

Conway's Game of Life

The **Game of Life**, also known as **Conway's Game of Life** or simply **Life**, is a cellular automaton devised by the British mathematician John Horton Conway in 1970.^[1] It is a zero-player game,^{[2][3]} 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 in discrete time steps called generations. There is no limit to the number of generations (computational resources permitting) or win condition.

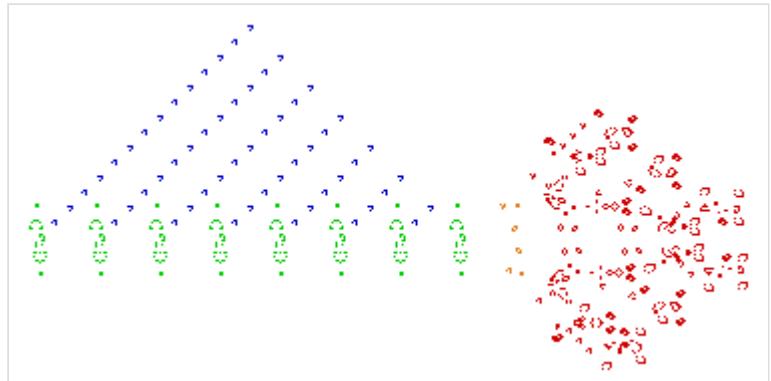


A single [Gosper's glider gun](#) creating [gliders](#)

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, *live* or *dead* (or *populated* and *unpopulated*, respectively). 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:

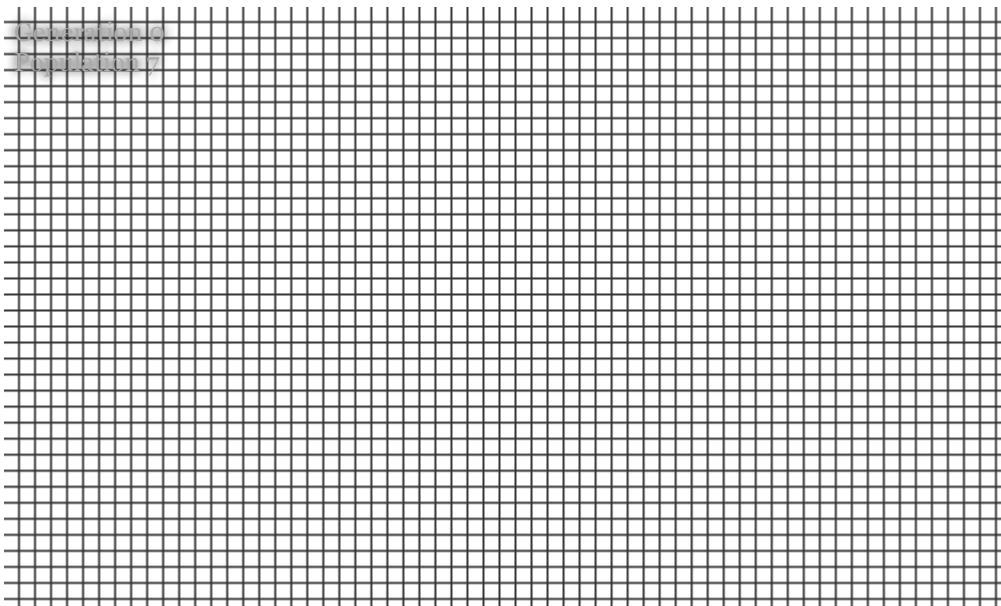
1. Any live cell with fewer than two live neighbours dies, as if by underpopulation.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbours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



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

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, live or dead; births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick*.^[nb 1] Each generation is a *pure function* of the preceding one. The rules continue to be applied repeatedly to create further generations.

In the editable widget below, one of the patterns discovered by Conway, the I-heptomino can be simulated.



🔍 🔍 # ⏪ ⏴ ⏵

History

Cellular automata have their origins in the work of Stanisław Ulam and John von Neumann in the 1940s. Development continued through the 1950's and 1960's when, in 1968, John Horton Conway began doing experiments with a variety of different two-dimensional cellular automaton rules. Conway's initial goal was to define an interesting and unpredictable cellular automaton.^[3] According to Martin Gardner, Conway experimented with different rules, aiming for a ruleset that would allow for patterns to "apparently" grow without limit, while keeping it difficult to *prove* that any given pattern would do so. Moreover, some "simple initial patterns" should "grow and change for a considerable period of time" before settling into a static configuration or a repeating loop.^[1] Conway later wrote that the basic motivation for Life was to create a "universal" cellular automaton.^[7]

The game made its first public appearance in the October 1970 issue of *Scientific American*, in Martin Gardner's "Mathematical Games" column, which was based on personal conversations with Conway. Theoretically, the Game of Life has the power of a universal Turing machine: anything that can be computed algorithmically can be computed within the Game of Life.^{[8][2]} Gardner wrote, "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)."^[1]

The popularity of the Game of Life was helped by its coming into being at the same time as increasingly inexpensive computer access. 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, the Game of Life was simply a programming challenge: a fun way to use otherwise wasted CPU cycles. 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 Game of Life board.^{[9][10]} Users of online communities such as LifeWiki continue to develop new patterns.

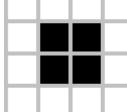
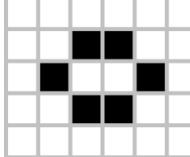
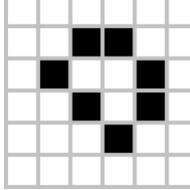
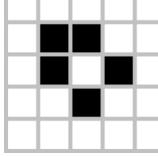
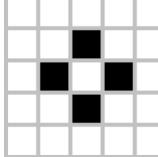
Analysis

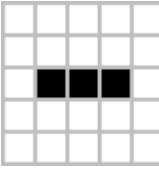
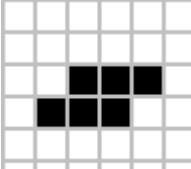
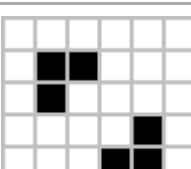
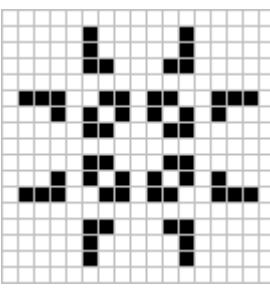
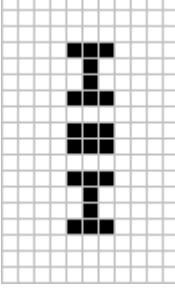
Pattern taxonomy

