Smalltalk: Conway's Game of Life

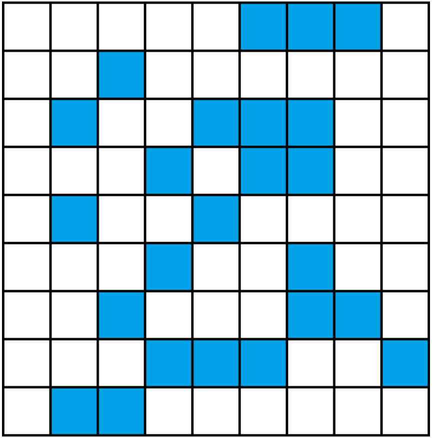
Brian Heinold

Mount St. Mary's University

February 16, 2018

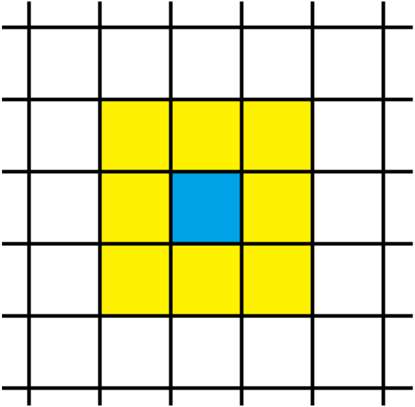
Board

The "game" consists of a grid of cells. Each cell is either on (alive) or off (dead).

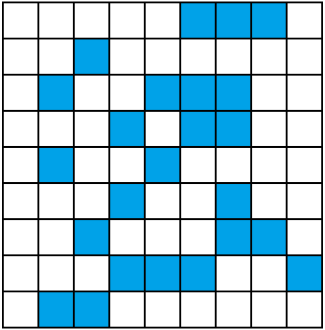


Neighbors

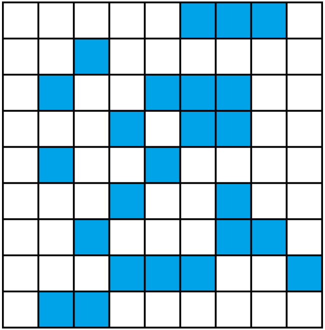
A cell has 8 neighbors (above, below, left, right, and diagonals).



Rules

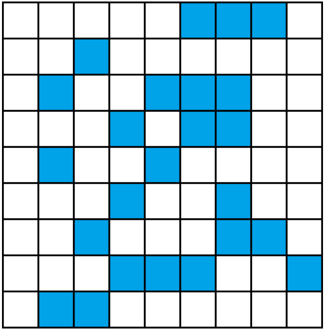


Rules



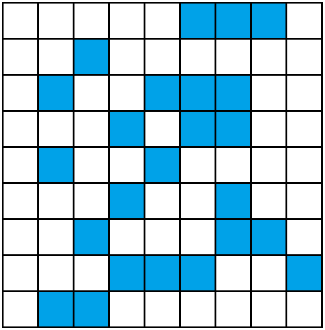
1. Live cells stay alive if 2 or 3 of their neighbors are alive.

Rules



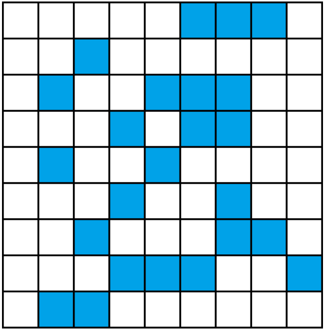
1. Live cells stay alive if 2 or 3 of their neighbors are alive.
2. Live cells with under 2 live neighbors die of loneliness.

Rules



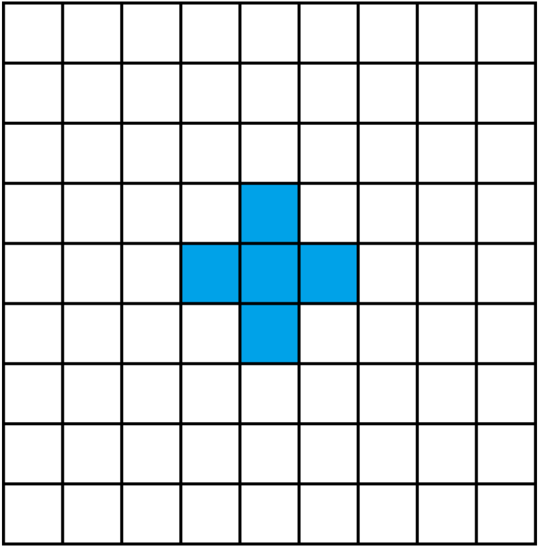
1. Live cells stay alive if 2 or 3 of their neighbors are alive.
2. Live cells with under 2 live neighbors die of loneliness.
3. Live cells with over 3 live neighbors die of overcrowding.

Rules

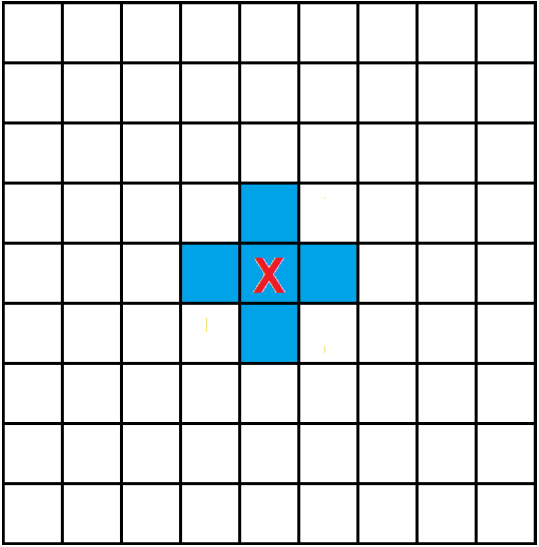


1. Live cells stay alive if 2 or 3 of their neighbors are alive.
2. Live cells with under 2 live neighbors die of loneliness.
3. Live cells with over 3 live neighbors die of overcrowding.
4. Dead cells come to life if exactly 3 neighbors are alive.

