App Physics 157

COMPLEXITY SCIENCE
Cellular Automata
by Abdel Jalal D. Sinapilo

Objective
Cellular Automata

09 Week 3: Cellular automata

Learning tasks

By the end of this week, you must be able to:

1. Define a cellular automaton.
2. Demonstrate Wolfram's 1D CA models.
3. In the context of CA, discuss determinism, randomness and universality.
4. Implement CAs.

Background
Cellular Automata

A cellular automaton (CA) is a model of a world with very simple physics. "Cellular" means that the world is divided into discrete chunks, called cells. An "automaton" is a machine that performs computations - it could be a real machine, but more often the "machine" is a mathematical abstraction or a computer simulation. Cellular automatons are governed by rules that determine how the state of the cells changes over time.

CA is a collection of cells arranged in a grid of specified shape, such that each cell changes state as a function of time, according to a defined set of rules driven by the states of neighboring cells. CAs have been suggested for possible use in public key cryptography, as well as for applications in geography, anthropology, political science, sociology and physics, among others.

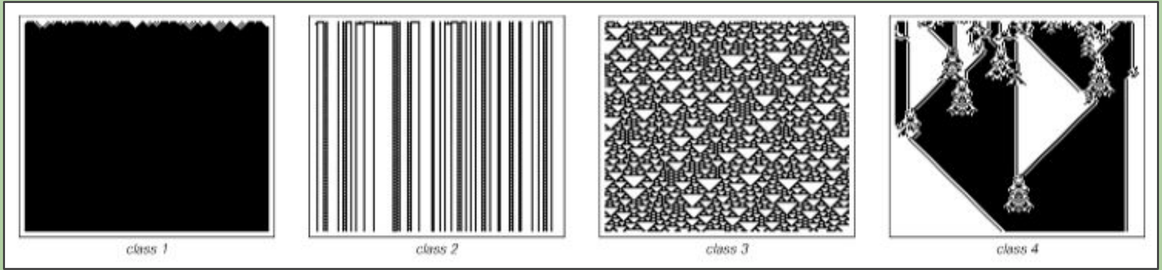
CELLULAR AUTOMATA

In class 1, the behavior is very simple, and almost all initial conditions lead to exactly the same uniform final state.

In class 2, there are many different possible final states, but all of them consist just of a certain set of simple structures that either remain the same forever or repeat every few steps.

In class 3, the behavior is more complicated, and seems in many respects random, although triangles and other small-scale structures are essentially always at some level seen.

Class 4 involves a mixture of order and randomness: localized structures are produced which on their own are fairly simple, but these structures move around and interact with each other in very complicated ways.

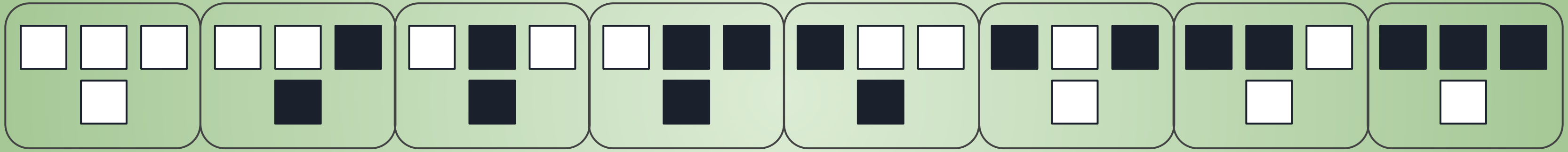


CELLULAR AUTOMATA

Elementary cellular automata is the simplest class of one-dimensional cellular automata. It has two possible values for each cell (0 or 1), and rules that depend only on nearest neighbor values. As a result, the evolution of an elementary cellular automaton can completely be described by a table specifying the state a given cell will have in the next generation based on the value of the cell to its left, the value the cell itself, and the value of the cell to its right, equal to 3 cells. Since there are $2 \times 2 \times 2 = 2^3 = 8$ possible binary states for the three cells neighboring a given cell, there are a total of $2^8 = 256$ elementary cellular automata, each of which can be indexed with an 8-bit binary number

ELEMENTARY CA

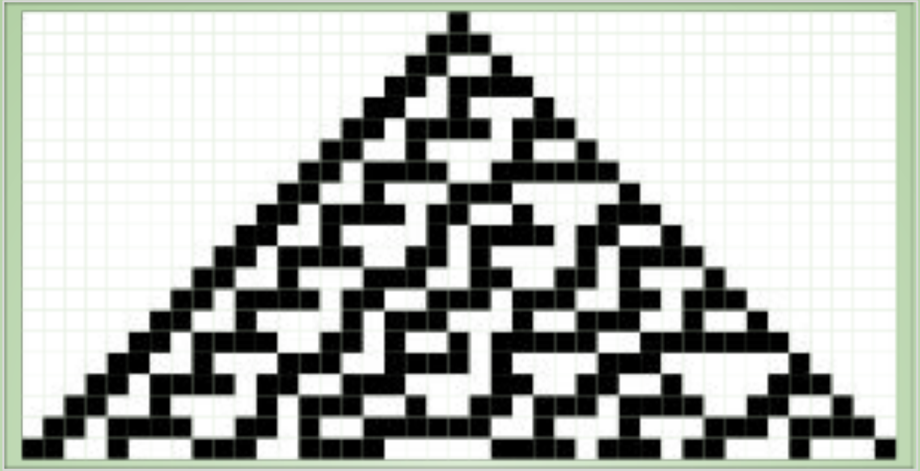
The diagram and table below provides the ruleset of rule 30, given by the binary number in the 2nd row of the table. In this ruleset, the possible values of the three neighboring cells are shown in the top row of each panel, and the resulting value the central cell takes in the next generation is shown below in the center.