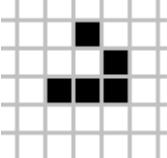
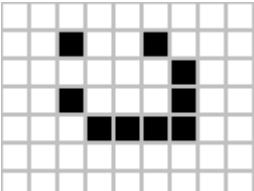
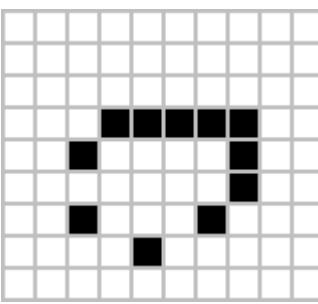
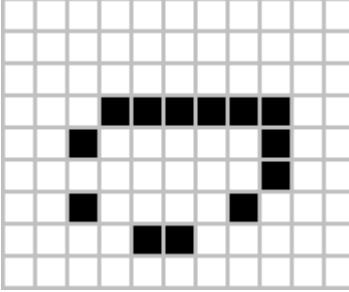
Many different types of patterns occur in the Game of Life, which are classified according to their behaviour. Common pattern types include: *still lifes*, which do not change from one generation to the next; *oscillators*, which return to their initial state after a finite number of generations; and *spaceships*, which translate themselves across the grid.

The earliest interesting patterns in the Game of Life were discovered without the use of computers. The simplest still lifes and oscillators were discovered while tracking the fates of various small starting configurations using [graph paper](#), [blackboards](#), and physical game boards, such as those used in [Go](#). 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 generated six escaping [gliders](#);^[11] these were the first spaceships ever discovered.^[12]

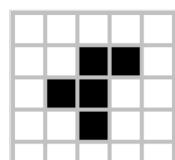
Frequently occurring^{[13][14]} examples (in that they emerge frequently from a random starting configuration of cells) of the three aforementioned pattern types are shown below, with live cells shown in black and dead cells in white. *Period* refers to the number of ticks a pattern must iterate through before returning to its initial configuration.

Still lifes	
Block	
Bee-hive	
Loaf	
Boat	
Tub	

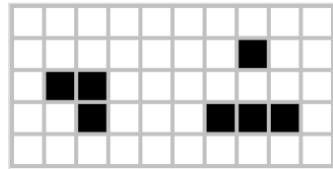
Oscillators	
Blinker (period 2)	 A 3x3 grid showing a horizontal line of three black cells in the second row.
Toad (period 2)	 A 3x4 grid showing a horizontal line of four black cells in the second row, with a single black cell below it in the fourth column.
Beacon (period 2)	 A 3x3 grid showing a vertical line of three black cells in the second column, with a single black cell below it in the third row.
Pulsar (period 3)	 A complex, symmetric pattern consisting of multiple vertical columns of black cells, forming a repeating structure every three generations.
Penta-decathlon (period 15)	 A vertical column of five black cells, spaced evenly along a vertical axis.

Spaceships	
Glider	
Light-weight spaceship (LWSS)	
Middle-weight spaceship (MWSS)	
Heavy-weight spaceship (HWSS)	

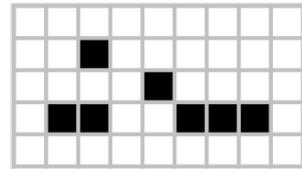
The *pulsar*^[15] is the most common period-3 oscillator. The great majority of naturally occurring oscillators have a period of 2, like the blinker and the toad, but oscillators of all periods are known to exist,^{[16][17][18]} and oscillators of periods 4, 8, 14, 15, 30, and a few others have been seen to arise from random initial conditions.^[19] Patterns which evolve for long periods before stabilizing are called *Methuselahs*, the first-discovered of which was the R-pentomino. *Diehard* is a pattern that disappears after 130 generations. Starting patterns of eight or more cells can be made to die after an arbitrarily long time.^[20] *Acorn* takes 5,206 generations to generate 633 cells, including 13 escaped gliders.^[21]