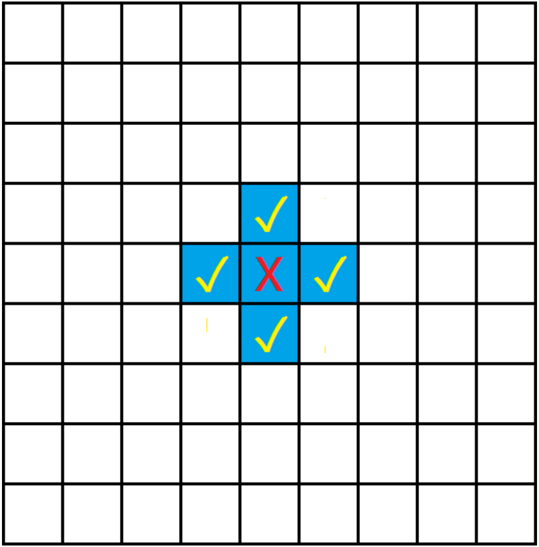
A walk-through — five live cells to start



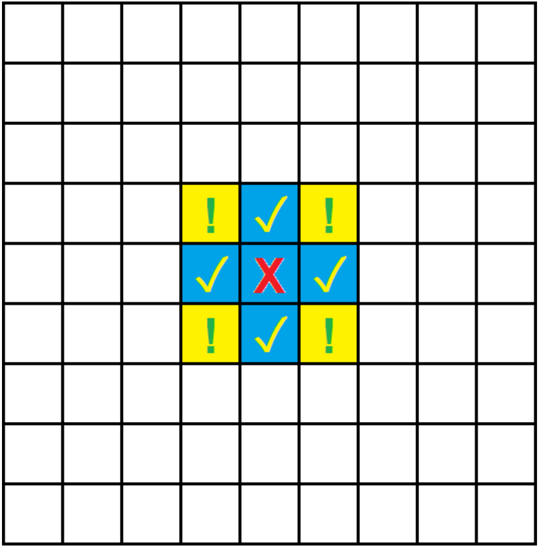
A walk-through — middle cell dies of overcrowding



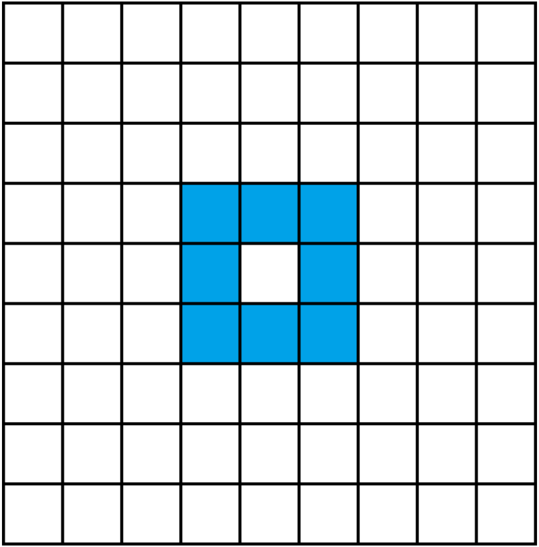
A walk-through — the other live cells are okay



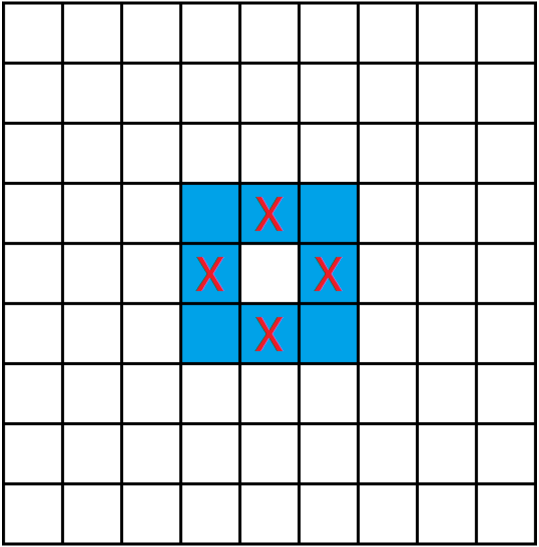
A walk-through — four new cells come to life!