0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	1	1	1	1	0	0	0

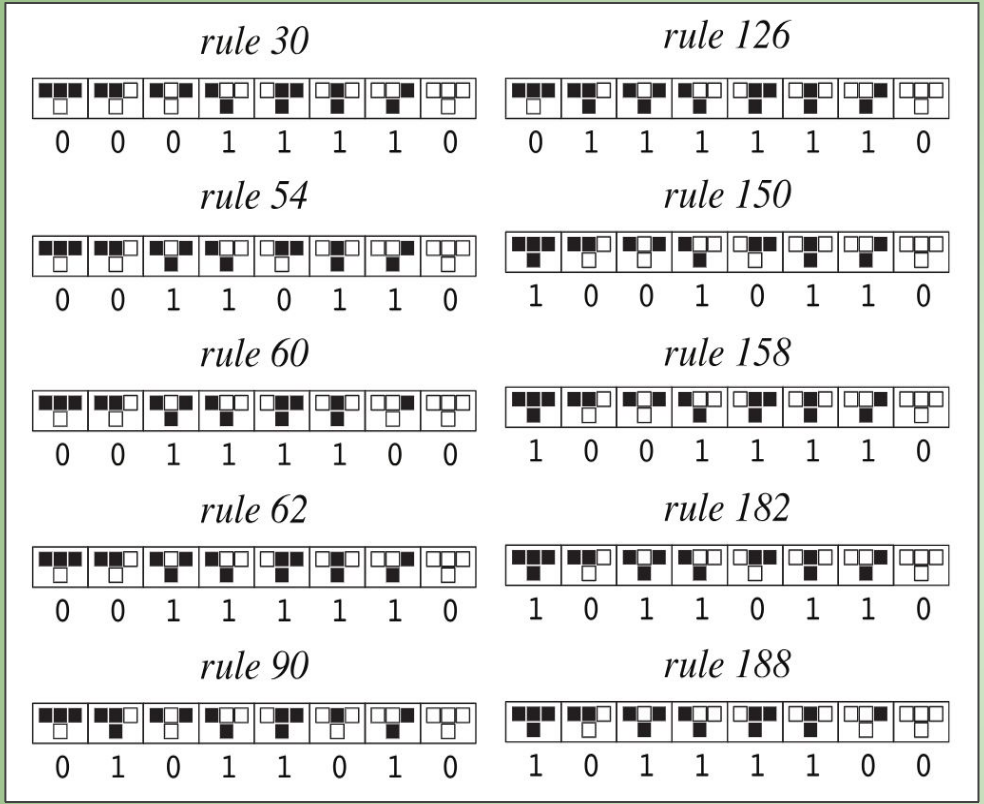
RULE 30

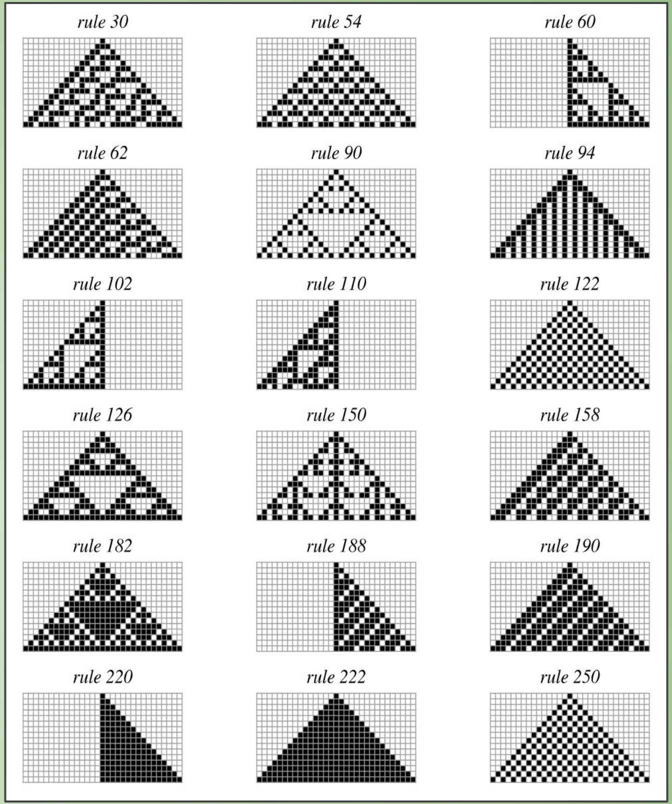
ELEMENTARY CA



The evolution of a one-dimensional cellular automaton can be illustrated by starting with the initial state (generation zero) in the first row, the first generation on the second row, and so on. For example, the figure above illustrated the first 20 generations of the rule 30 elementary cellular automaton starting with a single black cell.

ELEMENTARY CA





Famous CA rulesets and rules

ELEMENTARY CA

The Game of Life (an example of a cellular automaton) is played on a finite two-dimensional rectangular grid of cells. Each cell can be either alive or dead. The status of each cell changes each turn of the game (also called a generation) depending on the statuses of that cell's 8 neighbors. Neighbors of a cell are cells that touch that cell, either horizontal, vertical, or diagonal from that cell.

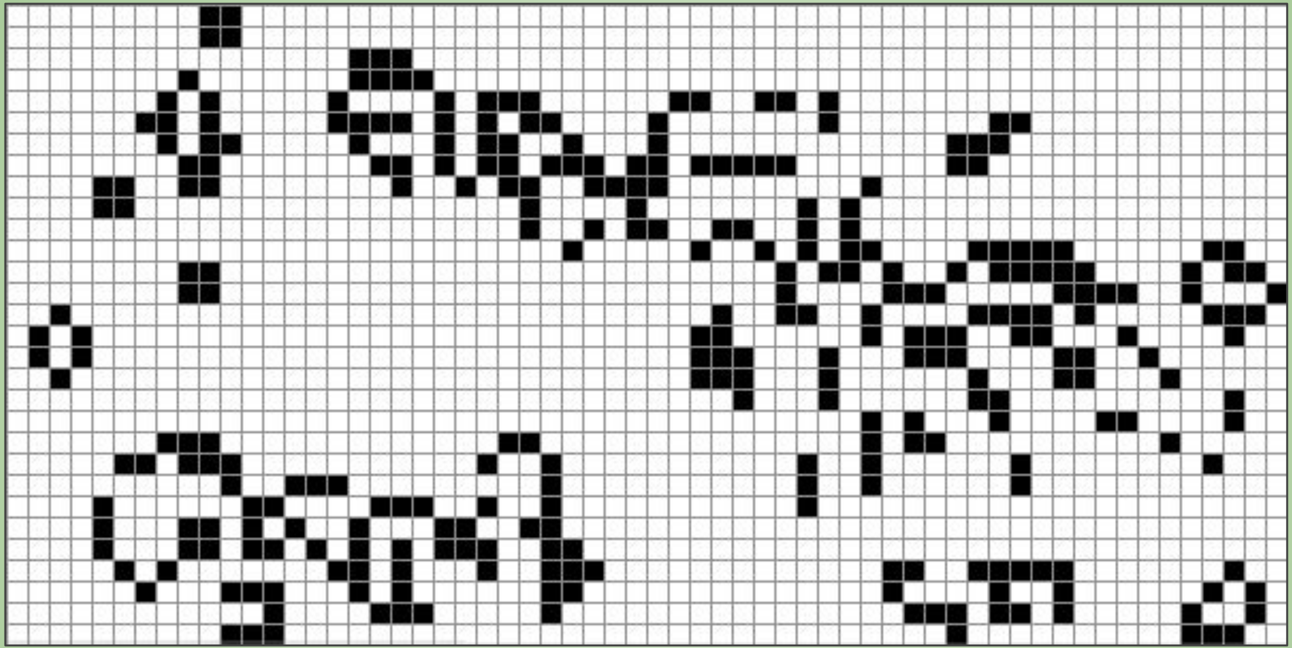
The initial pattern is the first generation. The second generation evolves from applying the rules simultaneously to every cell on the game board, i.e. births and deaths happen simultaneously. Afterwards, the rules are iteratively applied to create future generations. For each generation of the game, a cell's status in the next generation is determined by a set of rules. These simple rules are as follows:

Birth rule: An empty, or “dead,” cell with precisely three “live” neighbors (full cells) becomes live.

Death rule: A live cell with zero or one neighbors dies of isolation; a live cell with four or more neighbors dies of overcrowding.

Survival rule: A live cell with two or three neighbors remains alive.