The R-pentomino



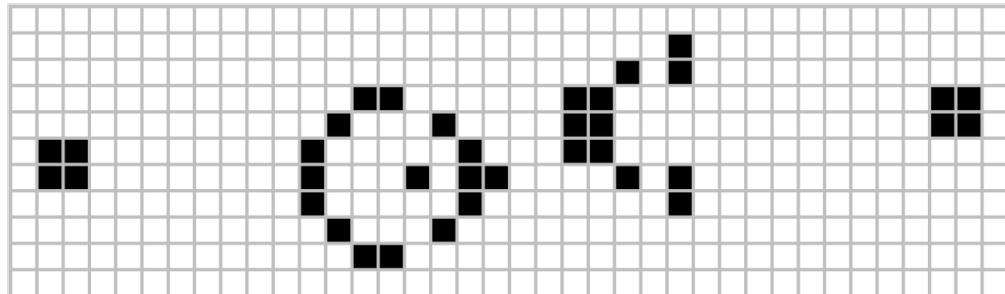
Diehard



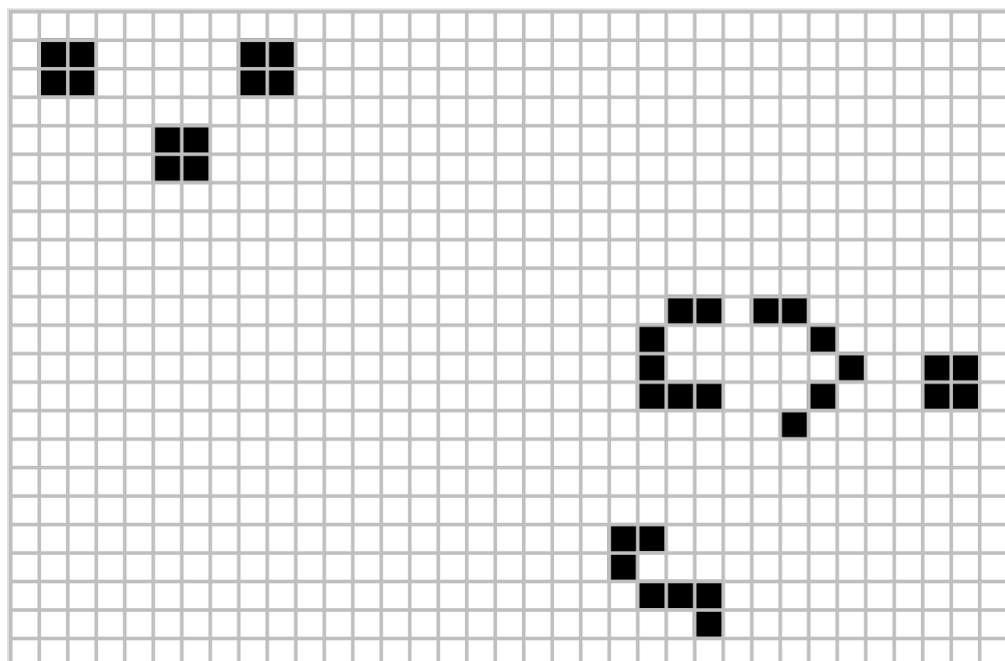
Acorn

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 prize of

fifty dollars (equivalent to \$400 in 2024) to the first person who could prove or disprove the conjecture before the end of 1970. The prize was won in November 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.^[22] In 2015, a gun called the "Simkin glider gun", which releases a glider every 120th generation, was discovered that has fewer live cells but which is spread out across a larger bounding box at its extremities.^[23]

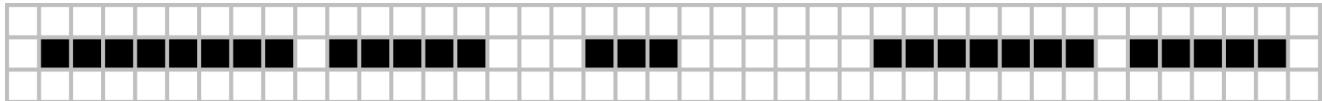
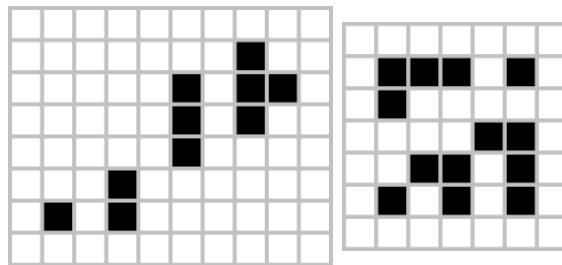


Gosper glider gun



Simkin glider gun

Smaller patterns were later found that also exhibit infinite growth. All three of the patterns shown below grow indefinitely. The first two create a single *block-laying switch engine*: a configuration that leaves behind two-by-two still life blocks as it translates itself across the game's universe.^[24] The third configuration creates two such patterns. The first has only ten live cells, which has been proven to be minimal.^[25] The second fits in a five-by-five square, and the third is only one cell high.



Later discoveries included other *guns*, which are stationary, and which produce gliders or other spaceships; *puffer trains*, which move along leaving behind a trail of debris; and *rakes*, which move and emit spaceships.^[26] 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 a specific position, 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.^{[8][2]} In fact, several different programmable computer architectures^{[27][28]} have been implemented in the Game of Life, including a pattern that simulates Tetris.^[29]

Oblique spaceships

Until the 2010s, all known spaceships could only move orthogonally or diagonally. Spaceships which move neither orthogonally nor diagonally are commonly referred to as *oblique spaceships*.^{[30][31]} On May 18, 2010, Andrew J. Wade announced the first oblique spaceship, dubbed "Gemini", that creates a copy of itself on (5,1) further while destroying its parent.^{[32][31]} This pattern replicates in 34 million generations, and uses an instruction tape made of gliders oscillating between two stable configurations made of Chapman–Greene construction arms. These, in turn, create new copies of the pattern, and destroy the previous copy. In December 2015, diagonal versions of the Gemini were built.^[33]

A more specific case is a *knightship*, a spaceship that moves two squares left for every one square it moves down (like a knight in chess), whose existence had been predicted by Elwyn Berlekamp since 1982. The first elementary knightship, Sir Robin, was discovered in 2018 by Adam P. Goucher.^[34] This is the first new spaceship movement pattern for an elementary spaceship found in forty-eight years. "Elementary" means that it cannot be decomposed into smaller interacting patterns such as gliders and still lifes.^[35]

Self-replication

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.^[2] On November 23, 2013, Dave Greene built the first replicator in the Game of Life that creates a complete copy of itself, including the instruction tape.^[36] In October 2018, Adam P. Goucher finished his construction of the oEoP metacell, a metacell capable of self-replication. This differed from previous metacells, such as the OTCA metapixel by Brice Due, which only worked with already constructed copies near them. The oEoP metacell works by using construction arms to create copies that simulate the programmed rule.^[37] The actual simulation of the Game of Life or other Moore neighbourhood rules is done by simulating an equivalent rule using the von Neumann neighbourhood with more states.^[38] The name oEoP is short for "Zero Encoded by Zero Population", which indicates that instead of a metacell being in an "off" state simulating empty space, the oEoP metacell removes itself when the cell enters that state, leaving a blank space.^[39]

Computability theory

Many starting patterns in the Game of Life eventually become a combination of still lifes, 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.

Patterns may be constructed to emulate logical gates and replicate computing components such as information storage, logical instruction sets and program storage. These components then can be used to construct a computer within the Game of Life. Therefore, the Game of Life is Turing-complete and may therefore execute arbitrary programs. By the halting problem it is undecidable whether an arbitrary program executed in the Game of Life would ever terminate. Translating back into the terms of the Game of Life, given an initial configuration pattern and a target pattern, it is also undecidable whether the target pattern will ever appear.^[2]

Other disciplines

Since its publication, the Game of Life has attracted much interest because of the surprising ways in which the patterns can evolve. It provides an example of emergence and self-organization.^[3] A version of Life that incorporates random fluctuations has been used in physics to study phase transitions and nonequilibrium dynamics.^[40] 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 Daniel Dennett has used the analogy of the Game of 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.^{[41][42][43]}

From most random initial patterns 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 mathematical 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;^{[44][45]} many also produce one or more gliders or spaceships that travel indefinitely away from the initial location. Because of the nearest-neighbour 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 .