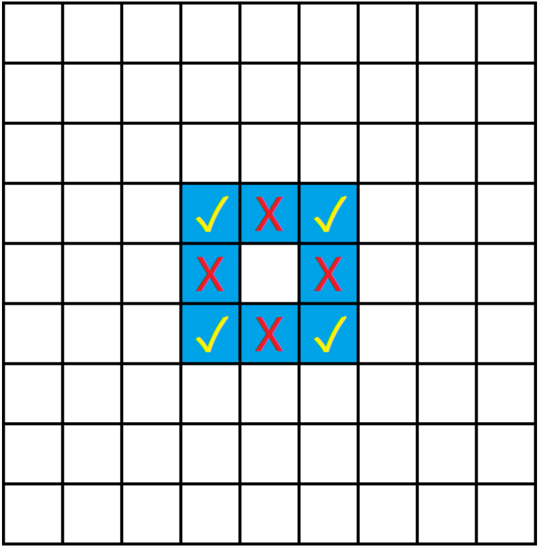
A walk-through — the next generation



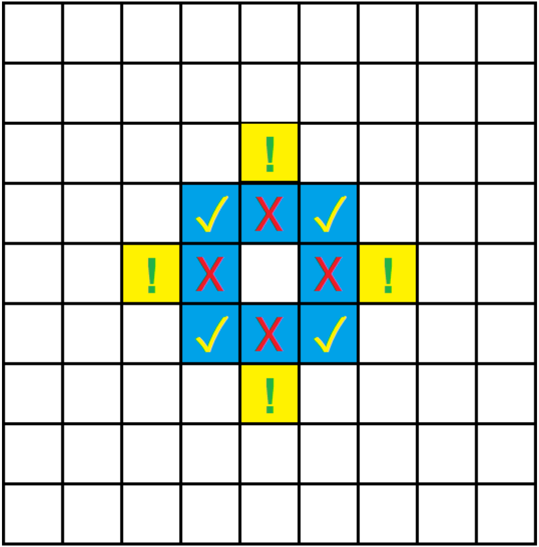
A walk-through — four cells die of overcrowding



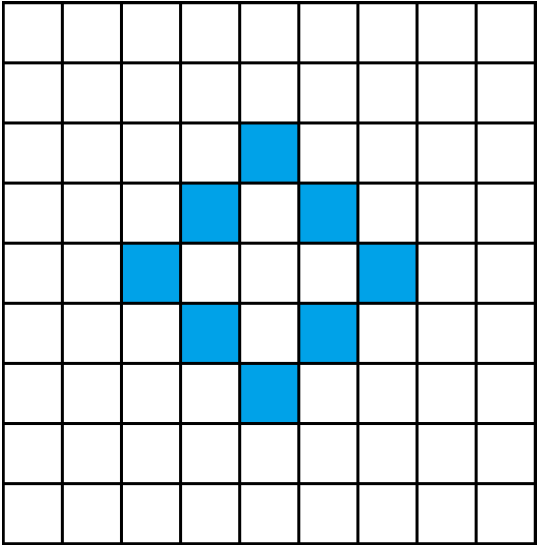
A walk-through — the other live cells are okay



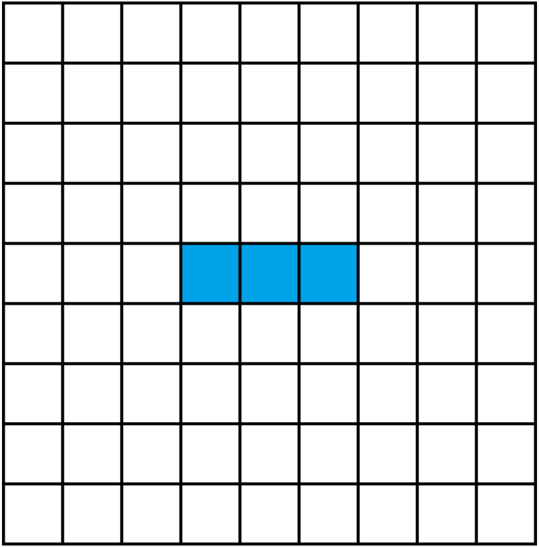
A walk-through — four new cells come to life