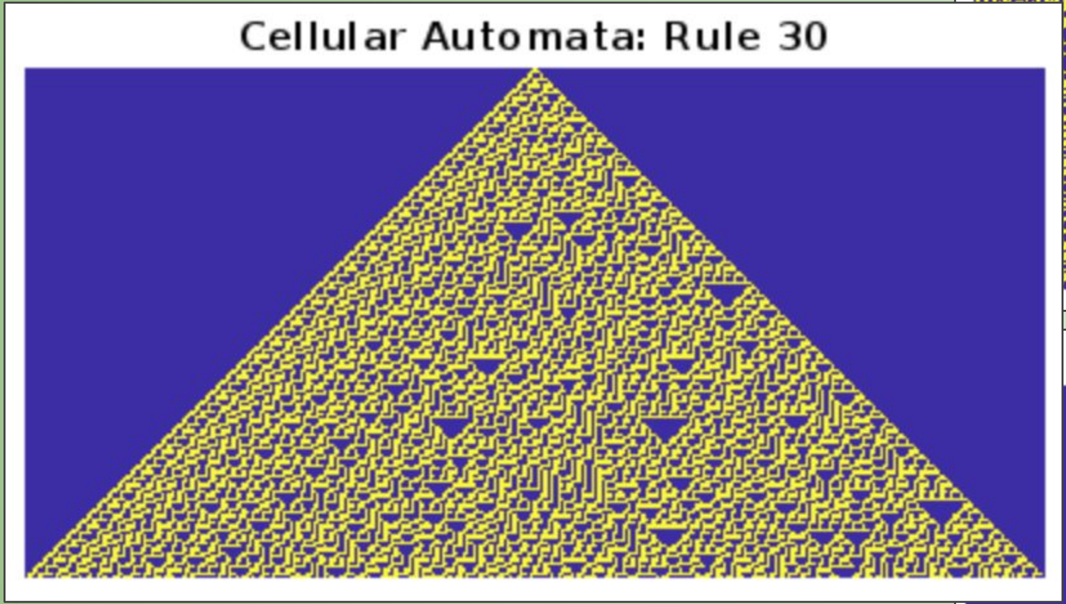
GAME OF LIFE

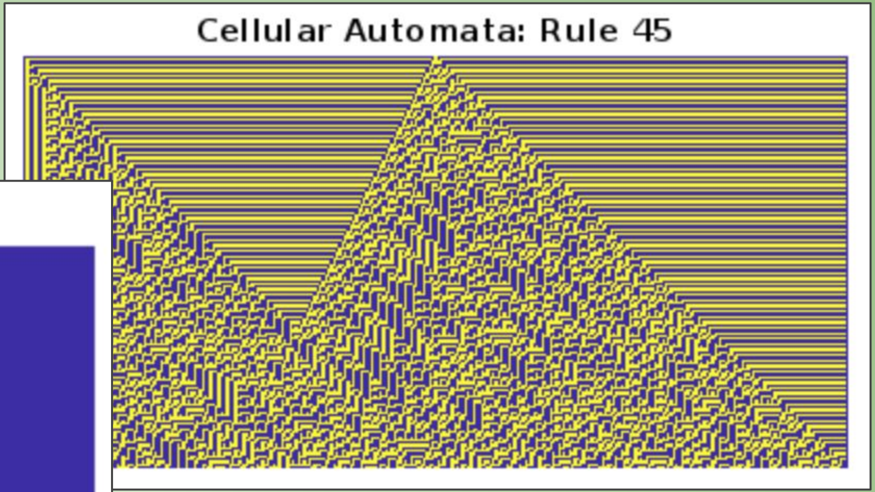


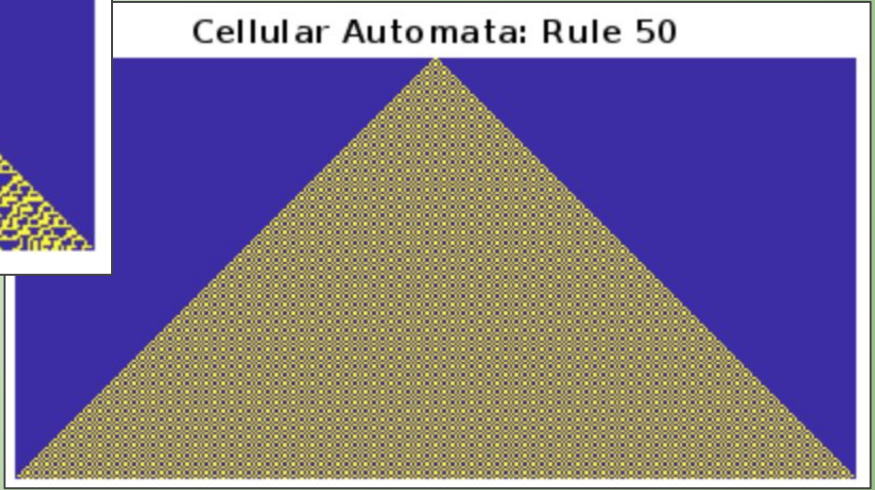
Game of Life at an arbitrary timestep

GAME OF LIFE

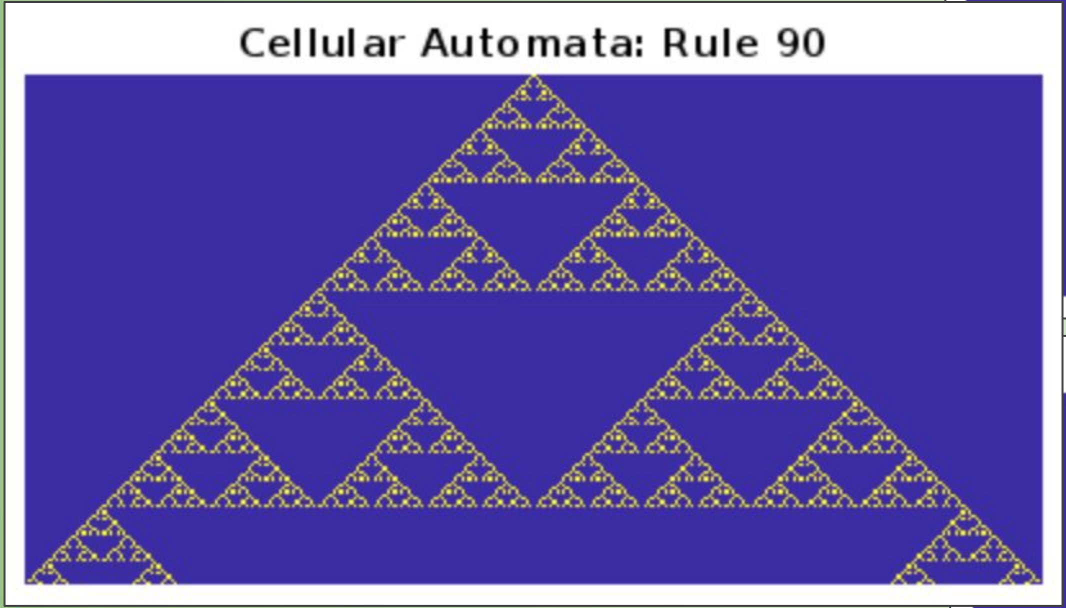
Results

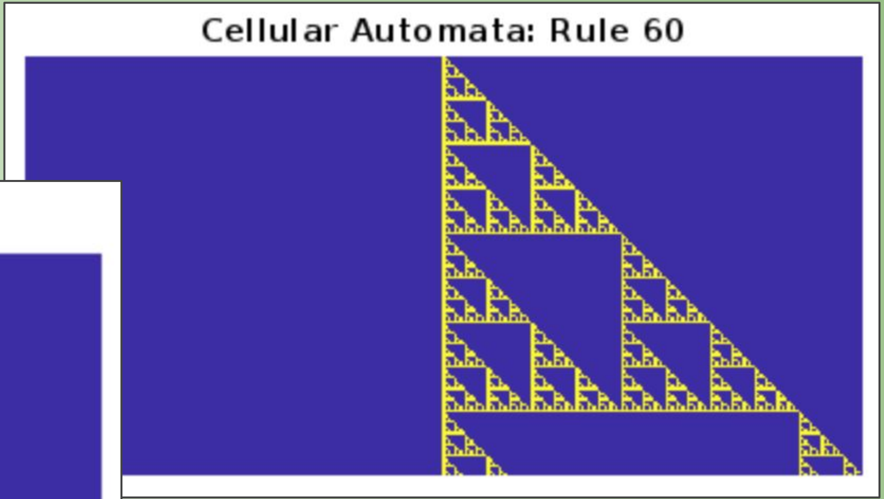


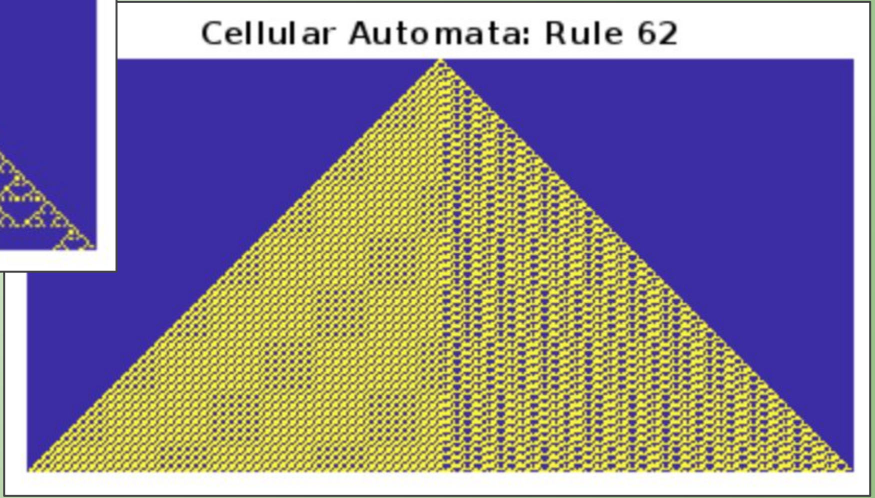




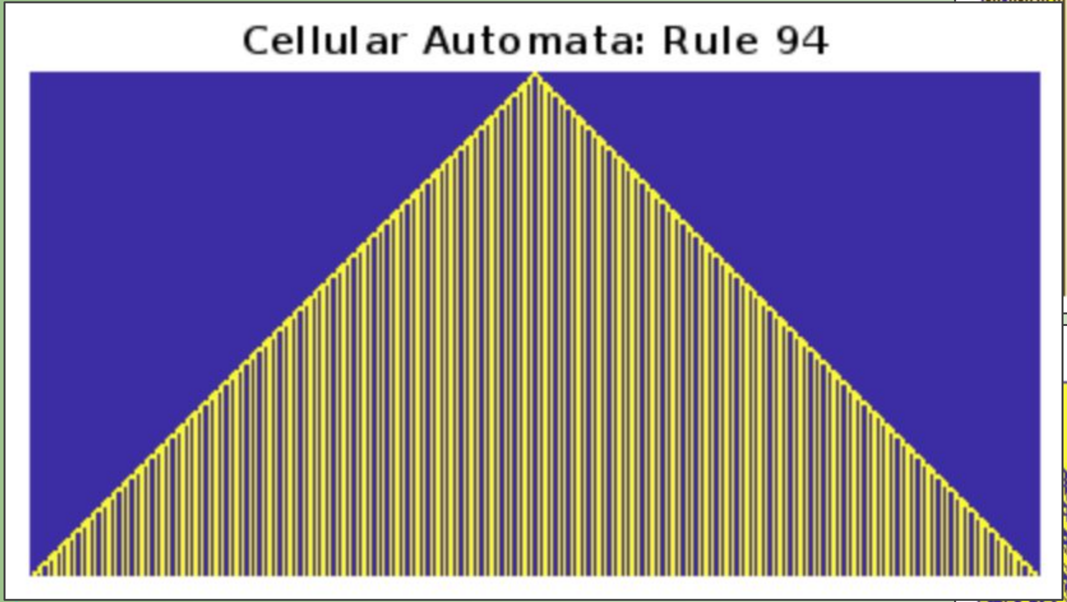
