Analyzing Rules, Properties, and Objects Created in Conway's Game of Life

Abdulaziz Al-Said, Kalvin Xiong, Weston Hawes, Anthony Lopez
California State University, Fresno
April 14, 2023

Abstract

The purpose of this report is to introduce and analyze Conway's Game of Life as well as going over the approach that was taken to implement Conway's Game of Life in a scripting language. This report will breakdown the game's rules, interesting properties, and interesting objects that can be observed. All objects mentioned in this report were plotted and ran on our implementation of Conway's Game of Life.

1 Introduction

Our goal with this project was to observe how complex behavior can emerge from simple rules using cellular automata. The challenge was to implement Conway's Game of Life in a scripting language and then to analyze the rules followed, interesting properties, and interesting objects. All information presented in this report was already studied extensively and is not new information. Thus, the goal of this report is not to contribute new information to this field of study, but to instead analyze the information that has already been found and present it in an easily understandable way. This is important due to this report explaining our approach to implementing Conway's Game of Life in Python.

2 Background Material

2.1 Transition Rules

In Conway's Game of Life, there are 4 transition rules that govern each change of state through each iteration. The rules are as follows: each cell with one or no neighbors dies, as if by solitude, each cell with four or more neighbors dies, as if by overpopulation, each cell with two or three neighbors survives, and for a space that is empty or unpopulated, each cell with three neighbors becomes populated. Each cell is placed in a 2D matrix, and the transition rules are evaluated by the neighboring cells. This would be an example of a 2D cellular automaton.

2.2 Interesting Properties

Some interesting properties in Conway's Game of Life are as follows: this is a zero-player game, meaning it is deterministic and the evolutions are determined by the initial state, it can function as a Turing machine, meaning any finite set of solutions with a finite set of transition functions can be simulated using the Game of Life, and it is also undecidable, meaning that there is not an algorithm that exists that can tell if a specific pattern will ever appear in later iterations with a given initial state.

2.3 Interesting Objects

Some interesting objects that can be create are classified as sill objects, oscillating objects, and traveling objects. Still objects are ones that do not move at all in every following evolution. There are various initial states that fall under this category. Oscillating objects are ones that stay confined between a few points in the matrix through the evolution iterations. Traveling objects are ones that will traverse the matrix with unlimited bounds in each direction, if uninterrupted. All three of these interesting object categories have various initial states that fulfill the category's properties. The objects categories can then be divided further into subcategories. One of the most common objects is the glider, an object belonging to the traveling object category, has an initial state that will move in one direction in the matrix given its initial positioning. This object will continue in one direction infinitely if the matrix has no bounds and thus is a representation of simple emergence. Another common object, belonging to the oscillating category is the blinker, and object that is just three cells across in either a vertical or horizontal direction and will transition between being three cells vertical and horizontal through an infinite number of evolutions, with no external interruptions. Lastly one of the most important objects is the glider gun. This object creates glider objects from the same initial state and thus each glider created travels in the same pattern and direction. The importance of this object lies in its ability to endlessly create gliders in a stable way. If multiple glider guns along with other objects are placed in the initial state in specific ways, complex systems can be formed and thus making this object a staple in the reason why Conway's Game of Life is Turing complete.

3 Approach

Our approach to implement Conway's Game of Life in Python followed a sequential order. We decided to breakdown the entire implementation into five simple steps to assist in understanding.

3.1 Implementation Steps

1. Create a matrix to represent the evolution space.
2. Create a function to validate all 4 rules.
3. Create a function to handle a surrounded condition (8 cells surrounding the current cell) all 8 boundary conditions (current cell is not surrounded by 8 cells.
4. Hardcode patterns to be plotted on evolution space.
5. Use the command line to pass arguments that can select between plotting different initial patterns and to determine matrix size.

4 Results

This section will display the results of each object mentioned in this report when plotted in our Python Implementation. Figure 1 is a glider object. Figure 2 is a still object. Figure 3 is a blinker object. Figure 4 is a glider gun object.

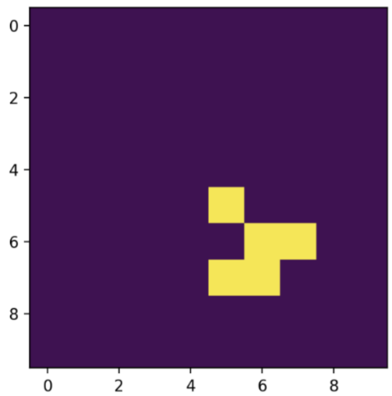

Figure 1

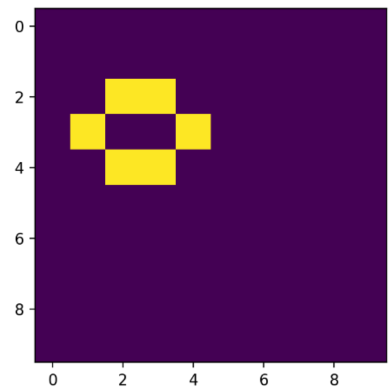

Figure 2

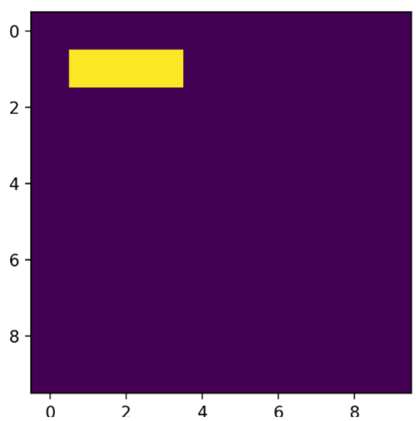

Figure 3

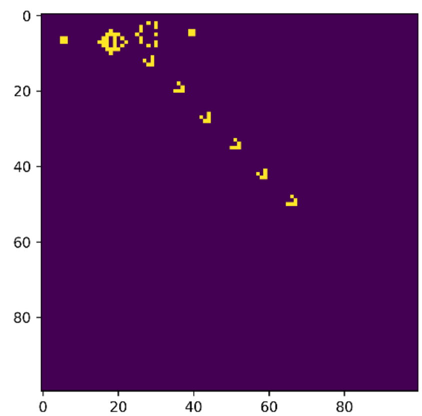

Figure 4

5 Conclusion

All in all, through our implementation of Conway's Game of Life, we can observe interesting objects and gain a better understanding of the interesting properties. We can better understand how objects such as blinkers, gliders, stills, and glider guns can be created and their significance. We can see how an initial state, the glider, and some simple rules can create simple emergence. We can also intuit how more complex objects such as the glider gun can be used in tandem with other objects to make Conway's Game of Life Turing complete.

6 Revisions Based on Feedback

Revisions that were made to this project based on the feedback were adding the glidergun initial state to the code, presentation, and report. A more intuitive approach to explaining how the game of life is turing complete was also pushed to all aspects of this project.

7 References

Eugene M. Izhikevich et al. (2015) Game of Life. Scholarpedia, 10(6):1816.