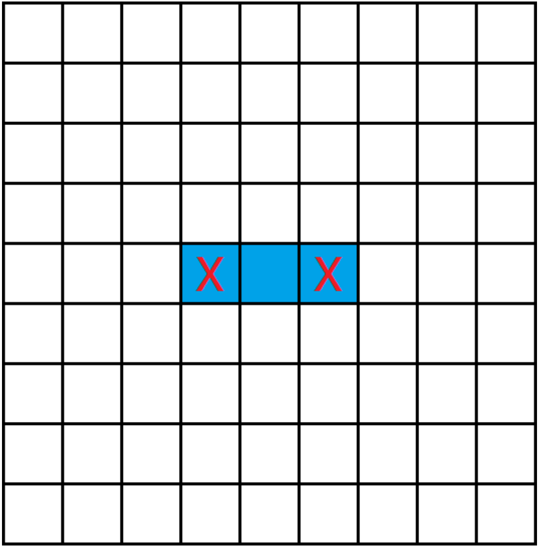
A walk-through — the third generation



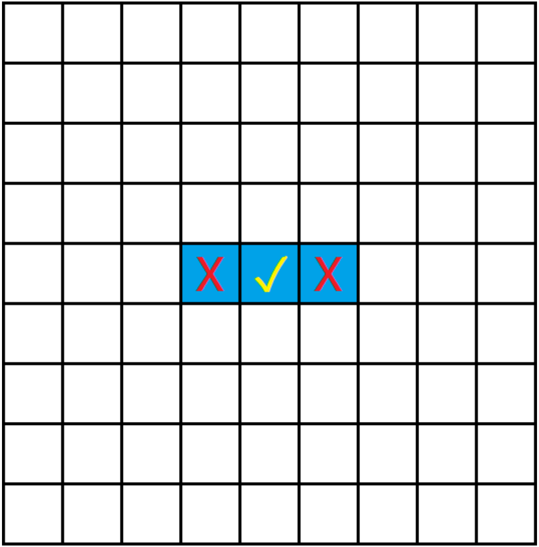
Another walk-through



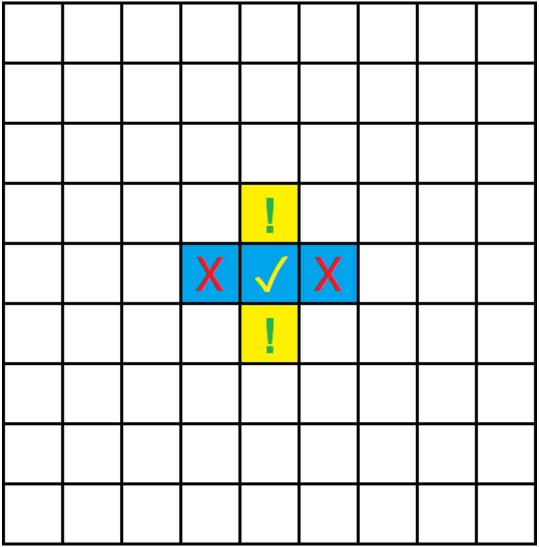
Another walk-through — outside cells die of loneliness



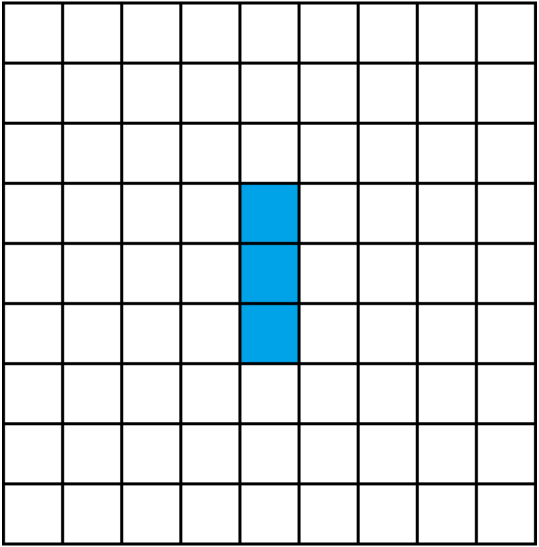
Another walk-through — middle cell is okay



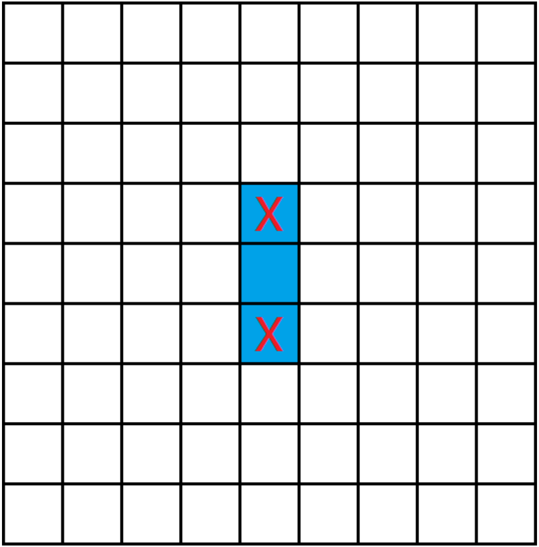
Another walk-through — two new cells come to life



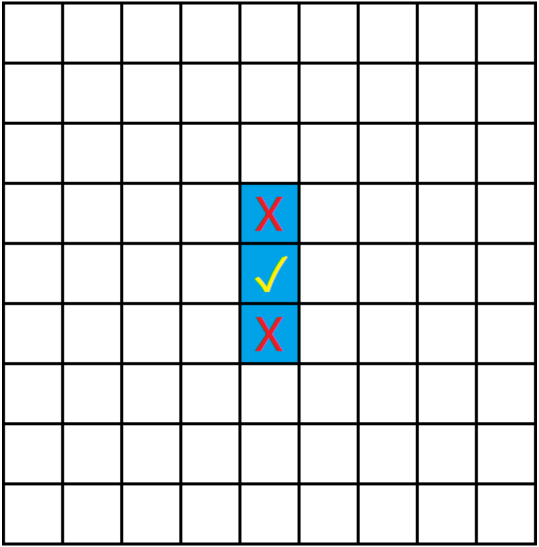
Another walk-through — the next generation



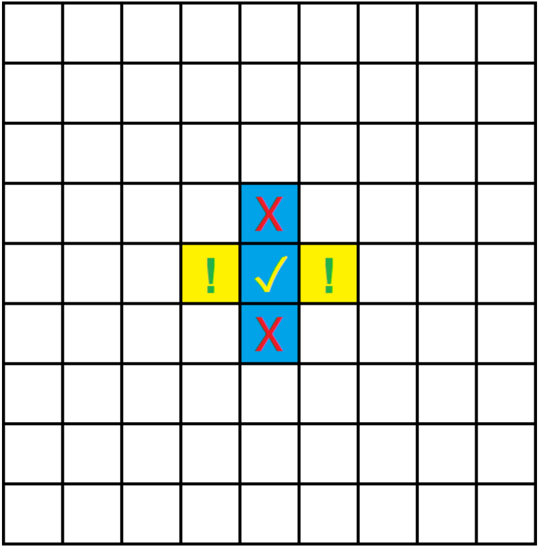
Another walk-through — outside cells die of loneliness