ELEMENTARY CA
(150 Generations)

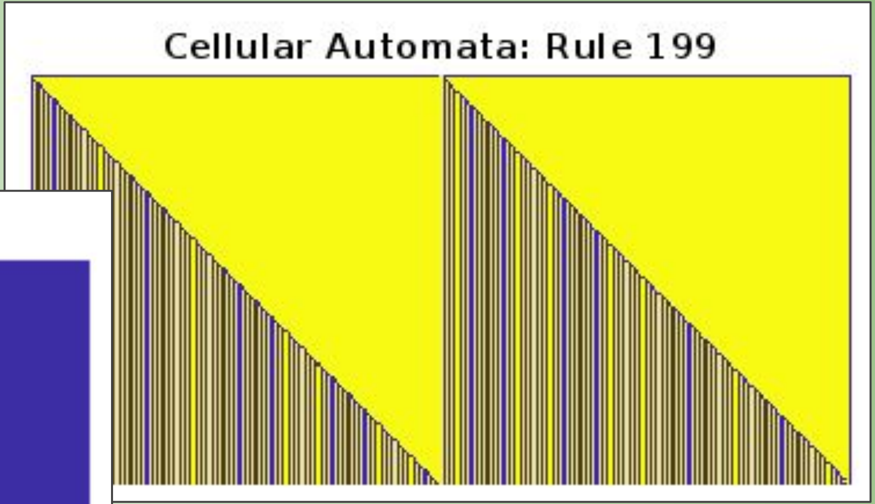




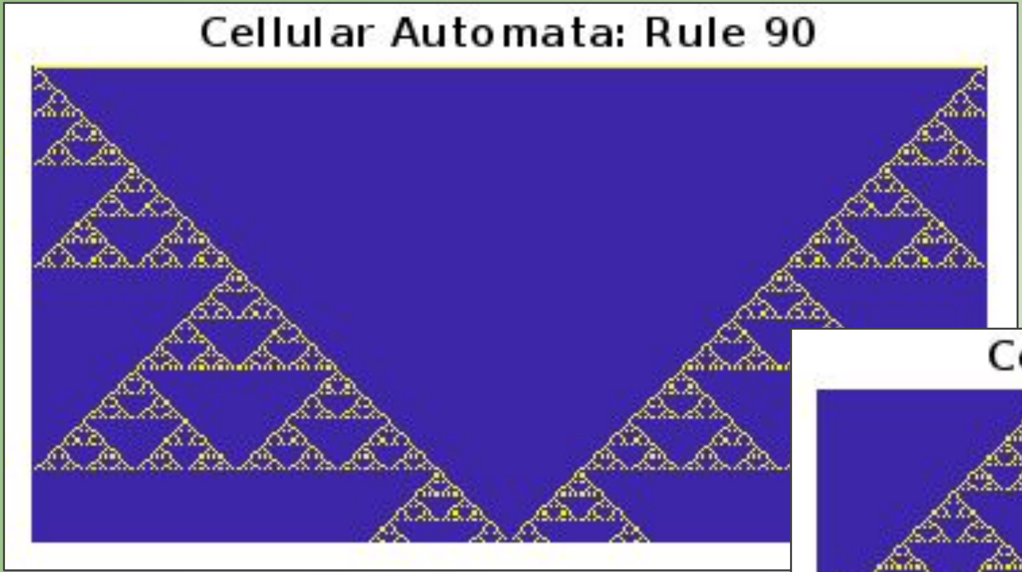


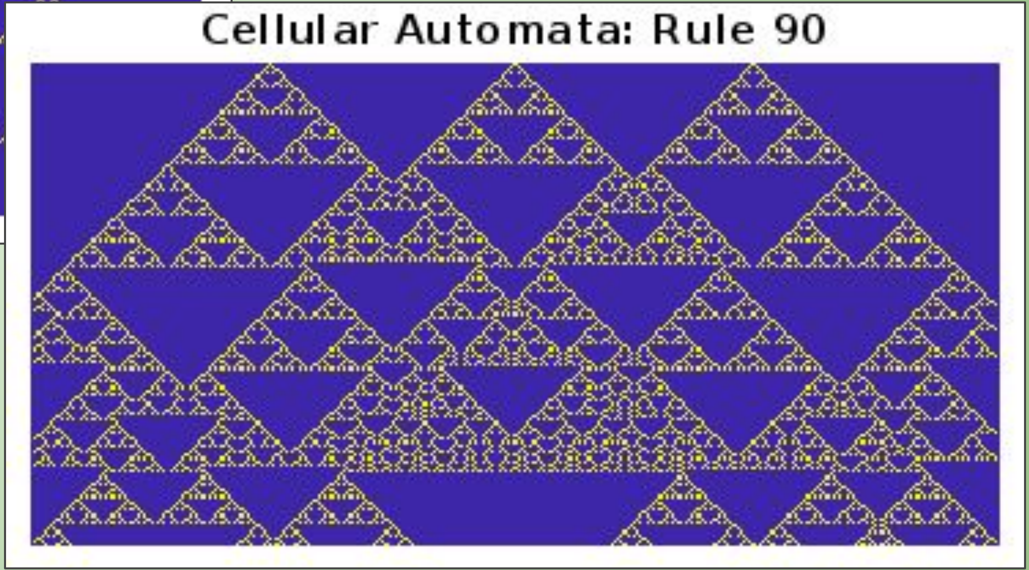
ELEMENTARY CA
(150 Generations)





ELEMENTARY CA
(150 Generations)



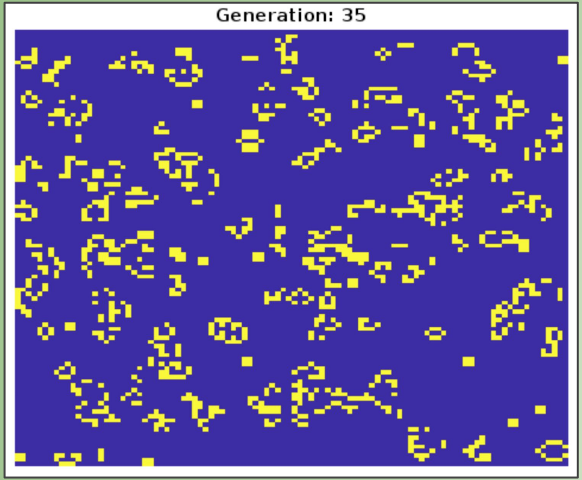


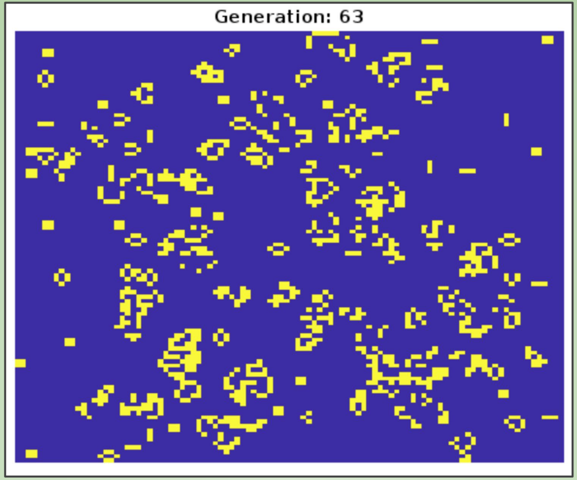
ELEMENTARY CA
(150 Generations - Varying Initial State)

