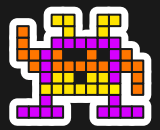
1 UP 25000

2 UP 003200

Conway's Game of Life

By universe 7

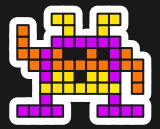
Rajan Sidhu,
Jason Whitlow,
Geremias Montano,
Eduardo Martinez



© UNIVERSE 7

Motivation:

- Cellular automata serve as a powerful computational model, with applications across various fields, such as biology, physics, and computer science.
- Understanding emergent complexity and self-organization is essential for studying real-world phenomena.



© UNIVERSE 7

Problem Statement:

- The objective of this project is to explore Conway's Game of Life, its rules, emergent properties, interesting objects, and potential applications. The study aims to gain insight into the automaton’s behavior, and contribute to the understanding of emergent complexity in cellular automata.

© UNIVERSE 7

Approach:

01

Choosing the correct libraries

- Numpy for the speed increase
- Pygame for the graphics
- Tkinter for the control panel

02

Set up the grid

We simply created a 2D array using numpy for our grid and set each cell to 0 for dead cells and 1 for live cells. We then created a function to check for neighbors.

03

Implementing the rules

After taking all cells and their neighbors into account, we create or destroy cells based on the classic Conway's game of life rules.

© UNIVERSE 7

Approach:

04

Graphics

For graphics we used Pygame since it’s a very easy graphics library to use. We also created a control panel to make testing the patterns easier.

05

Implemented a control panel

We also implemented a control panel for convenience when testing and adding new patterns.

06

Research and Creation

Finally we started researching different facts about the game of life and implemented a few patterns for your viewing.

© UNIVERSE 7

Rules of the Game

Classic Rules

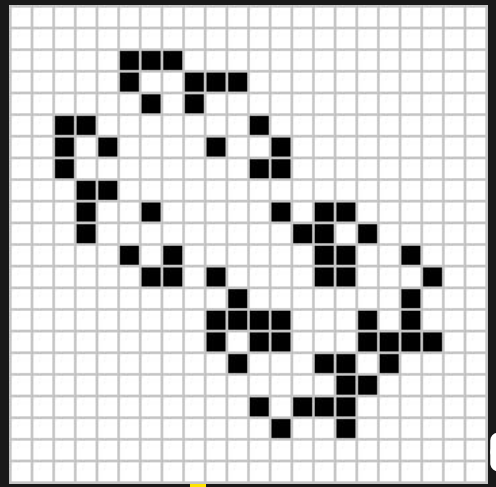
- A live cell with less than 2 neighbors dies.
- A live cell with more than 3 neighbors dies.
- A live cell with 2 or 3 neighbors lives.
- Dead cells with 3 live neighbors become alive.

© UNIVERSE 7

Research

- Still lifes: Patterns made in such a way that every live cell in the pattern has exactly enough neighbors to move on to the next generation, seeming to be completely motionless.
- Oscillators: Oscillators are patterns which tend to go through a cycle and will go on forever.
- Spaceships: Space ships are patterns which move across the screen, there are small spaceships like “gliders” and larger ones like the “Big Glider”

Big glider



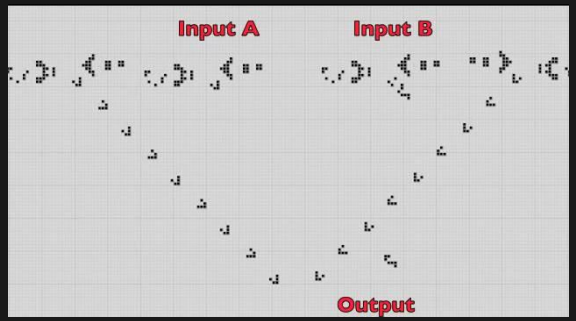
© UNIVERSE 7

Research

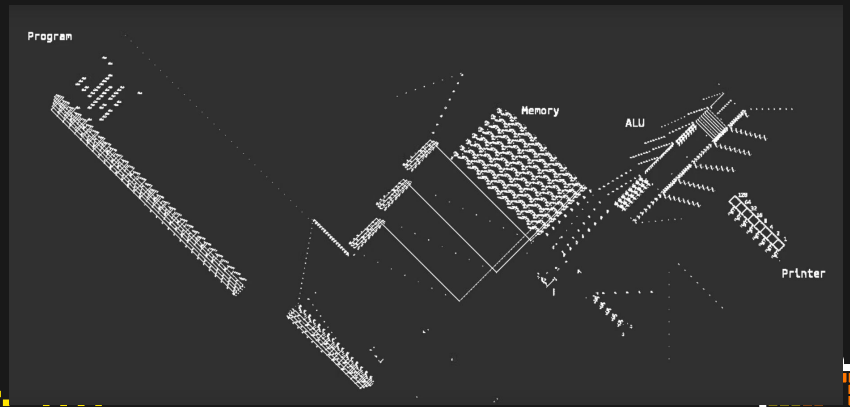
The game of life is turing complete!

