehaniv, Chrystopher L. (15–18 July 2002). *Self-Reproduction in Asynchronous Cellular Automata*. 2002 NASA/DoD Conference on Evolvable Hardware. Alexandria, Virginia, USA: IEEE Computer Society Press. pp. 201–209. doi:10.1109/EH.2002.1029886. ISBN 0-7695-1718-8. Archived from the original on April 3, 2015. Retrieved 17 March 2015.

[35] "Conway's Game of Life".

[36] HighLife – An Interesting Variant of Life by David Bell (.zip file)

[37] Stephen A. Silver. "Replicator". The Life Lexicon. Retrieved July 12, 2009.

[38] "Elementary Cellular Automaton". Wolfram Mathworld. Retrieved July 12, 2009.

[39] "First gliders navigate ever-changing Penrose universe". New Scientist.

[40] "One cell thick pattern". LifeWiki. Retrieved July 12, 2009.

[41] "Life Imitates Sierpinski". ConwayLife.com forums. Retrieved July 12, 2009.

[42] "Immigration". Eric Weisstein's Treasure Trove of Life. Retrieved 2008-09-16.

[43] "QuadLife". Eric Weisstein's Treasure Trove of Life. Retrieved 2008-09-16.

[44] Burraston, Dave; Edmonds, Ernest; Livingstone, Dan; Miranda, Eduardo Reck (2004). "Cellular Automata in MIDI based Computer Music". *Proceedings of the 2004 International Computer Music Conference*. CiteSeerX: 10.1.1.6.3882 hdl:10453/1425.

[45] "glitchDS – Cellular Automaton Sequencer For The Nintendo DS". Synthtopia.com. 2008-05-29. Retrieved 2012-06-24.

[46] "Game Of Life Music Sequencer". Synthtopia.com. 2009-04-29. Retrieved 2012-06-24.

[47] "Game Of Life Music Sequencer For iOS, Runxt Life". Synthtopia.com. 2011-01-12. Retrieved 2012-06-24.

[48] Gardner, Martin (October 1970). "Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"". *Scientific American*. **223**: 120–123.

13 External links

- Conway's Game of Life at DMOZ
- Life Lexicon, Extensive Life Lexicon (with many patterns)
- Game of Life News
- LifeWiki
- Conway Life forums
- Catagolue, an online database of objects in Conway's Game of Life and similar cellular automata
- Cellular Automata FAQ – Conway's Game of Life
- For platform .net, mono – Game of Life