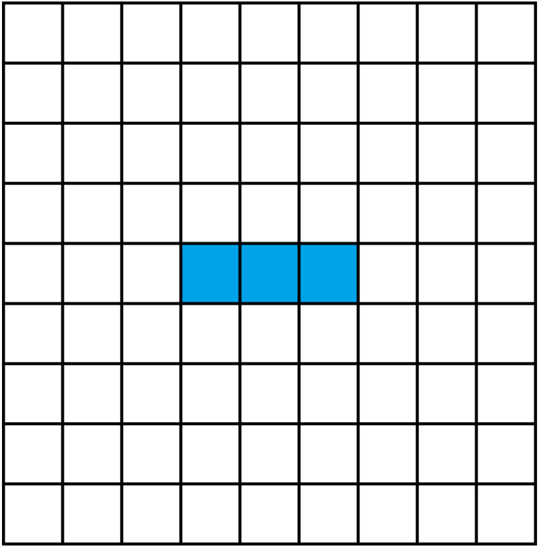
Another walk-through — middle cell is okay



Another walk-through — two new cells come to life

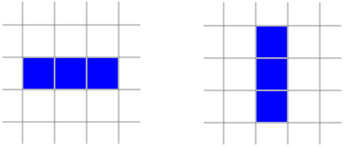


Another walk-through — back where we started!

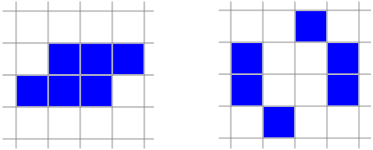


Oscillating patterns

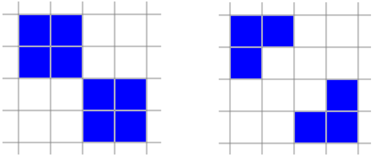
- Blinker



- Toad



- Beacon



Demonstration

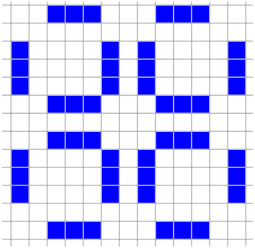
Throughout this talk, I'll be switching back and forth between slides and demonstrations.

If you want to try it out for yourself, an online app is here:

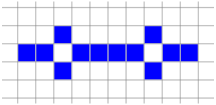
`https://brianheinold.net/life.html`

More oscillators

Pulsar (period 3):



Pentadecathlon (period 15)

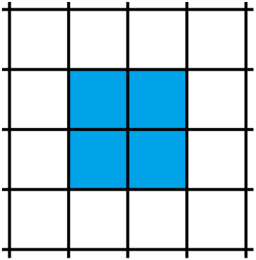


Even more oscillators

From `http://www.conwaylife.com/wiki/Oscillator`

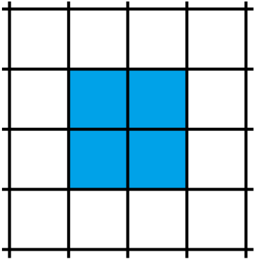
Period	First discovered	Discoverer	Year of discovery		Smallest known	Minimum # of cells
2	blinker	John Conway	1970		blinker	3
3	pulsar	John Conway	1970		caterer	12
4	pinwheel	Simon Norton	1970		mazing, mold	12
5	octagon 2	Sol Goodman, Arthur Taber	1971		pseudo-barberpole	15
6	$rats	David Buckingham	1972		unix	16
7	burloaferimeter	David Buckingham	1972		28P7.1, 28P7.2, 28P7.3, burloaferimeter	28
8	figure eight	Simon Norton	1970		figure eight	12
9	worker bee	David Buckingham	1972		29P9	29
10	42P10.1	David Buckingham	≤1976		24P10	24
11	38P11.1	David Buckingham	1977		rattlesnake	33
12	dinner table	Robert Wainwright	1972		dinner table	33
13	65P13.1	David Buckingham	1976		34P13	34
14	tumbler	George Collins, Jr.	1970		tumbler	16
15	pentadecathlon	John Conway	1970		pentadecathlon	12
16	two pre-L hasslers	Robert Wainwright	1983		Achim's p16	32
17	54P17.1	Dean Hickerson	1997		Honey thieves	36
18	117P18	David Buckingham	≤1991		43P18	43
19	*None found*					
20	145P20	Noam Elkies	1995		mold on fumarole	30

A *still life*



This is stable. It will not change unless something affects it from outside.

A *still life*

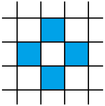


This is stable. It will not change unless something affects it from outside.

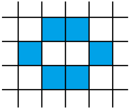
Every live cell has 3 live neighbors; every dead cell has 2 live neighbors.