Simulation

History

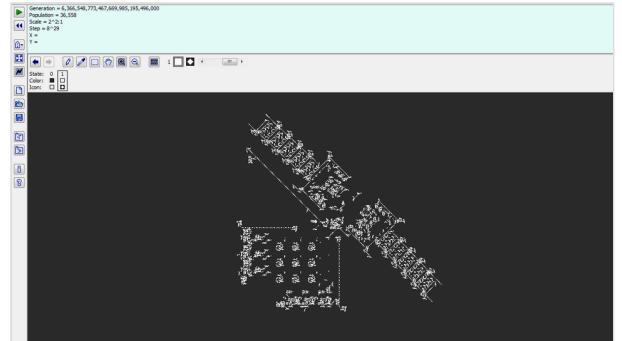
Computers have been used to follow and simulate the Game of Life since it was first publicized. When John Conway was first investigating how various starting configurations developed, he tracked them by hand using a go board with its black and white stones. This was tedious and prone to errors. The first interactive Game of Life program was written in an early version of ALGOL 68C 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, along with the statement: "Without its help, some discoveries about the game would have been difficult to make."^[1]

A color version of the Game of Life was written by Ed Hall in 1976 for Cromemco microcomputers, and a display from that program filled the cover of the June 1976 issue of Byte.^[46] The advent of microcomputer-based color graphics from Cromemco has been credited with a revival of interest in the game.^[47]

Two early implementations of the Game of Life on home computers were by Malcolm Banthorpe written in BBC BASIC. The first was in the January 1984 issue of Acorn User magazine, and Banthorpe followed this with a three-dimensional version in the May 1984 issue.^[48] Susan Stepney, Professor of Computer Science at the University of York, followed this up in 1988 with Life on the Line, a program that generated one-dimensional cellular automata.^[49]

Algorithms

Early patterns with unknown futures, such as the R-pentomino, led computer programmers to write programs to track the evolution of patterns in the Game of Life. Most of the early algorithms were similar: they represented the patterns as two-dimensional arrays in computer memory. Typically, two arrays are used: one to hold the current generation, and one 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 may swap roles so that the successor array in the last iteration becomes the current array in the next iteration, or one may copy the values of the second array into the first array then update the second array from the first array again.



The

6 366 548 773 467 669 985 195 496 000th
(6×10^{27}) 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

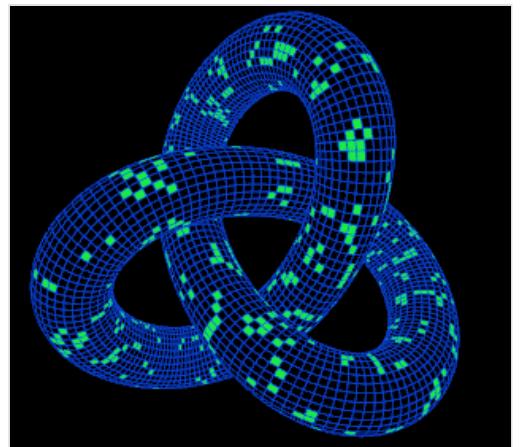
A variety of minor enhancements to this basic scheme are possible, and there are many ways to save unnecessary computation. A cell that did not change at the last time step, and none of whose neighbours changed, is guaranteed not to change at the current time step as well, so a program that keeps track of which areas are active can save time by not updating inactive zones.^[50]

To avoid decisions and branches in the counting loop, the rules can be rearranged from an egocentric approach of the inner field regarding its neighbours to a scientific observer's viewpoint: if the sum of all nine fields in a given neighbourhood is three, the inner field state for the next generation will be life; if the all-field sum is four, the inner field retains its current state; and every other sum sets the inner field to death.

To save memory, the storage can be reduced to one array plus two 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.

In principle, the Game of 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 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 there are no pathological edge effects. Techniques of dynamic storage allocation may also be used, creating ever-larger arrays to hold growing patterns. The Game of Life on a finite field is sometimes explicitly studied; some implementations, such as Golly, support a choice of the standard infinite field, a field infinite only in one dimension, or a finite field, with a choice of topologies such as a cylinder, a torus, or a Möbius strip.

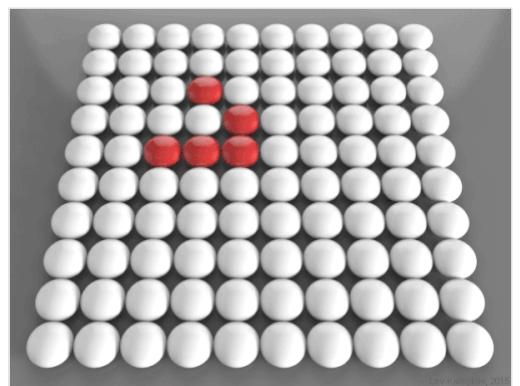
Alternatively, programmers may abandon the notion of representing the Game of Life field with a two-dimensional array, and use a different data structure, such as a vector of coordinate pairs representing live cells. This allows the pattern to move about the field unhindered, as long as the population 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.



The Game of Life on the surface of a toroidal trefoil knot



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



Red glider on a square lattice with periodic boundary conditions

For exploring large patterns at great time depths, sophisticated algorithms such as [Hashlife](#) may be useful. There is also a method for implementation of the Game of Life and other cellular automata using arbitrary asynchronous updates while still exactly [emulating](#) the behaviour of the synchronous game.^[51] [Source code](#) examples that implement the basic Game of Life scenario in various programming languages, including [C](#), [C++](#), [Java](#) and [Python](#) can be found at [Rosetta Code](#).^[52]

Modern Programs

There are now thousands of Game 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 the Game of Life, and a large library of interesting patterns in the Game of Life and other cellular automaton rules.

- [Golly](#) is a cross-platform (Windows, Macintosh, Linux, iOS, and Android) open-source simulation system for the Game of Life and other cellular automata (including all Life-like cellular automata, the Generations family of cellular automata from Mirek's Cellebration, and John von Neumann's 29-state cellular automaton) built by Andrew Trevorrow and Tomas Rokicki. It includes the Hashlife algorithm for extremely fast generation, and [Lua](#) or [Python](#) scriptability for both editing and simulation.
- [Mirek's Cellebration](#) is a freeware one- and two-dimensional cellular automata viewer, explorer, and editor for Windows. It includes powerful facilities for simulating and viewing a wide variety of cellular automaton rules, including the Game of Life, and a scriptable editor.
- [Xlife](#) is a cellular-automaton laboratory by Jon Bennett. The standard UNIX X11 Game of Life simulation application for a long time, it has also been ported to Windows. It can handle cellular automaton rules with the same neighbourhood as the Game of Life, and up to eight possible states per cell.^[53]

Variations

Since the Game of Life's inception, new, similar cellular automata have been developed. The standard Game of Life is symbolized in rule-string notation as B3/S23. A cell is born if it has exactly three neighbours, survives if it has two or three living neighbours, and 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 six neighbours, and lives on if there are either one or six 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 six neighbours, in addition to the original game's B3/S23 rule, causes a birth. HighLife is best known for its frequently occurring replicators.^{[54][55]}