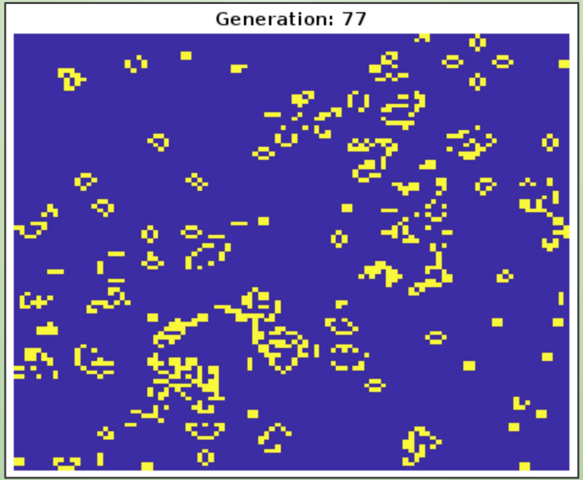
ECA rules can produce a wide range of interesting patterns and behaviors, ranging from highly ordered and predictable such as Rule 30 and Rule 90, to chaotically random such as Rule 45 and Rule 225 . The behavior of an ECA rule depends on its underlying rule number and the initial configuration of the cells. As can be seen from the previous slide, even though they are applying the same rule of Rule 90, since they have different initial states, they exhibit different behaviors.

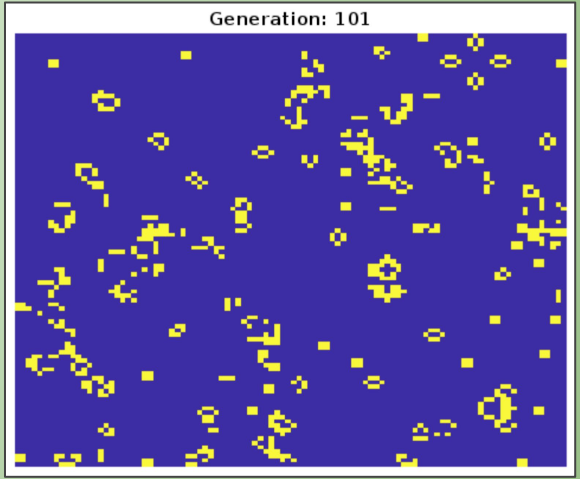
Some rules are exemplary as they display interesting characteristics such as Rule 30 and Rule 90. Rule 30 possesses interesting properties related to randomness and universality, making it a pet topic among researchers on computation theory, cryptography, and theoretical physics. Rule 90 on the other hand, displays the mathematically famous Sierpinski triangle, which is a fractal based on a triangle with four equal triangles inscribed in it, where the central triangle is removed and each of the other three treated as the original was, and so on, creating an infinite regression in a finite space

ANALYSIS

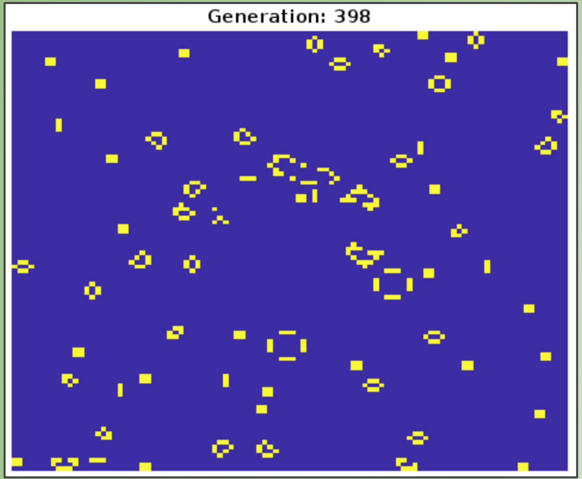


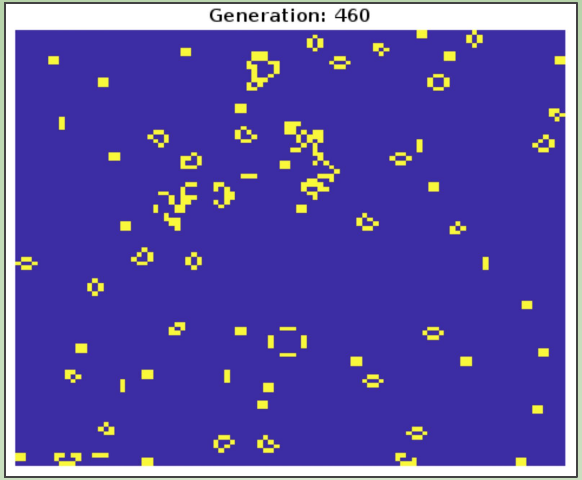


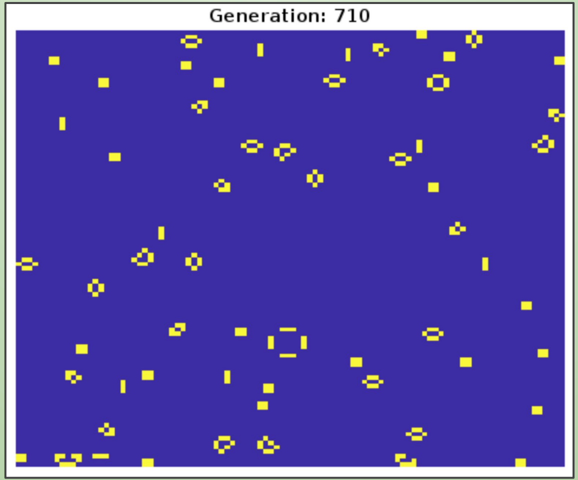


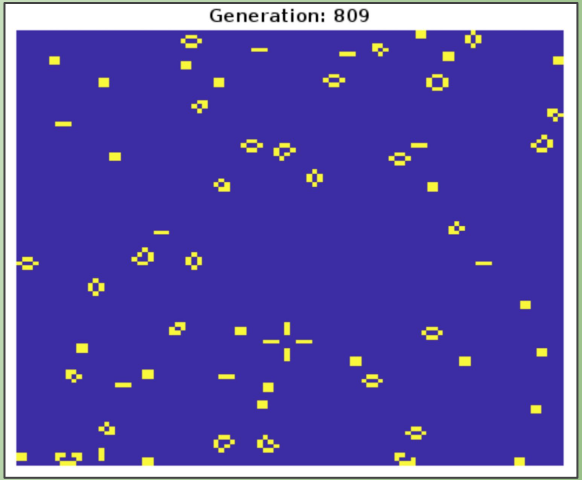


GAME OF LIFE
(Random Initial Condition)

