Game of Life

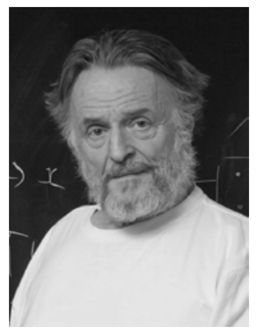
Alpaca - Jianing Duan, Haitong LIU, Haoyu LIU.

April 2020

Background

Cellular Automaton
- A model consists of a regular grid of *cells*
  - The grid can be in any finite number of dimensions
  - *Rule:* determines the new state of each cell in terms of the current state of the cell and its neighborhood
  - Select an initial state → Creat a new generation

Game of Life
- Devised by the British mathematician **John Horton Conway** in 1970
- A cellular automaton & a zero-player game



John Horton Conway
(1937-April 11,2020)

Rules

- The universe - infinite, two-dimensional orthogonal grid of square cells
- States of each cells - alive or dead
- Every cell - eight neighbors

Underpopulation

Any live cell with fewer than two live neighbors dies.

Survival

Any live cell with two or three live neighbors lives on to the next generation.

Overpopulation

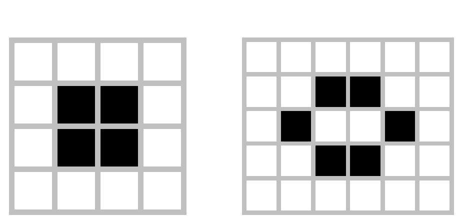
Any live cell with more than three live neighbors dies.

Reproduction

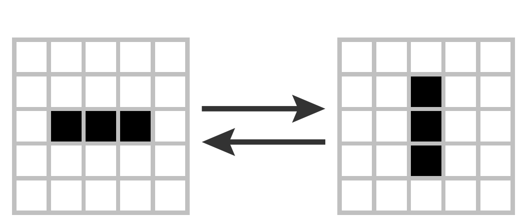
Any dead cell with exactly three live neighbors becomes a live cell.

Patterns

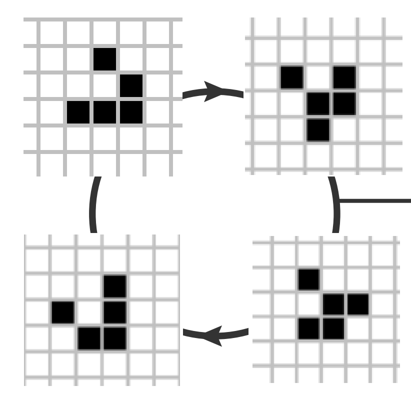
Three common pattern types in Game of Life
- *still lifes* — do not change from one generation to the next
- *oscillators* — return to their initial state after a finite number of generations
- *spaceships* — translate themselves across the grid



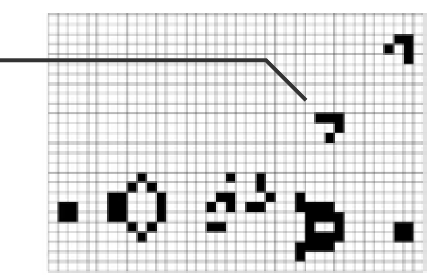
Block Beehive
Still Lifes



Blinker (period 2)
Oscillators



Glider
Spaceships



Gosper Glider Gun
(period 30)
Infinite Growth

- An online generator for Game of Life: https://funnyjs.com/jspages/game-of-life.html
- What will happen if we change the rules of GoL?

Turing Completeness of GoL

Turing machine
- a mathematical model of computation
- capable of simulating any computer algorithm's logic

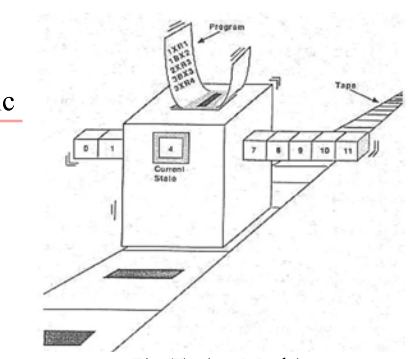


Fig. Turing Machine.

Two equivalent conditions for Turing Completeness:
- **1. Conditional branching**
- **2. The ability to change an arbitrary amount of memory**

For *Condition 1*, we try to make **logical gates** (Fig.1) in GoL.

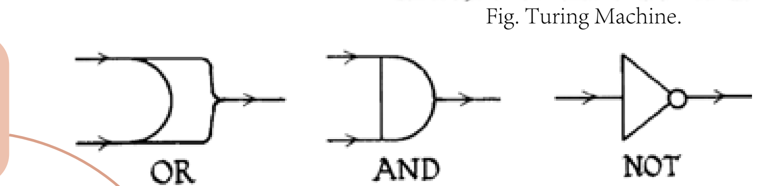


Figure 1. The Three Logical Gates.