Additional Life-like cellular automata exist. The vast majority of these 2^{18} different rules^[56] produce universes that are either too chaotic or too desolate to be of interest, but a large subset do display interesting behaviour. A further generalization produces the *isotropic* rulespace, with 2^{102}

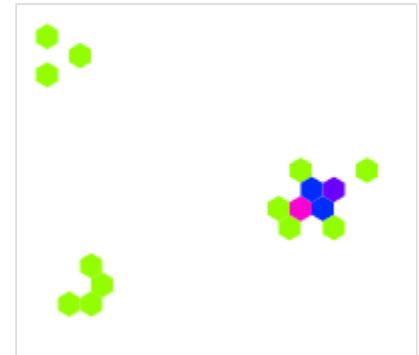
possible cellular automaton rules^[57] (the Game of Life again being one of them). These are rules that use the same square grid as the Life-like rules and the same eight-cell neighbourhood, and are likewise invariant under rotation and reflection. However, in isotropic rules, the positions of neighbour cells relative to each other may be taken into account in determining a cell's future state –not just the total number of those neighbours.

Some variations on The Game of Life modify the geometry of the universe as well as the rules. The above variations can be thought of as a two-dimensional square, because the world is two-dimensional and laid out in a square grid. One-dimensional square variations, known as elementary cellular automata,^[58] and three-dimensional square variations have been developed, as have two-dimensional hexagonal and triangular variations. A variant using aperiodic tiling grids has also been made.^[59]

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 the Game of 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 Sierpinski triangle when applied to a single live cell. The Sierpinski triangle can also be observed in the Game of Life by examining the long-term growth of an infinitely long single-cell-thick line of live cells,^[60] as well as in Highlife, Seeds (B2/S), and Stephen Wolfram's Rule 90.^[61]

Immigration is a variation that is very similar to the 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.^[62] Another similar variation, called QuadLife, involves four different on states. When a new cell is born from three different on neighbours, it takes the fourth value, and otherwise, like Immigration, it takes the majority value.^[63] Except for the variation among on cells, both of these variations act identically to the Game of Life.



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

In popular culture

Dr. Blob's Organism is a Shoot 'em up based on Conway's Life. In the game, Life continually generates on a group of cells within a "petri dish". The patterns formed are smoothed and rounded to look like a growing amoeba spewing smaller ones (actually gliders). Special "probes" zap the "blob" to keep it from overflowing the dish while destroying its nucleus.^[64]

Google implemented an easter egg of the Game of Life in 2012. Users who search for the term are shown an implementation of the game in the search results page.^[65]

Various musical composition techniques use the Game of Life, especially in MIDI sequencing.^[66] A variety of programs exist for creating sound from patterns generated in the Game of Life.^{[67][68][69]}

See also

- Artificial life – Field of study
- Glory Season – 1993 science fiction novel by David Brin, is set in a future society where the Game of Life is played in a competitive two-player mode
- Langton's ant – Two-dimensional Turing machine with emergent behavior
- Poietic Generator – Social network game played on a two-dimensional matrix, a "human" Game of Life.
- Self-organization § Computer science
- Of Man and Manta – Trilogy of science fiction novels by Piers Anthony; the novel 'OX' features a cellular automaton lifeform based on Game of Life
- LifeWiki – Wiki dedicated to Conway's Game of Life
- Boids – Artificial life program (simulation of flocking birds)

Notes

1. The simultaneity means that when each cell counts the number of live neighbours around it, it uses its neighbours' old states before the update, not their new states after the update. If the cells are instead updated in reading order, so that each cell uses the old states of the cells to its right and below it but the new states of the cells to its left and above it, a different cellular automaton results, which is known as NaiveLife^{[4][5]} because it is a common beginners' mistake among people attempting to program Conway's Game of Life.^[6]

References

1. Gardner, Martin (October 1970). "The fantastic combinations of John Conway's new solitaire game 'life'" (<https://web.stanford.edu/class/sts145/Library/life.pdf>) (PDF). Mathematical Games. *Scientific American*. Vol. 223, no. 4. pp. 120–123. doi:10.1038/scientificamerican1070-120 (<https://doi.org/10.1038%2Fscientificamerican1070-120>). JSTOR 24927642 (<https://www.jstor.org/stable/24927642>). Archived (<https://ghostarchive.org/archive/20221009/https://web.stanford.edu/class/sts145/Library/life.pdf>) (PDF) from the original on 2022-10-09.
2. Berlekamp, E. R.; Conway, John Horton; Guy, R. K. (2001–2004). *Winning Ways for your Mathematical Plays* (2nd ed.). A K Peters Ltd.

3. Izhikevich, Eugene M.; Conway, John H.; Seth, Anil (2015-06-21). "Game of Life" ([http://doi.org/10.4249%2Fscholarpedia.1816](https://doi.org/10.4249%2Fscholarpedia.1816)). *Scholarpedia*. **10** (6): 1816. Bibcode:2015SchpJ..10.1816I (<https://ui.adsabs.harvard.edu/abs/2015SchpJ..10.1816I>). doi:10.4249/scholarpedia.1816 (<https://doi.org/10.4249%2Fscholarpedia.1816>). ISSN 1941-6016 (<https://search.worldcat.org/issn/1941-6016>).
4. "NaiveLife Emulated: A reading-order simulation of Life" (<https://conwaylife.com/forums/viewtopic.php?f=11&t=4523&p=128919>). *ConwayLife.com*. 24 May 2020. Retrieved 29 November 2021.
5. Goucher, Adam. "Re: Thread For Your Accidental Discoveries" (<https://conwaylife.com/forums/viewtopic.php?f=2&t=279&p=13168#p13168>). *ConwayLife.com*. Archived (<https://web.archive.org/web/20211129154301/https://www.conwaylife.com/forums/viewtopic.php?f=2&t=279&p=13168#p13168>) from the original on 29 November 2021. Retrieved 29 November 2021.
6. Ian07. "Re: Strange spaceship that is supposed to be impossible and infinite cell spread" (<https://web.archive.org/web/20211129164454/https://conwaylife.com/forums/viewtopic.php?f=9&t=4288&p=88701#p88701>). *ConwayLife.com*. Archived from the original (<https://conwaylife.com/forums/viewtopic.php?f=9&t=4288&p=88701#p88701>) on 29 November 2021. Retrieved 29 November 2021. "*I'm pretty sure this is because you've accidentally created an implementation of what's sometimes known as NaiveLife (as it's a common mistake made by many people coding CGoL for the first time):*"
7. Conway, private communication to the 'Life list', 14 April 1999.
8. 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" (<https://web.archive.org/web/20090906014935/http://www.igblan.free-online.co.uk/igblan/ca/>). Archived from the original (<http://www.igblan.free-online.co.uk/igblan/ca/>) on 6 September 2009. Retrieved 12 July 2009.
9. Paul Rendell (January 12, 2005). "A Turing Machine in Conway's Game of Life" (<http://rendell-attic.org/gol/tm.htm>). Archived (<https://web.archive.org/web/20190417075720/http://rendell-attic.org/gol/tm.htm>) from the original on April 17, 2019. Retrieved July 12, 2009.
10. Adam P. Goucher. "Spartan universal computer-constructor" (https://conwaylife.com/wiki/Spartan_universal_computer-constructor). LifeWiki. Retrieved December 5, 2021.
11. "R-pentomino" (<https://conwaylife.com/wiki/R-pentomino>). LifeWiki. 1983. pp. 219, 223. Archived (<https://web.archive.org/web/20211206012009/https://www.conwaylife.com/wiki/R-pentomino>) from the original on December 6, 2021. Retrieved December 5, 2021.
12. Stephen A. Silver. "Glider" (https://conwaylife.com/ref/lexicon/lex_g.htm#glider). The Life Lexicon. Retrieved March 4, 2019.