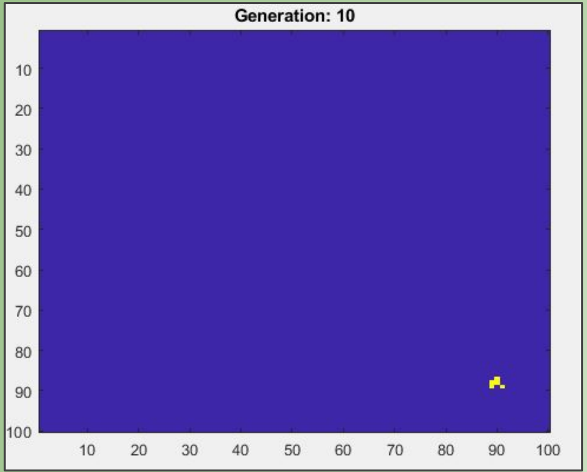


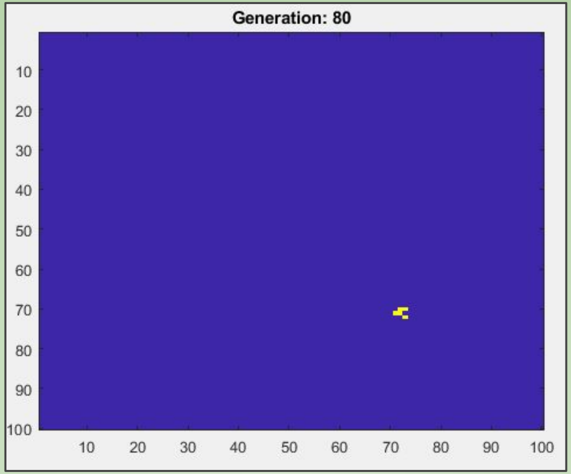


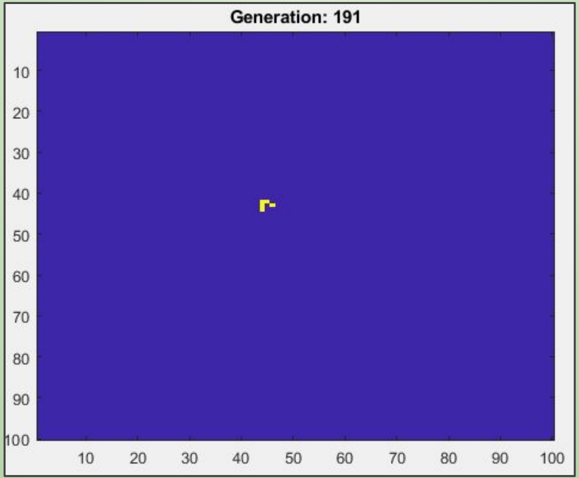




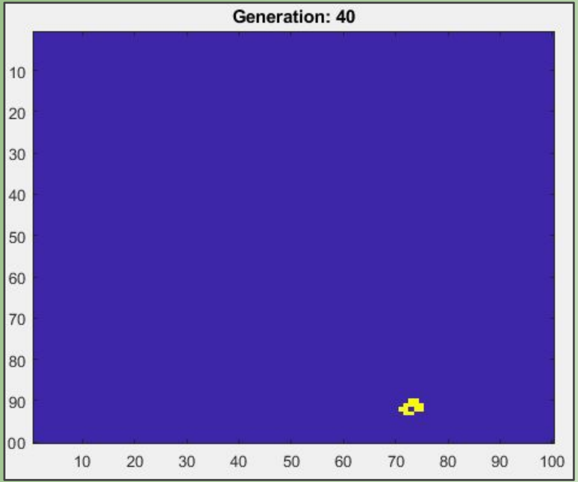
GAME OF LIFE
(Random Initial Condition)

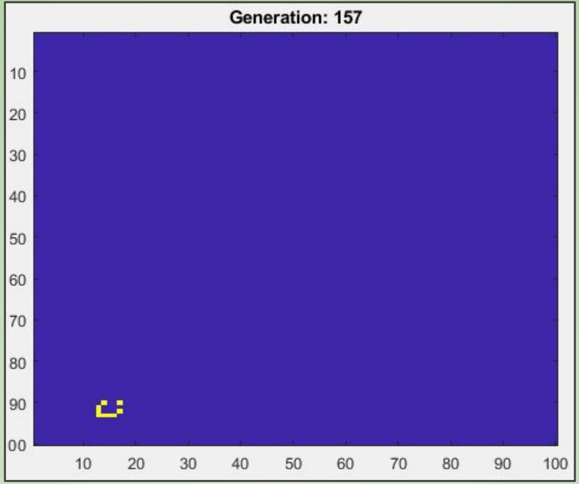


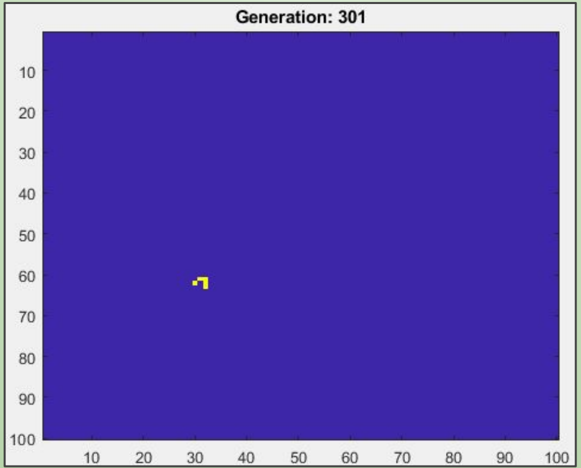


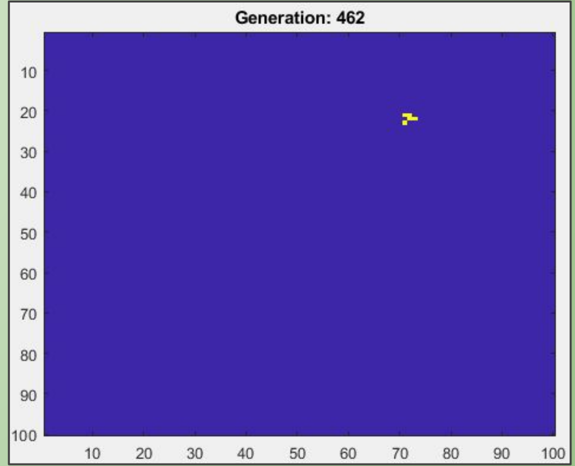


GAME OF LIFE (Glider)

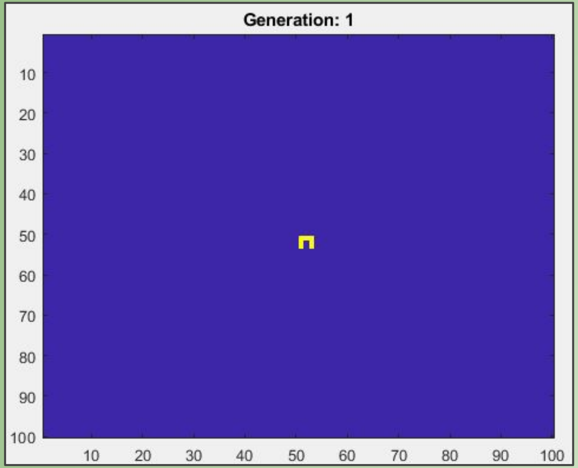


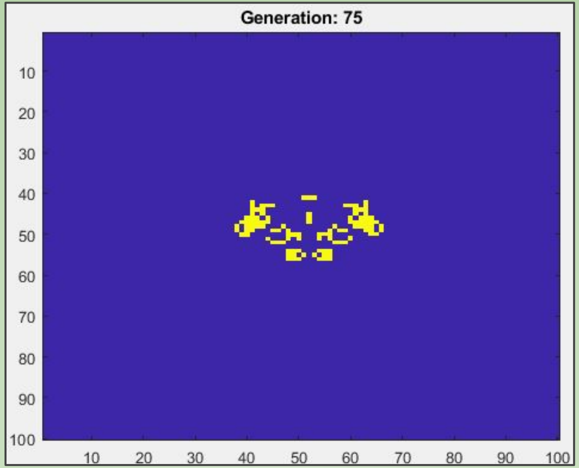


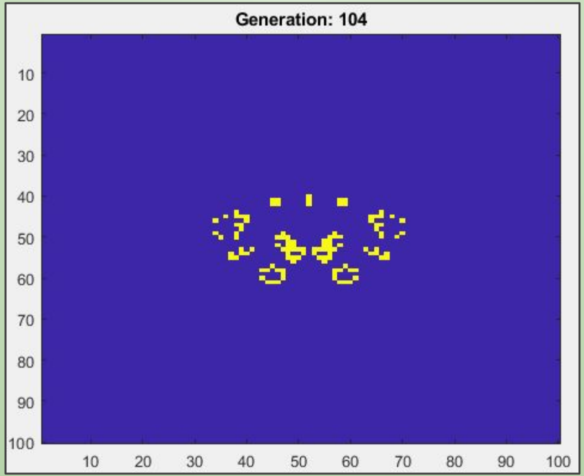


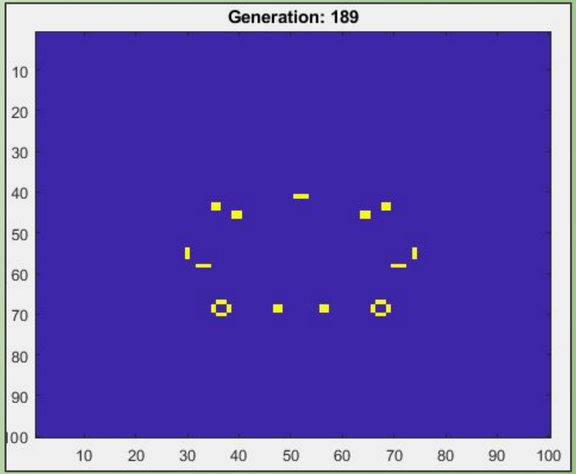


GAME OF LIFE (Lightweight Spaceship)