More still lifes

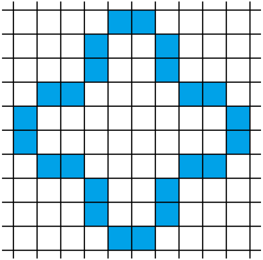
- Tub



- Beehive



- Lake 2

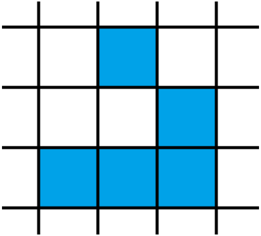


More still lifes

From http://www.conwaylife.com/wiki/Still_life

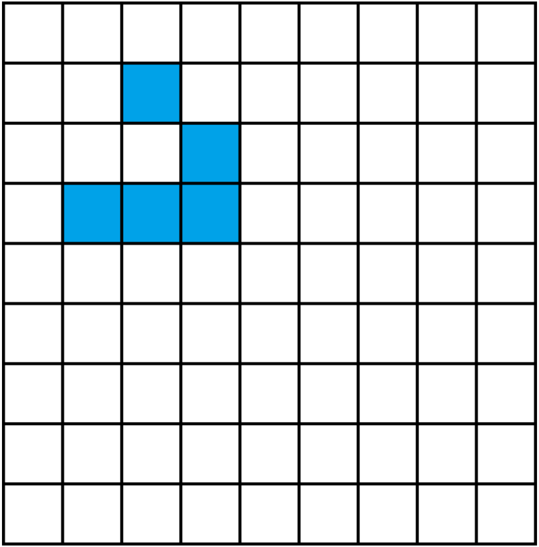
Live cells	Strict still lifes			Pseudo still lifes			Quasi still lifes		
	Count (A019473)	Examples	List	Count (A056613)	Examples	List	Count	Examples	List
1	0			0			0		
2	0			0			0		
3	0			0			0		
4	2	block, tub	Full list	0			0		
5	1	boat	Full list	0			0		
6	5	beehive, ship	Full list	0			0		
7	4	eater 1, loaf	Full list	0			0		
8	9	canoe, pond	Full list	1	bi-block	Full list	6		
9	10	hat, integral sign	Full list	1	block on boat	Full list	13		
10	25	boat-tie, loop	Full list	7	bi-boat	Partial list	57		
11	46	elevener	Full list	16			141		
12	121	honeycomb, table on table	Partial list	55			465		
13	240	sesquihat	Partial list	110			1,224		
14	619	fourteener, paperclip	Partial list	279			3,956		
15	1,353	moose antlers	Partial list	620			11,599		
16	3,286	bi-cap, scorpion	Partial list	1,645	pond on pond	Partial list	36,538		
17	7,773	twin hat	Partial list	4,067			107,415		
18	19,044	dead spark coil	Partial list	10,843			327,250		

Glider

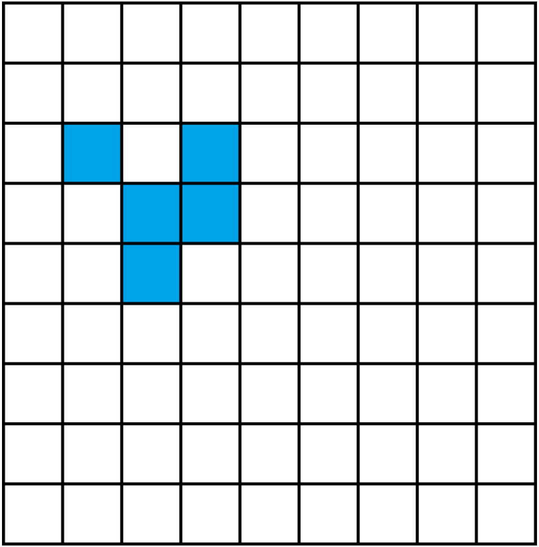


This pattern "glides". It oscillates, but moves as it oscillates. It is extremely important.