14 Text and image sources, contributors, and licenses

14.1 Text

- **Conway's Game of Life** *Source:* https://en.wikipedia.org/wiki/Conway'{}s_Game_of_Life?oldid=730406576 *Contributors:* Damian Yerrick, Carey Evans, Chuck Smith, LC~enwiki, Zundark, The Anome, Tarquin, Awaterl, Maury Markowitz, Frecklefoot, Nealmcb, Oliver Pereira, Tenbaset, DIG~enwiki, Shellreef, SGBailey, Ixfd64, Graue, Dcljr, Dori, Paul A, Eric119, DavidWBrooks, Muriel Gottrop~enwiki, Snoyes, Notheruser, Grontesca, Sir Paul, AugPi, Evercat, Schneelocke, Ffransoo, Charles Matthews, Dysprosia, Doradus, Markhurd, Tkorrovi, Val42, Sabbut, Bevo, Shizhao, Dbabbitt, AnonMoos, Pakaran, Banno, AnthonyQBachler, Rossumcapek, PuzzletChung, Phil Boswell, Sjorford, Robbot, Owain, Fredrik, Daelin, Chris 73, P0lyglut, Sverdrup, Henrygb, Lesonyrra, Pcr, Bkell, Paul G, Sheridan, UtherSRG, Stay cool~enwiki, Jleedev, Jpo, Giftlite, SamB, DavidCary, Kim Bruning, Tom harrison, Ds13, Perl, Home Row Keysplurge, Raekwon, Solipsist, Python eggs, Aesir~enwiki, Ogxela, Gdr, LucasVB, Noe, MisfitToys, Kaldari, DNewhall, BookgirlST, DragonflySixtyseven, Michael L. Kaufman, Robin klein, D6, JTN, Discospinster, FiP, Rspeer, MeltBanana, Abelson, Paul August, ESkog, CanisRufus, Livajo, Shanes, Aplusbi, Meggar, Kappa, La goutte de pluie, ACW, Ferkel, Wayfarer, Dvgrn, Steven Watson, Borisblue, Shadowcheets, Ashley Pomeroy, Ynhockey, Dirac1933, RogerBarnett, DV8 2XL, Bookandcoffee, Markaci, Arbol01, Weyes, LOL, Ae-a, KymFarnik, Xhin, GregorB, Gerbrant, Marudubshinki, Graham87, BD2412, TobyJ, Melesse, Dimitrii, Panoptical, Gevil, Agorf, Quiddity, Salix alba, Bubba73, Rangek, Mathbot, JYOuyang, Mitsukai, Quuxplusone, Julescubtree, Kosmar~enwiki, Srleffler, Butros, NevilleDNZ, Chobot, DaGizza, Turidoth, Scuzzi, YurikBot, Wavelength, Quentin X, Hydrargyrum, Gaius Cornelius, CambridgeBayWeather, Ihope127, Anomalocaris, Scote, Trovatore, Sciguy47, Retired username, Voidxor, BMAH07, Mkill, Palpalpalpal, Sir Isaac, Wknight94, Gcsnelgar, Closedmouth, Zdim~enwiki, Gulliveig, LeonardoRob0t, SDS, Trubye, Wiggy04, GrinBot~enwiki, Zvika, Plook~enwiki, SmackBot, RDBury, John Lunney, Tomyumgoong, Meshach, Faisal.akeel, McGeddon, Dxco, Jrockley, Delldot, Betacommand, Amux, Chris the speller, Bluebot, Kurykh, Thumperward, Darth Panda, Emurphy42, Hgrosser, Froese, Simpsons contributor, Nick Levine, John.constantine, Ioscius, OrphanBot, Onorem, Racklever, Lesnail, Andy120290, Richard001, ABoerma, Dvorak729, DMacks, Govvy, Sam Tobar, Davipo, Cabbers, Christopher.Madsen, Sandman303, Jim.belk, Tedweird, E-Kartoffel, Wody1025, Atakdoug, Dan Gluck, E-trail, Kenirwin, RekishiEJ, Amakuru, Arto B, Ken-Walker, Dclayh, Tawkerbot2, Fprefect111, ZICO, IntrigueBlue, Tim1988, Hyperdeath, Dr unix, DumbBOT, Thijs!bot, RodrigoCamargo, Barticus88, Mojo Hand, Headbomb, Geneffects, Gioto, Opelio, Adijk, Dylan Lake, Pichote, Omeganian, Joee92, Savant13, TheAllSeeingEye, Hillgentleman, Nikevich, Baccyak4H, Quanyails, Torchiest, David Eppstein, Spellmaster, Heliac, Pavel Jelínek, Gwern, Dr. Morbius, MartinBot, Ariel., Bloodrage, LedgendGamer, Pharaoh of the Wizards, Ifomichev~enwiki, RJBotting, Maurice Carbonaro, Darth Mike, Dispenser, Bot-Schafter, Katalaveno, Darthrevan1789, NewEnglandYankee, Moondoll, Paulmmn, Diego, Jarl Friis, VolkovBot, JohnBlackburne, Lexein, Ipso2, Clarince63, SQL, Gorank4, Falcon8765, EverGreg, AlphaPyro, Gerakibot, Vvevo, Flyer22 Reborn, NuclearCarnage, HairyWombat, ImageRemovalBot, Rat at WikiFur, Martarius, ClueBot, DFRussia, Justin W Smith, Jwz, Marktompsett, Excirial, Caekaert, BobKawanaka, NuclearWarfare, Gleishma, El bot de la dieta, Versus22, Rbtmdl, Johnuniq, Obrienmi8, Bitbutter, InternetMeme, Jeanclaude perez, XLinkBot, Shoemaker's Holiday, HexaChord, Addbot, Tcncv, M.nelson, Esger, PauloCalipari, Deadlyfish, LinkFA-Bot, Jtradke, ScAvenger, Jarble, B1u SkR33N, Legobot, Yobot, The Earwig, Nallimbot, Bryan.burgers, Narumara, Groaznic, AnomieBOT, Ciphers, Flewis, Citation bot, Twri, ArthurBot, Capricorn42, Etoombs, Reonic, Jmundo, Ecomba, Andykt, Captain-n00dle, FrescoBot, Satbir15, JokeySmurf, Wikiachmed, Citation bot 1, Alessio88, Theusernameiwantedisalreadyinuse, DrilBot, Achim1999, MisterSanderson, MondalorBot, Serols, TobeBot, Trappist the monk, Throwaway85, Standardfact, Callanecc, Updatehelper, RjwilmsiBot, Calcyman, EmausBot, John of Reading, WikitanvirBot, Immunize, Mk5384, Dewritech, Mz7, Ὁ οἶστρος, H3llBot, Bud Charles, Johnf.at.home, Randomphantom45, Dmitry123456, Usb10, David van bruwaene, Baltar, Gaius, ClueBot NG, Jbragadeesh, Lanthanum-138, Frietjes, BooHamster, Redslazer, Helpful Pixie Bot, Zibart, Jlopez1967, Candleabracadabra, UBERNESS, BG19bot, Leonxlin, Andrew.brusentsov, Pieboy32, Hughsk, Steeldragon7, Merritttttt, The1337gamer, BattyBot, Millennium bug, Cyberbot II, TAlex, Luca Edd Fike, Kperry1408, Dexbot, ToBeFree, Toudou, Lugia2453, Aris.olt, ElliottBelardo, Ziddarth, Demdepp, Alexwho314, Genaro.juarez.martinez, Eyesnore, Felixgers, Peter Meyer (TWZ author), Drshch08, JeremyLThompson, Paul2520, Lev Kalmykov, St170e, Monkbot, Narky Blert, 12Me21, Julietdeltalima, C.vessiaris, Timburget, Mischizzle, Sequoia 42, Captain Chesapeake, DERPALERT, Furlongs P. Fortnight, Mpritham, TastedWasted, Nina2589, Stavelef, Lokananda Hari, Anareth, Dasschloss, Skyebug, Oligalma and Anonymous: 408