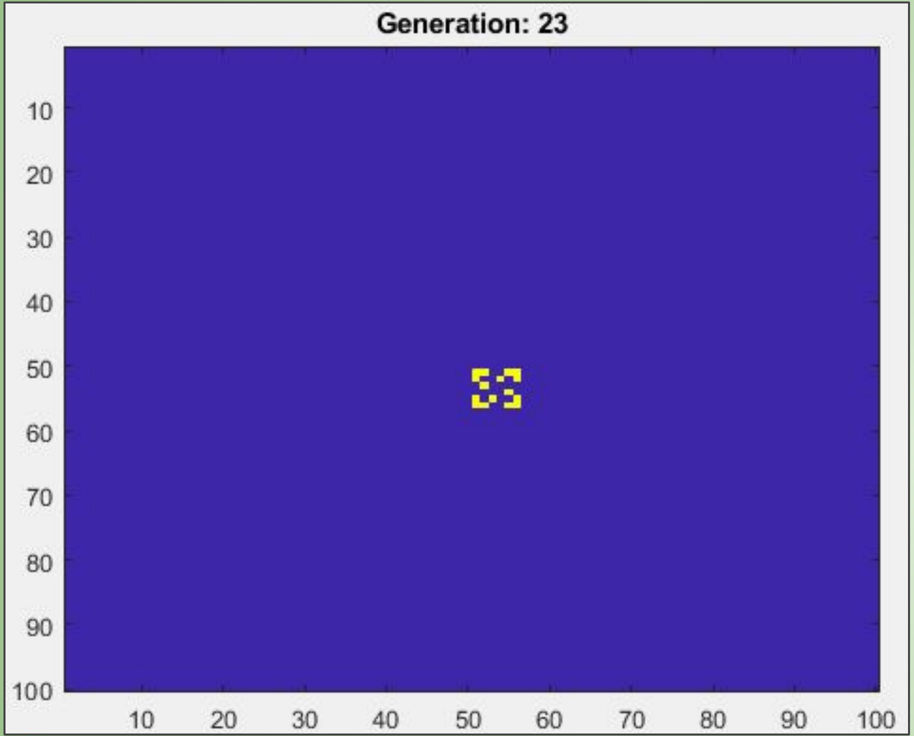


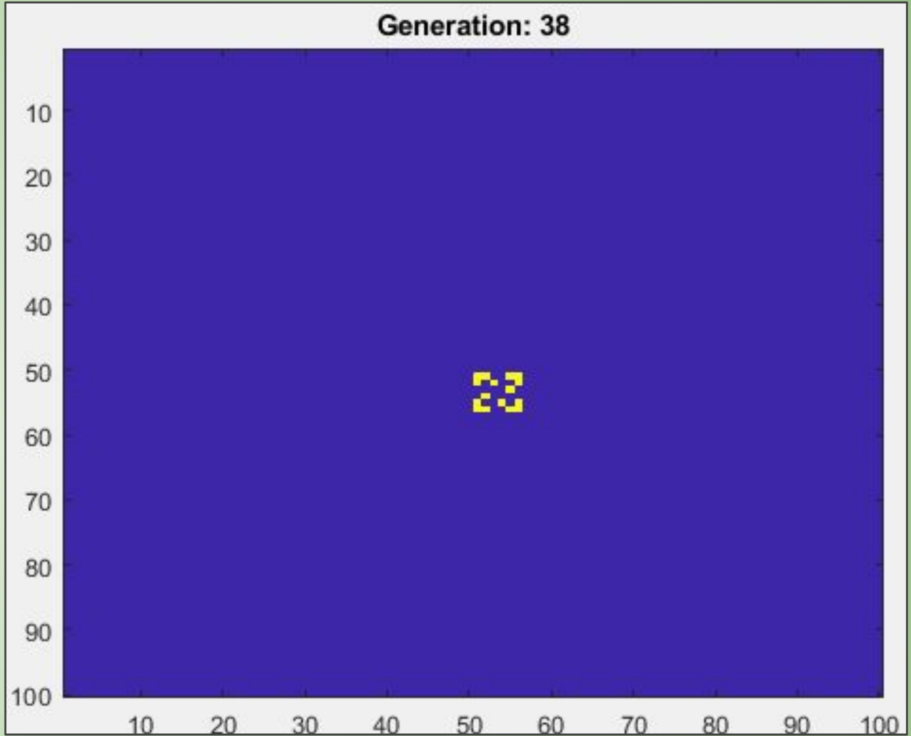






GAME OF LIFE
(Pi Heptomino)





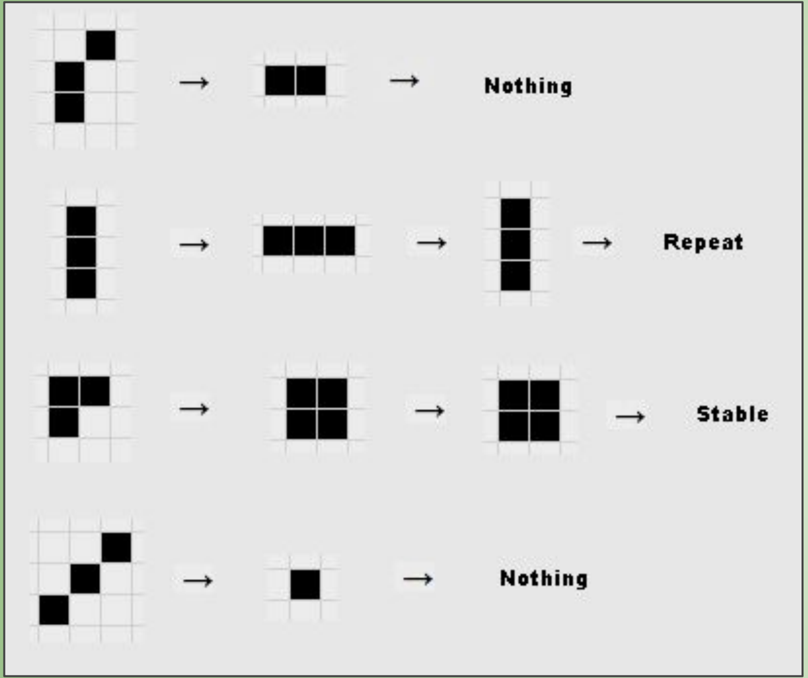
GAME OF LIFE
(Quad)

One of the most intriguing aspects of the Game of Life is its ability to exhibit a wide range of behaviors, from simple oscillations and static structures to complex, self-replicating organisms and shapes. Some patterns in the Game of Life with random initial conditions are stable and persist indefinitely, while others may grow, die out, or interact with each other in interesting ways.