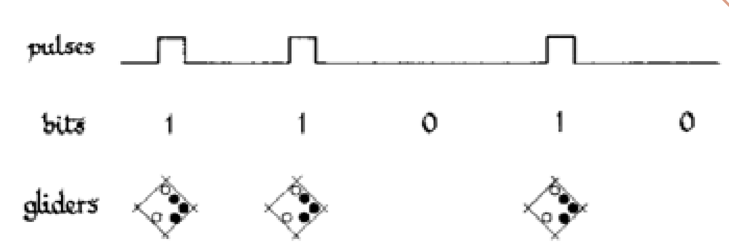
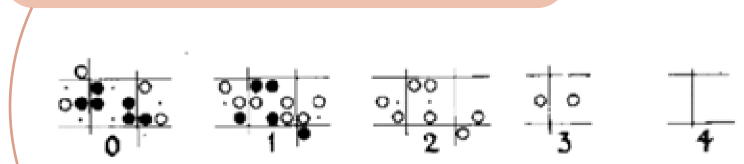
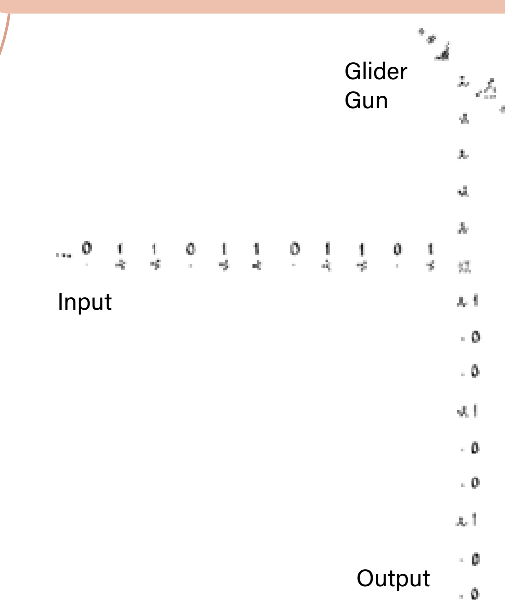


Figure 2. Gliding Pulses

Use Glider Guns as pulse generators, each glider represents 1 bit (Fig.2).

When two gliders meet in some particular ways, both of them will vanish.





With a vanishing collision and a Glider Gun, we can create a **NOT gate** (Fig. 3).

Figure 3. A Glider Gun and a Vanishing Reaction Make a NOT Gate.

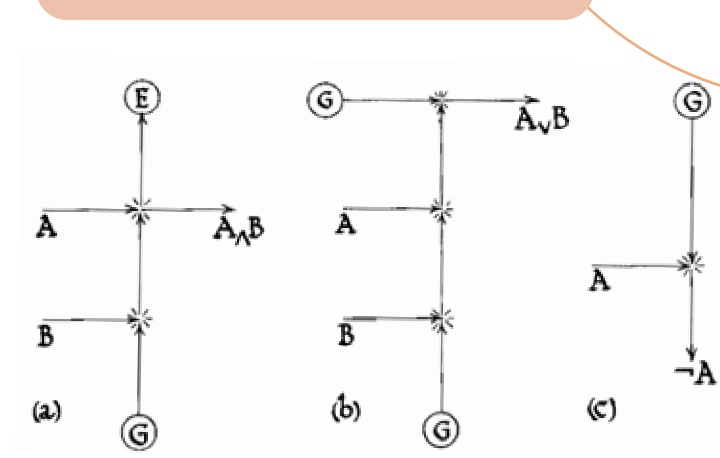


Figure 4. (a) An AND Gate. (b) An OR Gate. (c) A NOT Gate.

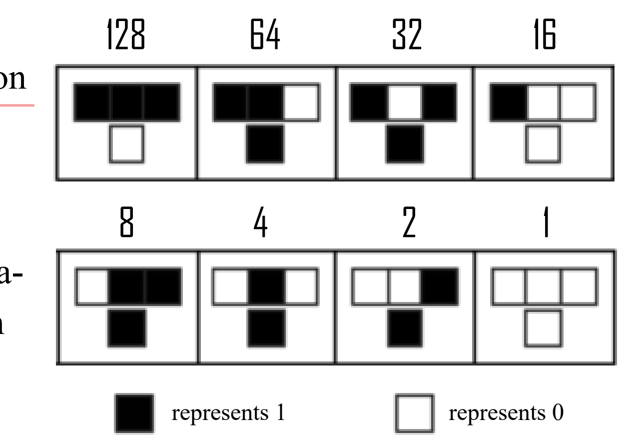
Similarly, we can also build the other two logical gates (Fig. 4).