13. "Census Results in Conway's Game of Life" (<https://web.archive.org/web/20090910010855/https://conwaylife.com/soup/census.asp?rule=B3%2FS23&sl=1&os=1&ss=1>). The Online Life-Like CA Soup Search. Archived from the original (<https://conwaylife.com/soup/census.asp?rule=B3/S23&sl=1&os=1&ss=1>) on 2009-09-10. Retrieved July 12, 2009.
14. "Spontaneous appeared Spaceships out of Random Dust" (<http://wwwhomes.uni-bielefeld.de/achim/moving.html>). Achim Flammenkamp (1995-12-09). Archived (<https://web.archive.org/web/20090413192821/http://wwwhomes.uni-bielefeld.de/achim/moving.html>) from the original on 2009-04-13. Retrieved July 10, 2012.
15. Stephen A. Silver. "Pulsar" (https://conwaylife.com/ref/lexicon/lex_p.htm#pulsar). The Life Lexicon. Retrieved March 4, 2019.
16. Brown, Nico; Cheng, Carson; Jacobi, Tanner; Karpovich, Maia; Merzenich, Matthias; Raucci, David; Riley, Mitchell (5 December 2023). "Conway's Game of Life is Omniprimitive". arXiv:2312.02799 (<https://arxiv.org/abs/2312.02799>) [math.CO (<https://arxiv.org/archive/math.CO>)].
17. "LifeWiki:Game of Life Status page - LifeWiki" (https://conwaylife.com/wiki/LifeWiki:Game_of_Life_Status_page). conwaylife.com. Retrieved 2023-12-16.
18. Stone, Alex (2024-01-18). "Math's 'Game of Life' Reveals Long-Sought Repeating Patterns" (<https://www.quantamagazine.org/math-game-of-life-reveals-long-sought-repeating-patterns-20240118/>). Quanta Magazine. Archived (<https://web.archive.org/web/20240118161936/https://www.quantamagazine.org/math-game-of-life-reveals-long-sought-repeating-patterns-20240118/>) from the original on 2024-01-18. Retrieved 2024-01-18.
19. Achim Flammenkamp (2004-09-07). "Most seen natural occurring ash objects in Game of Life" (http://wwwhomes.uni-bielefeld.de/achim/freq_top_life.html). Archived (https://web.archive.org/web/20081022033319/http://wwwhomes.uni-bielefeld.de/achim/freq_top_life.html) from the original on 2008-10-22. Retrieved 2008-09-16.
20. Stephen A. Silver. "Diehard" (https://conwaylife.com/ref/lexicon/lex_d.htm#diehard). The Life Lexicon. Retrieved March 4, 2019.
21. Koenig, H. (February 21, 2005). "New Methuselah Records" (https://web.archive.org/web/20190910130327/https://pentadecathlon.com/lifeNews/2005/02/new_methuselah_records.html). Archived from the original (https://pentadecathlon.com/lifeNews/2005/02/new_methuselah_records.html) on September 10, 2019. Retrieved January 24, 2009.
22. Stephen A. Silver. "Gosper glider gun" (https://conwaylife.com/ref/lexicon/lex_g.htm#gosperglidergun). The Life Lexicon. Retrieved March 4, 2019.
23. The Hunting of the New Herschel Conduits (<https://conwaylife.com/forums/viewtopic.php?f=2&t=1599&start=200#p19125>) Archived (<https://web.archive.org/web/20220224001858/https://conwaylife.com/forums/viewtopic.php?f=2&t=1599&start=200#p19125>) 2022-02-24 at the Wayback Machine, ConwayLife forums, April 28th, 2015, posts by Michael Simkin ("simsim314") and Dongook Lee ("Scorbie").