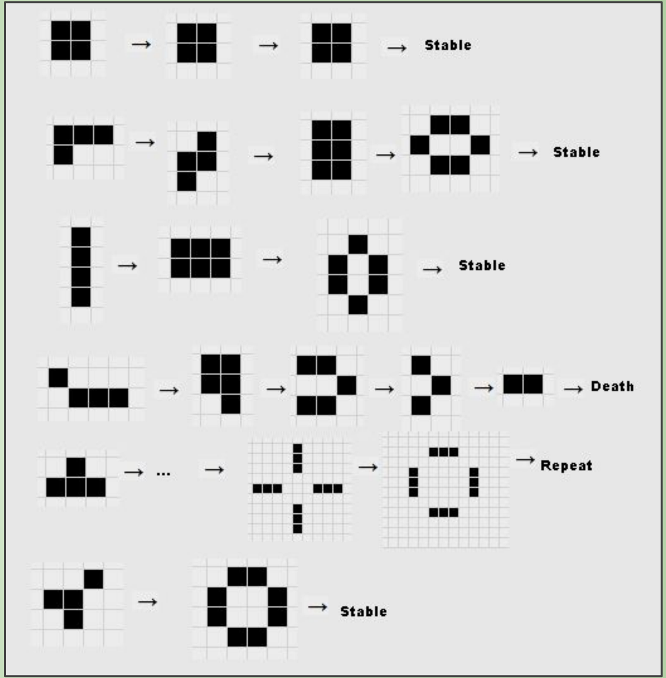
Some initial organisms exhibit exceptional and exclusive behaviors. The Glider is noticed to move towards the upper left direction, no matter where in the grid it started. The Lightweight Spaceship is seen to bounce on the walls of the grid. The Pi Heptomino emits two gliders in opposite directions, resulting in a stable configuration where the pi heptomino disappears, and only the two gliders continue their motion. Lastly, the Quad shows an binary oscillation of the 4 cells in the middle of the region.

ANALYSIS

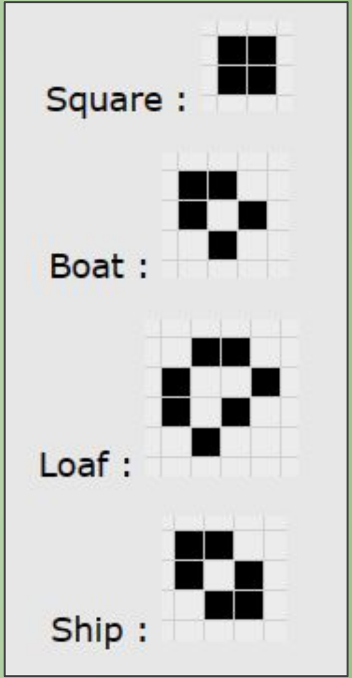
Here are some notable organisms and their evolution in GoL:



Triomino Patterns



Tetromino Patterns

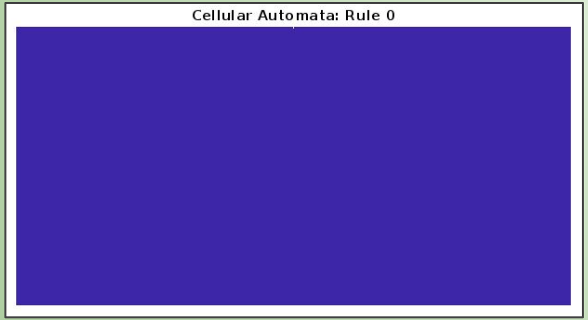


Still Life

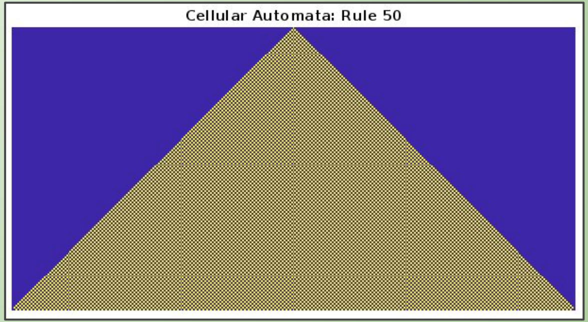
ANALYSIS

Determinism, Randomness, & Universality

Cellular automata are inherently deterministic systems. The future states of the cells are completely determined by the current states and the fixed rules of the automaton, such that given the same initial configuration, the same rule, and the same neighborhood scheme, the cellular automaton will always produce the same evolution pattern. The behavior of each cell is entirely predictable, and there is no element of randomness involved in the evolution of the system. These are best displayed by Class 1 and 2 CAs.



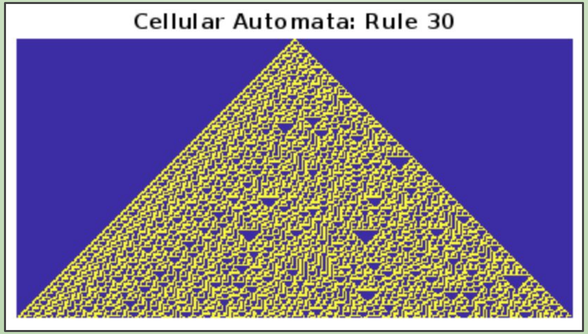
Class 1 CA



Class 2 CA

DETERMINISM

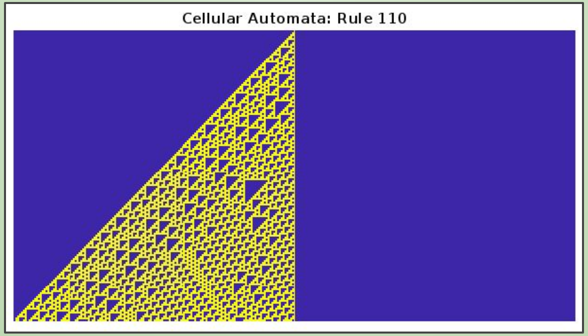
However, while cellular automata are deterministic, they can exhibit complex and seemingly random behavior under certain conditions. This behavior emerges from the interactions and patterns formed by the simple local rules governing the cells. Even though the system is entirely predictable, the overall evolution can appear random and chaotic due to the intricate interactions between neighboring cells, comparable to modern Pseudo Random Number Generators (PRNGs). Class 3 CAs demonstrate randomness.



Class 3 CA

RANDOMNESS

Universality refers to the ability of a particular automaton to simulate any other computational system. If a cellular automaton is universal, it means that it can be used to represent any arbitrary computation, given an appropriate configuration and rule setup. It can emulate any computation performed by a Turing machine, indicating that it possesses the computational power to simulate any other computational system. This is best exhibited by Class 4 CAs.



Class 4 CA

UNIVERSALITY

Reflection

This activity was a highly insightful introduction to complexity science. Cellular automata provide a simple yet powerful framework to study and understand systems that display complexity. From the interactions of simple local rules, cellular automata can exhibit emergent behaviors and patterns that are not explicitly encoded in the rules themselves, which are very interesting. Moreover, Conway's Game of Life is such a complicated and colorful topic on its own, even though it is just a subtopic of CA.

The only challenge I had to face in the activity was the coding, although it was not a necessarily difficult challenge, as the theory and process of CA are logical, with properties that are based in conditions and grids, which makes it straightforward to implement in code, using conditional statements and matrices operated in loops.

Self Grade

CRITERIA	QUALIFICATIONS	SCORE
Technical Correctness	❑ Met all objectives ❑ Results are complete ❑ Results are verifiably correct ❑ Understood the lesson	35
Presentation Quality	❑ All text and images are good quality ❑ Code has sufficient comments/guides ❑ Plots are properly labeled and visually understandable ❑ Report is clear	35
Self Reflection	❑ Explained validity of results ❑ Discussed what went wrong/right in activity ❑ Justified self score ❑ Acknowledged sources	30
Initiative	❑ Experimented beyond what was required ❑ Made significant improvements to existing code	10

References

[1] Weisstein, Eric. (n.d.). "Elementary Cellular Automaton." From MathWorld--A Wolfram Web Resource. Retrieved from https://mathworld.wolfram.com/ElementaryCellularAutomaton.html

[2] Downey, A. (2016). *Think Complexity.* Green Tea Press.

[3] Awati, R. (n.d.). Cellular automaton (CA). *TechTarget.* Retrieved from https://www.techtarget.com/searchenterprisedesktop/definition/cellular-automaton

[4] Lipa, C. (n.d.). Conway's game of life. *Chaos and Fractals.* Retrieved from https://pi.math.cornell.edu/~lipa/mec/lesson6.html

[5] Roberts, S. (2020, December 8). The lasting lessons of John Conway's game of life. *The New York Times.* Retrieved from https://www.nytimes.com/2020/12/28/science/math-conway-game-of-life.html

[6] Jose, K.M. (2013, January 13). Up for a game of life, anyone?. Retrieved from https://kevinmartinjose.com/2013/01/13/up-for-a-game-of-life-anyone/

[7] Callahan, P. (2001, February 11). Patterns, programs, and links for Conway's game of life. Retrieved from http://www.radicaleye.com/lifepage/picgloss/picgloss.html

[8] Wolfram, S. (2002). A New Kind of Science.