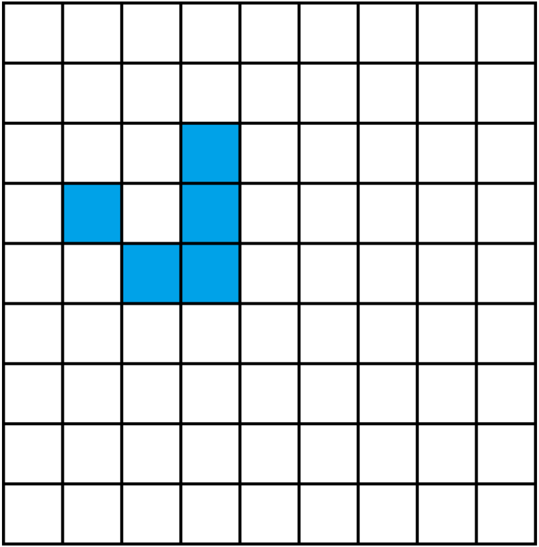
Glider moving



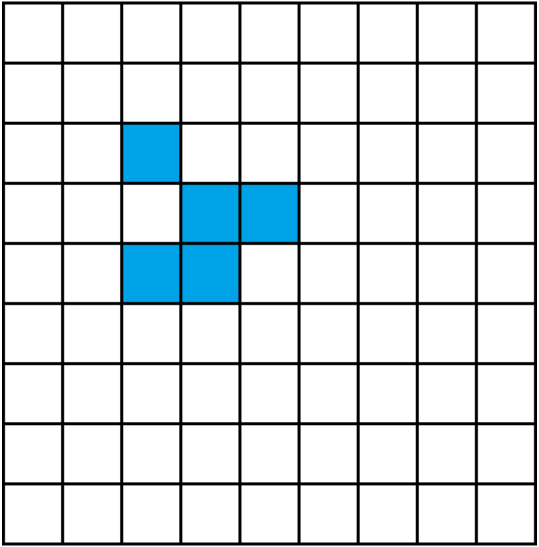
Glider moving



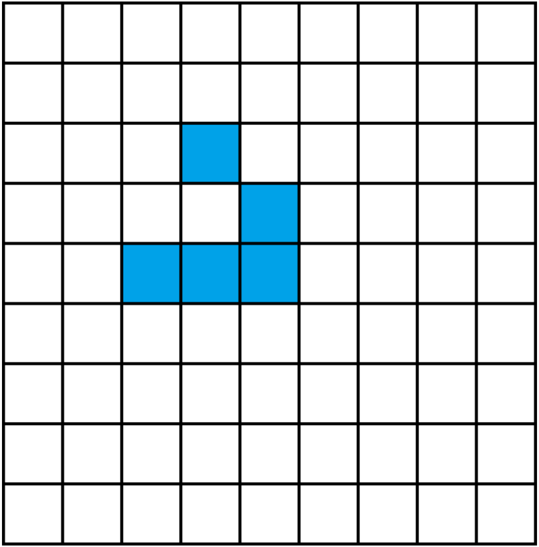
Glider moving



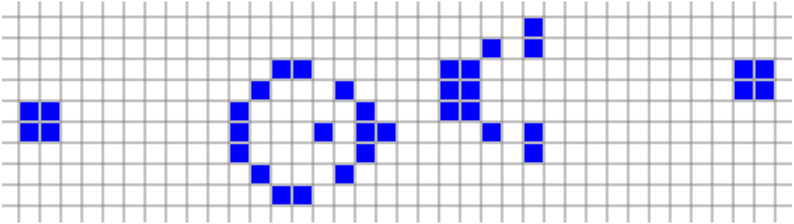
Glider moving



Glider moving

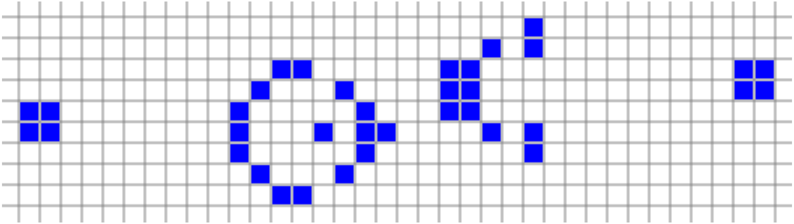


Glider gun



This pattern generates gliders.

Glider gun

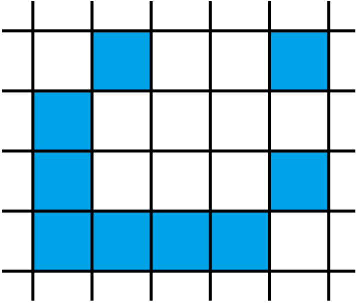


This pattern generates gliders.

It shows that infinite growth is possible (starting from a finite number of cells, it is possible to generate an unlimited amount).

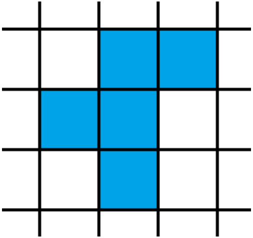
Other gliding patterns

There are various other types of gliding patterns, like this one, called a *lightweight spaceship*:



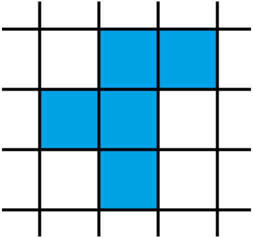
R-Pentomino

The R-Pentomino below is an example of a “Methuselah pattern”.



R-Pentomino

The R-Pentomino below is an example of a “Methuselah pattern”.



The terminology comes from the fact that it takes a long time before it settles into a steady-state (1103 generations).

History

Invented by John Horton Conway in late 1960s.

History

Invented by John Horton Conway in late 1960s.

Some fun things he is known for from his Wikipedia entry: Surreal numbers, Monstrous moonshine, Doomsday algorithm, Look-and-say sequence, Icosians, Free will theorem.

History, continued

- Initially the game was run by hand on tables with counter pieces and necklace shells in the University of Cambridge math common room.

History, continued

- Initially the game was run by hand on tables with counter pieces and necklace shells in the University of Cambridge math common room.
- Article published in Martin Gardner's *Scientific American* column about it in 1970.

History, continued

- Initially the game was run by hand on tables with counter pieces and necklace shells in the University of Cambridge math common room.
- Article published in Martin Gardner's *Scientific American* column about it in 1970.
- After that, it really took off, with people all over writing computer programs for it.

History, continued

- Initially the game was run by hand on tables with counter pieces and necklace shells in the University of Cambridge math common room.
- Article published in Martin Gardner's *Scientific American* column about it in 1970.
- After that, it really took off, with people all over writing computer programs for it.
- Various papers and were written, informal journals created, and 50 years later people are still discovering new things about it.

A one-line program for the Game of Life

From the Wikipedia page on the APL programming language

```
life←{↑1 ⍵∨.∧3 4=+/,¯1 0 1∘.⊖¯1 0 1∘.⌽⊂⍵}
```

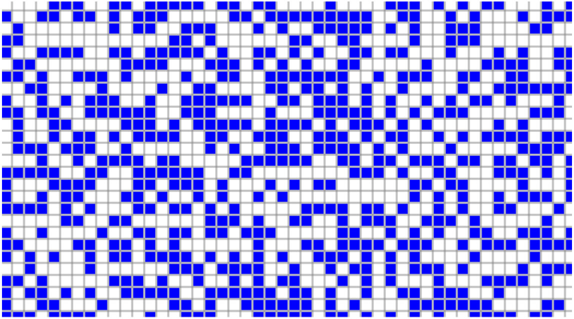
A one-line program for the Game of Life

From the Wikipedia page on the APL programming language

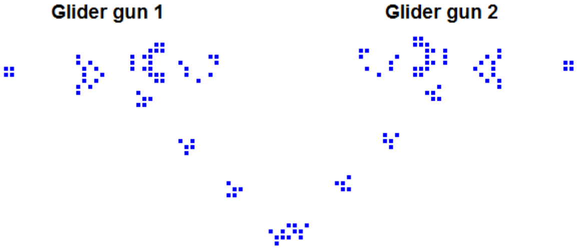
```
life←{↑1 ω∨.∧3 4=+/,¯1 0 1∘.⊖¯1 0 1∘.⌽⊂ω}
```

More reasonably, most Mount CS majors could probably write a Game of Life program in no more than a few hours in their favorite programming language.