24. "Block-laying switch engine" (https://conwaylife.com/wiki/Block-laying_switch_engine). LifeWiki. Retrieved December 5, 2021.
25. Stephen A. Silver. "Infinite Growth" (https://conwaylife.com/ref/lexicon/lex_i.htm#infinitegrowth). The Life Lexicon. Retrieved March 4, 2019.
26. Stephen A. Silver. "Rake" (https://conwaylife.com/ref/lexicon/lex_r.htm#rake). The Life Lexicon. Archived (https://web.archive.org/web/20190301232016/http://www.conwaylife.com/ref/lexicon/lex_r.htm#rake) from the original on March 1, 2019. Retrieved March 4, 2019.
27. "Programmable computer" (<https://conwaylife.com/forums/viewtopic.php?f=2&t=2561#p37428>). conwaylife.com forums. Retrieved August 23, 2018.
28. "A Turing Machine in Conway's Game of Life, extendable to a Universal Turing Machine" (<http://rendell-attic.org/gol/tm.htm>). Paul Rendell. Archived (<https://web.archive.org/web/20190417075720/http://rendell-attic.org/gol/tm.htm>) from the original on April 17, 2019. Retrieved August 23, 2018.
29. "Build a working game of Tetris in Conway's Game of Life" (<https://codegolf.stackexchange.com/questions/11880/build-a-working-game-of-tetris-in-conways-game-of-life/142673#142673>). StackExchange. Retrieved August 23, 2018.
30. Aron, Jacob (16 June 2010). "First replicating creature spawned in life simulator" (<https://www.newscientist.com/article/mg20627653.800-first-replicating-creature-spawned-in-life-simulator.html>). *New Scientist*. Retrieved 12 October 2013.
31. "Gemini – LifeWiki" (<https://conwaylife.com/wiki/Gemini>). Conwaylife.com. Retrieved 2013-10-16.
32. "Universal Constructor Based Spaceship" (<https://conwaylife.com/forums/viewtopic.php?f=2&t=399&p=2327#p2327>). Conwaylife.com. Retrieved 2012-06-24.
33. "Demonoid" (<https://conwaylife.com/wiki/Demonoid>). LifeWiki. Retrieved 18 June 2016.
34. "Elementary knightship" (<https://conwaylife.com/forums/viewtopic.php?f=2&t=3303>). Retrieved 9 March 2018.
35. "Elementary" (<https://conwaylife.com/wiki/Elementary>), LifeWiki. Retrieved 2018-11-21
36. "Geminoid Challenge" (<https://conwaylife.com/forums/viewtopic.php?f=2&t=1006&p=9917#p9901>). Conwaylife.com. Retrieved 2015-06-25.
37. Passe-Science (2019-05-29). "Automate Cellulaire - Passe-science #27" (<https://www.youtube.com/watch?v=CfRSVPhzN5M>). Archived (<https://ghostarchive.org/archive/youtube/20211211/CfRSVPhzN5M>) from the original on 2021-12-11. Retrieved 2019-06-25 – via YouTube.
38. appoucher (2018-11-12). "Fully self-directed replication" (<https://cp4space.wordpress.com/2018/11/12/fully-self-directed-replication/>). *Complex Projective 4-Space*. Retrieved 2019-06-25.
39. "OEOP metacell - LifeWiki" (<https://conwaylife.com/wiki/OEOP>). Conwaylife.com. Retrieved 2019-06-24.

40. Alstrøm, Preben; Leão, João (1994-04-01). "Self-organized criticality in the game of Life" (<https://link.aps.org/doi/10.1103/PhysRevE.49.R2507>). *Physical Review E*. **49** (4): R2507–R2508. Bibcode:1994PhRvE..49.2507A (<https://ui.adsabs.harvard.edu/abs/1994PhRvE..49.2507A>). doi:10.1103/PhysRevE.49.R2507 (<https://doi.org/10.1103%2FPhysRevE.49.R2507>). PMID 9961636 (<https://pubmed.ncbi.nlm.nih.gov/9961636/>).
41. Dennett, D. C. (1991). *Consciousness Explained* (<https://archive.org/details/consciousnessexp00denn>). Boston: Back Bay Books. ISBN 978-0-316-18066-5.
42. Dennett, D.C. (1995). *Darwin's Dangerous Idea: Evolution and the Meanings of Life* (<https://archive.org/details/darwinstdangerous0000denn>). New York: Simon & Schuster. ISBN 978-0-684-82471-0.
43. Dennett, D.C. (2003). *Freedom Evolves*. New York: Penguin Books. ISBN 978-0-14-200384-8.
44. Andrzej Okrasinski. "Game of Life Object Statistics" (<https://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.
45. Nathaniel Johnston. "The Online Life-Like CA Soup Search" (<https://web.archive.org/web/20090910010849/https://conwaylife.com/soup/>). Archived from the original (<https://conwaylife.com/soup/>) on 2009-09-10. Retrieved July 12, 2009.
46. Helmers, Carl (June 1976). "About the Cover" (https://archive.org/stream/bytemagazine-1976-06/1976_06_BYTE_00-10_The_Game_of_LIFE_in_Color#page/n7/mode/2up). *Byte*. No. 10. pp. 6–7. Retrieved February 18, 2013.
47. McIntosh, Harold (2008). "Introduction" (http://comunidad.escom.ipn.mx/LCCOMP/Announces/Entries/2015/12/23_LCCOMP_Obituaries_2015_files/181-186pp%20JCA-HM07-00.pdf) (PDF). *Journal of Cellular Automata*. **13**: 181–186. Archived (https://ghostarchive.org/archive/20221009/http://comunidad.escom.ipn.mx/LCCOMP/Announces/Entries/2015/12/23_LCCOMP_Obituaries_2015_files/181-186pp%20JCA-HM07-00.pdf) (PDF) from the original on 2022-10-09. Retrieved 3 November 2021.
"With the advent of microcomputers and Cromemco's graphics board, Life became a favorite display program for video monitors and led to a revival of interest in the game."
48. "Acorn User Magazine Scans" (<http://8bs.com/aumags.htm>). The BBC and Master Computer Public Domain Library. Archived (<https://archive.today/20130116100942/http://8bs.com/aumags.htm>) from the original on 2013-01-16. Retrieved 2018-12-29.
49. Stepney, Susan. "AcornUser articles" (<https://www-users.cs.york.ac.uk/susan/bib/ss/au.htm>). www-users.cs.york.ac.uk. AcornUser. Retrieved 2018-12-29.
50. Alan Hensel. "About my Conway's Game of Life Applet" (<http://www.ibiblio.org/lifepatterns/lifeapplet.html>). Archived (<https://web.archive.org/web/20090716231902/http://www.ibiblio.org/lifepatterns/lifeapplet.html>) from the original on July 16, 2009. Retrieved July 12, 2009.