14.2 Images

- **File:Commons-logo.svg** *Source:* https://upload.wikimedia.org/wikipedia/en/4/4a/Commons-logo.svg *License:* CC-BY-SA-3.0 *Contributors:* ? *Original artist:* ?
- **File:Conways_game_of_life_breeder.png** *Source:* https://upload.wikimedia.org/wikipedia/commons/e/ec/Conways_game_of_life_breeder.png *License:* CC BY-SA 3.0 *Contributors:* Own work *Original artist:* Hyperdeath
- **File:Game_of_life_acorn.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/b/b9/Game_of_life_acorn.svg *License:* Public domain *Contributors:* Own work *Original artist:* None
- **File:Game_of_life_animated_LWSS.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/3/37/Game_of_life_animated_LWSS.gif *License:* Public domain *Contributors:* Own work (Original caption: "*This animation was made by myself, and I release it into the public domain.*")
  Original artist: Rodrigo Silveira Camargo a.k.a. RodrigoCamargo at en.wikipedia
- **File:Game_of_life_animated_glider.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/f/f2/Game_of_life_animated_glider.gif *License:* Public domain *Contributors:* Own work (Original caption: "*This animation was made by myself.*") *Original artist:* Rodrigo Silveira Camargo (RodrigoCamargo on commons or RodrigoCamargo on en.wikipedia
- **File:Game_of_life_beacon.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/1/1c/Game_of_life_beacon.gif *License:* Public domain *Contributors:* Own work (Original caption: "*Created myself using the tool at www.conwaylife.com*") *Original artist:* JokeySmurf at en.wikipedia
- **File:Game_of_life_beehive.svg** *Source:* https://upload.wikimedia.org/wikipedia/commons/6/67/Game_of_life_beehive.svg *License:* Public domain *Contributors:* Own work *Original artist:* None
- **File:Game_of_life_blinker.gif** *Source:* https://upload.wikimedia.org/wikipedia/commons/9/95/Game_of_life_blinker.gif *License:* Public domain *Contributors:* Created myself using the tool at www.conwaylife.com *Original artist:* JokeySmurf at en.wikipedia

14.3 Content license