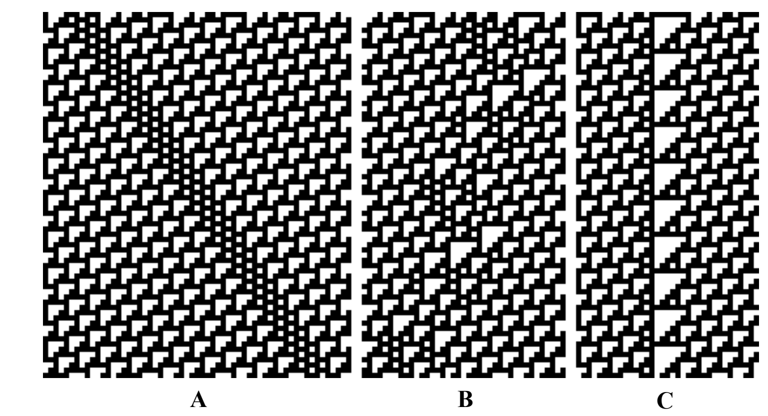
For *Condition 2*, we can use latches and flip-flops to read and write to storage memory.

- Want to see a Turing Machine in GoL? http://rendell-attic.org/gol/tm.htm

Another Cellular Automaton: Rule 110

- **Rule 110** - one-dimensional cellular automaton
- Two states of each cell - 0 or 1
- Each cell - two neighbors
- The name "Rule 110" - Its rule can be summarized in the binary sequence 01101110, which corresponds to the decimal value 110.

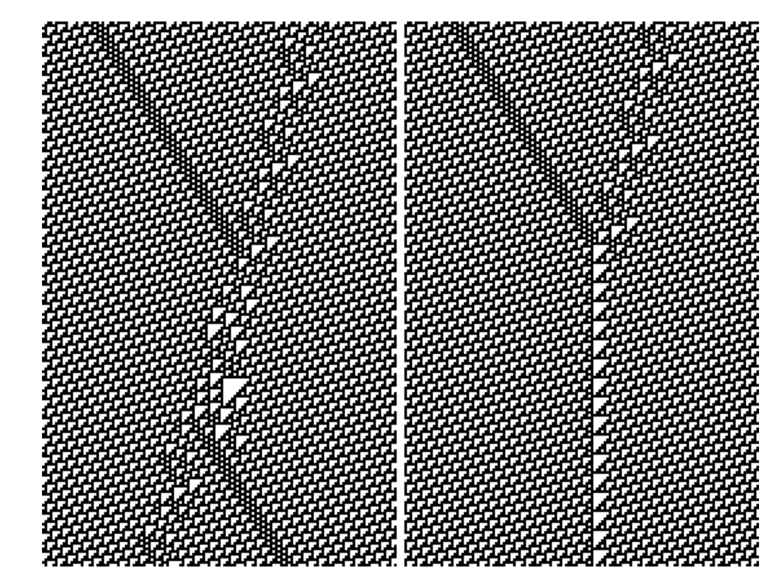




Spaceships in Rule 110

An infinitely repeating background: **00010011011111**

A: the sequence **0001110111**
B: the sequence **1001111**
C: the sequence **111**
— surrounded by the background



Collisions between Spaceships

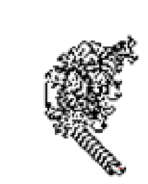
D: A and B passing through each other without interacting other than by translation

E: A and B interacting to form C

Turing-completeness of Rule 110

Rule 110 is also Turing-complete.
We can prove this by constructing a **cyclic tag system**, which is Turing-complete.

Other Turing-complete Examples

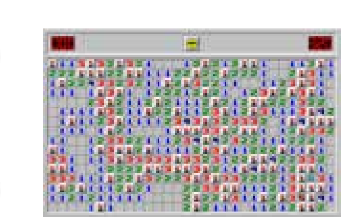


Langton's Ant

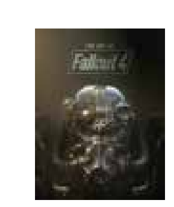
Minecraft

Super Mario Maker

Magic The Gathering



Minesweeper

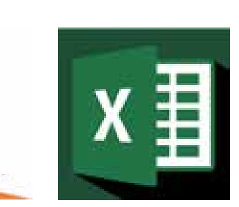


Fallout 4

Factorio

Cities: Skylines

Powerpoint



Excel

Reference

1. Berlekamp, E., Conway, J., & Guy, R.(1982) What is life (Chapter 25). *Winning Ways for Your Mathematical Plays*, vol. 4. London: Academic Press.
2. Cellular automaton. Retrieved from: https://en.wikipedia.org/wiki/Cellular_automaton
3. Conway's Game of Life. Retrieved from: https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
4. Turing machines. Retrieved from: https://en.wikipedia.org/wiki/Turing_machine#Universal_Turing_machines
5. Turing completeness. Retrieved from: https://en.wikipedia.org/wiki/Turing_completeness#Unintentional_Turing_completeness
6. Conway's Game of Life [gnuplot]. https://www.youtube.com/watch?v=k60nUT_IX6Y
7. Rule 110. Retrieved from https://en.wikipedia.org/wiki/Rule_110