It's fun to start with a random board and see what evolves.

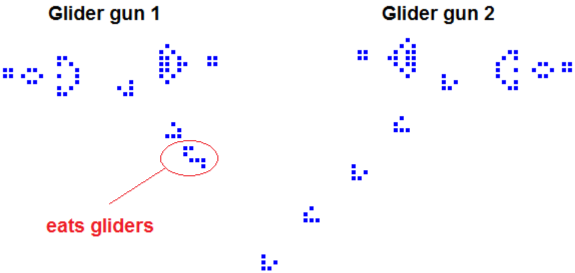


Putting glider guns together



If you put place glider guns in the right locations you can get interesting effects, like these gliders that cancel each other out.

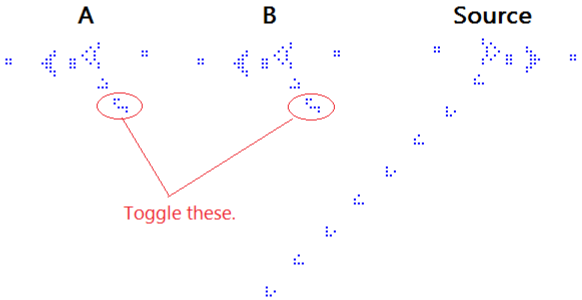
Blocking a glider gun



You can also block a glider gun.

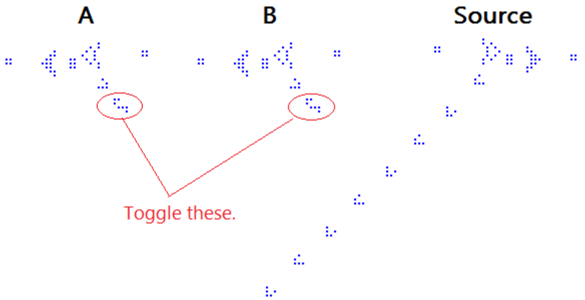
Logic gates

Putting these two ideas together, you can build logic gates, like this AND gate:



Logic gates

Putting these two ideas together, you can build logic gates, like this AND gate:



Removing either or both of the two blockers will cause the gliders from A or B to wipe out the glider stream from the source.

Creating a computer

- We can build logic gates.

Creating a computer

- We can build logic gates.
- We can also use these ideas to build things that remember state (memory).

Creating a computer

- We can build logic gates.
- We can also use these ideas to build things that remember state (memory).
- Put all this together, and we can build an actual working computer in Life:

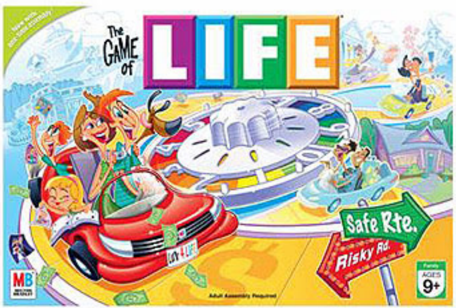
  `https://youtu.be/8unMqSp0bFY`

Life can solve anything a computer can

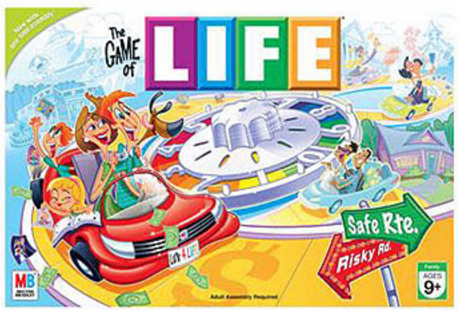
So in theory, Life can do anything that a computer can do

Life can solve anything a computer can

So in theory, Life can do anything that a computer can do(such as a program to play the Milton Bradley board game)



So in theory, Life can do anything that a computer can do(such as a program to play the Milton Bradley board game)



In computer science, people say that Life is *Turing Complete*.

Generalizations

- Fun to change around the rules for when cells turn on and off and see how things change.

Generalizations

- Fun to change around the rules for when cells turn on and off and see how things change.

- Life is an example of a *cellular automaton*.

Generalizations

- Fun to change around the rules for when cells turn on and off and see how things change.
- Life is an example of a *cellular automaton*.
- Cellular automata serve as models for physical phenomena like crystallization

Generalizations

- Fun to change around the rules for when cells turn on and off and see how things change.
- Life is an example of a *cellular automaton*.
- Cellular automata serve as models for physical phenomena like crystallization
- They are used in simulations of things like traffic flow, spread of a disease, spread of information, and much more.

More demonstrations

In whatever time we have left, let's look at some more demonstrations.

Image credits:
John Conway: By "Thane Plambeck" – https://www.flickr.com/photos/thane/20366806/, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=13076802

Life board game: enjoyingthegameoflife.blogspot.com