Can replicate all logic gates
Can preform algorithms (such as computing the fibonacci sequence)

“AND” gate



Algorithm computing the fibonacci sequence



© UNIVERSE 7

Test!

Different Rules Used

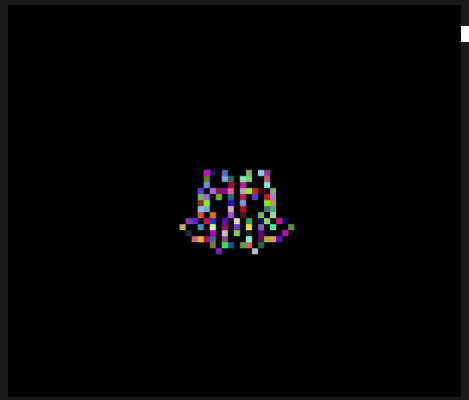
“Brain rule”:

- Live cells with less than 2 neighbors die
- Live cells with more than 4 neighbors die
- Dead cells with exactly 3 neighbors are born
- All other cells move on to the next generation

“Block rule”:

- Live cells with less than 7 neighbors die
- Live cells with more than 3 neighbors die
- Dead cells with one neighbor are born
- All other cells move on to the next generation

“Brain rule”



© UNIVERSE 7

Test!

Different Rules Used

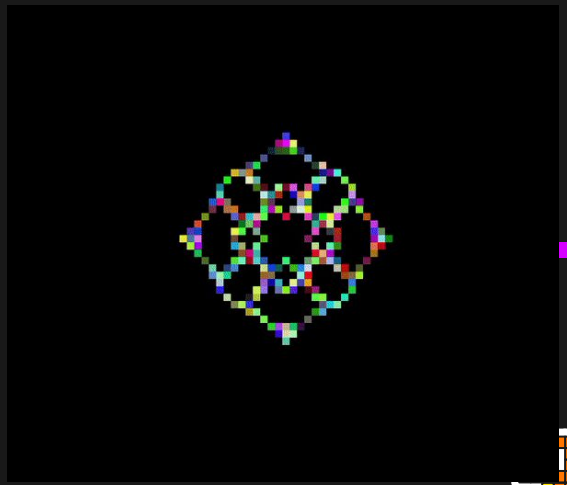
“Pattern rule”:

- Live cells with less than 2 neighbors die
- Live cells with more than 4 neighbors die
- Dead cells with 3 or more neighbors are born
- All other cells move on to the next generation

“Spark rule”:

- Live cells with less than 5 neighbors die
- Live cells with more than 4 neighbors die
- Dead cells with exactly 2 neighbors are born
- All other cells move on to the next generation

“Pattern rule”

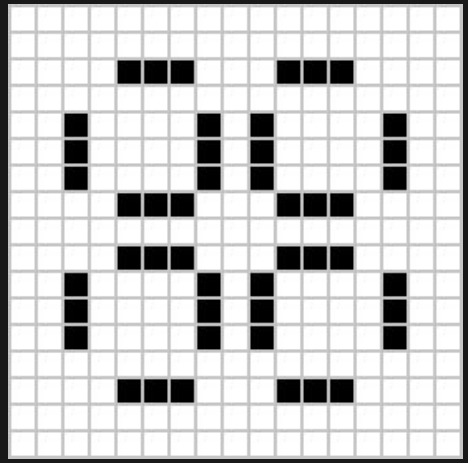


© UNIVERSE 7

Results:

- Experiments with different initial configurations show a wide range of emergent complexity, such as patterns forming, stabilizing, and interacting with other patterns.
- Interesting objects identified in the Game of Life include still lifes (e.g., Block, Beehive), oscillators (e.g., Blinker, Toad), spaceships (e.g., Glider, Lightweight Spaceship), and guns (e.g., Gosper Glider Gun).
- The Game of Life's emergent complexity and self-organization have potential applications in modeling natural systems, simulating computations, and studying artificial life.

Oscillator ex:



© UNIVERSE 7

Conclusion

- This project successfully explores the Game of Life by implementing it in Python, investigating its emergent complexity, interesting objects, and potential applications.
- The Game of Life and cellular automata serve as valuable tools for understanding complex phenomena and real-world applications.
- Future work could explore other cellular automata or extensions of the Game of Life, or investigate novel applications in interdisciplinary research.

THANKS!

DO YOU HAVE ANY QUESTIONS?