51. Nehaniv, Chrystopher L. (15–18 July 2002). *Self-Reproduction in Asynchronous Cellular Automata*. 2002 NASA/DoD Conference on Evolvable Hardware (<https://ieeexplore.ieee.org/xpl/conhome/8000/proceeding>). Alexandria, Virginia, USA: IEEE Computer Society Press. pp. 201–209. doi:10.1109/EH.2002.1029886 (<https://doi.org/10.1109%2FEH.2002.1029886>). hdl:2299/6834 (<https://hdl.handle.net/2299%2F6834>). ISBN 0-7695-1718-8.
52. "Conway's Game of Life" (https://rosettacode.org/wiki/Conway%27s_Game_of_Life). Rosetta Code. June 7, 2024. Archived (https://web.archive.org/web/20240718213019/https://rosettacode.org/wiki/Conway%27s_Game_of_Life) from the original on July 18, 2024. Retrieved July 2, 2024.
53. "Xlife - LifeWiki" (<https://conwaylife.com/wiki/Xlife>). conwaylife.com.
54. HighLife – An Interesting Variant of Life (<http://www.tip.net.au/~dbell/articles/HighLife.zip>) by David Bell (.zip file)
55. Stephen A. Silver. "Replicator" (https://conwaylife.com/ref/lexicon/lex_r.htm#replicator). The Life Lexicon. Archived (https://web.archive.org/web/20190301232016/http://www.conwaylife.com/ref/lexicon/lex_r.htm#replicator) from the original on March 1, 2019. Retrieved March 4, 2019.
56. "Life-like cellular automata - LifeWiki" (https://conwaylife.com/wiki/Life-like#Life-like_cellular_automata). Conwaylife.com. Archived (https://web.archive.org/web/20190306043758/http://conwaylife.com/wiki/Life-like#Life-like_cellular_automata) from the original on March 6, 2019. Retrieved March 4, 2019.
57. "Isotropic - LifeWiki" (<https://conwaylife.com/wiki/Isotropic>). Conwaylife.com. Archived (<https://web.archive.org/web/20190306043649/http://conwaylife.com/wiki/Isotropic>) from the original on March 6, 2019. Retrieved March 4, 2019.
58. "Elementary Cellular Automaton" (<http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>). Wolfram Mathworld. Archived (<https://web.archive.org/web/20090703200815/http://mathworld.wolfram.com/ElementaryCellularAutomaton.html>) from the original on July 3, 2009. Retrieved July 12, 2009.
59. "First gliders navigate ever-changing Penrose universe" (<https://www.newscientist.com/article/dn22134-first-gliders-navigate-everchanging-penrose-universe.html>). New Scientist.
60. Stephen Wolfram, *A New Kind of Science* online, Note (f) for structures in class 4 systems: Structures in the Game of Life (<https://www.wolframscience.com/nks/note-s-6-8--structures-in-the-game-of-life/>): "A simpler kind of unbounded growth occurs if one starts from an infinite line of black cells. In that case, the evolution is effectively 1D, and turns out to follow elementary rule 22"
61. "Life Imitates Sierpinski" (<https://conwaylife.com/forums/viewtopic.php?f=7&t=90>). ConwayLife.com forums. Retrieved July 12, 2009.
62. Stephen A. Silver. "Immigration" (https://conwaylife.com/ref/lexicon/lex_i.htm#immigration). The Life Lexicon. Retrieved March 4, 2019.

63. Stephen A. Silver. "QuadLife" (https://conwaylife.com/ref/lexicon/lex_q.htm#quadlife). The Life Lexicon. Retrieved March 4, 2019.
64. Cohen, Peter (23 September 2003). "Dr. Blob's Organism oozes onto Mac OS X" (<http://www.macworld.com/article/168098/organism.html>). *Macworld*. International Data Group. Archived (<https://web.archive.org/web/20210515185231/https://www.macworld.com/article/168098/organism.html>) from the original on 15 May 2021. Retrieved 8 July 2025.
65. Wasserman, Todd (12 July 2012). "Type 'Conway's Game of Life' on Google and See What Happens" (<https://mashable.com/2012/07/12/conways-game-of-life-google/>). *Mashable*. Retrieved 1 May 2020.
66. Burraston, Dave; Edmonds, Ernest; Livingstone, Dan; Miranda, Eduardo Reck (2004). "Cellular Automata in MIDI based Computer Music" (<http://quod.lib.umich.edu/i/icmc/c/bbp2372.2004.047?view=image>). *Proceedings of the 2004 International Computer Music Conference*. CiteSeerX 10.1.1.6.3882 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.3882>). hdl:10453/1425 (<https://hdl.handle.net/10453%2F1425>). ISBN 9780971319226. Archived (<https://web.archive.org/web/20230111115704/https://quod.lib.umich.edu/i/icmc/bbp2372.2004.047?view=image>) from the original on 2023-01-11. Retrieved 2012-07-05.
67. "glitchDS – Cellular Automaton Sequencer For The Nintendo DS" (<http://www.synthtopia.com/content/2008/05/29/glitchds-cellular-automaton-sequencer-for-the-nintendo-ds/>). Synthtopia.com. 2008-05-29. Archived (<https://web.archive.org/web/20120726170457/http://www.synthtopia.com/content/2008/05/29/glitchds-cellular-automaton-sequencer-for-the-nintendo-ds/>) from the original on 2012-07-26. Retrieved 2012-06-24.
68. "Game Of Life Music Sequencer" (<http://www.synthtopia.com/content/2009/04/29/game-of-life-music-sequencer/>). Synthtopia.com. 2009-04-29. Archived (<https://web.archive.org/web/20120726113942/http://www.synthtopia.com/content/2009/04/29/game-of-life-music-sequencer/>) from the original on 2012-07-26. Retrieved 2012-06-24.
69. "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/>). Synthtopia.com. 2011-01-12. Archived (<https://web.archive.org/web/20120726014246/http://www.synthtopia.com/content/2011/01/12/game-of-life-music-sequencer-for-ios-runxt-life/>) from the original on 2012-07-26. Retrieved 2012-06-24.

External links

- [Life Lexicon](https://conwaylife.com/ref/lexicon/lex_home.htm) (https://conwaylife.com/ref/lexicon/lex_home.htm) *conwaylife.com*: extensive lexicon with many patterns
- [LifeWiki](https://conwaylife.com/wiki/) (<https://conwaylife.com/wiki/>) *conwaylife.com*
- [ConwayLife.com Forums](https://conwaylife.com/forums/) (<https://conwaylife.com/forums/>) *conwaylife.com*

- [Catagolue](https://catagolue.hatsya.com/home) (<https://catagolue.hatsya.com/home>) *catagolue.hatsya.com*: online database of objects in Conway's Game of Life and similar cellular automata
 - [Cellular Automata FAQ – Conway's Game of Life](https://cafaq.com/lifeFAQ/index.php) (<https://cafaq.com/lifeFAQ/index.php>) *cafaq.com*
 - [Algebraic formula](https://uk.mathworks.com/matlabcentral/fileexchange/80176-conways-game-of-life-equation) (<https://uk.mathworks.com/matlabcentral/fileexchange/80176-conways-game-of-life-equation>) *uk.mathworks.com*: recurrence relation for iterating Conway's Game of Life.
-

Retrieved from "https://en.wikipedia.org/w/index.php?title=Conway%27s_Game_of_Life&oldid=1333414362"