Mitchell Browe & Benjamin Freeman

MCS 4653 - 01

4/30/2022

Cellular Automata Checkpoint 3

One of the first topics covered in our Theory of Computation class this semester is state machines. The simplest explanation for what these are, is a simple machine with states which accept input to move to a different state. Cellular Automata are simple state machines. A given cell in any Cellular Automata will take input in the form of the state of surrounding cells in order to determine which state that cell will transition to. These transitions, based on the given states around a given cell, can be visualized with finite automata.

In terms of time spent on the project Mitchell and Ben met at least once a week, sometimes two or three times a week, and spent around 10-12 hours planning, creating, and polishing the final project. Both group members spent an equal amount of time and effort working on this project and therefore would be given an A based on this aspect.

For our final checkpoint 3 project we decided to continue working with Conway's Game of Life and expand on this idea in particular. In our first project we wrote the code to generate a 1D cellular automata based on Rule 30 of the 256 different rules for 1D cellular automata. In our second checkpoint project we dove deeper into cellular automata and began researching Conway’s “Game of Life”. Lastly, for our 3rd checkpoint project we modified the rulesets and seeds that we used in the project as well as experimented with the implementation of spontaneous life and death of a cell and long range interactions between cells.

Conway’s Game of Life is the premier example of Cellular Automata. While it is a relatively simple task to manifest the idea of, and to generate, a One-Dimensional or Two-Dimensional cellular automata, Conway’s Game of Life is such a massive, universal display of the concept of cellular automata, that it is truly wondrous.

To reiterate, from our first 2 project checkpoints, Conway’s game of life is based on a simple ruleset. In the “game”, the rules can be discreetly described as the following:

1. Any live cell with fewer than two live neighbors dies, as if by underpopulation.
2. Any live cell with two or three live neighbors lives on to the next generation.
3. Any live cell with more than three live neighbors dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbors becomes a live cell, as if by reproduction.

These simple rules for the basis of the Game of Life are extremely versatile, and can be seen to produce a massive amount of different patterns, shapes, interactions, and results, all based upon the user’s initial seeding of the universe. Pictured in Figure 1 is a snapshot of the Game of Life.

*We have neither given nor received unauthorized aid in completing this work, nor have w presented someone else's work as our own*

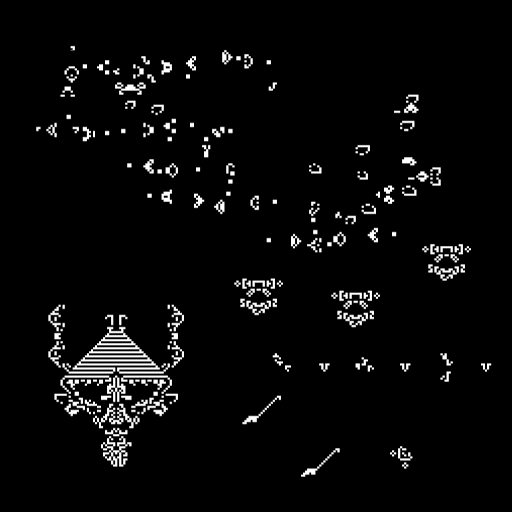


Figure 1.

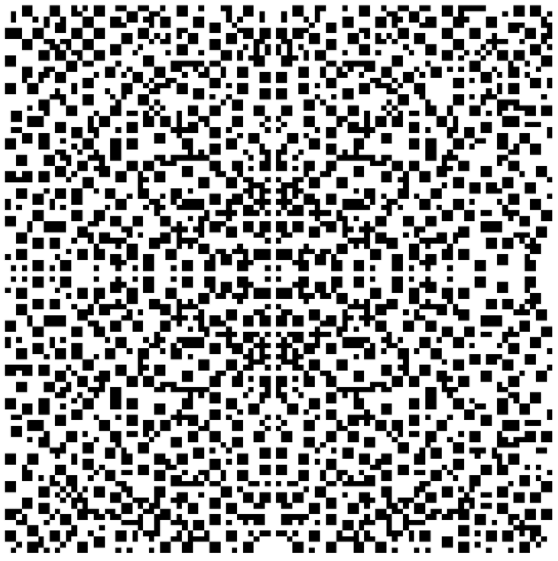
Moving on from project 2 we wanted to create Conway's Game of life on a larger scale so we decided to increase the size of our universe to 100x100. This increase in size greatly increased the possible combinations that could be used within our universe. We also experimented with changes in the ruleset being used to include rules such as:

1. Any live cell with more than 5 neighbors dies
2. Any live cell with fewer than 3 neighbors dies
3. Any live cell with 3 to 5 neighbors lives to the next generation
4. Any dead cell with 1 live neighbor comes to life

With this new ruleset we saw some interesting results in the patterns that were created when the program was run. (See Figure 2 and figure 3)

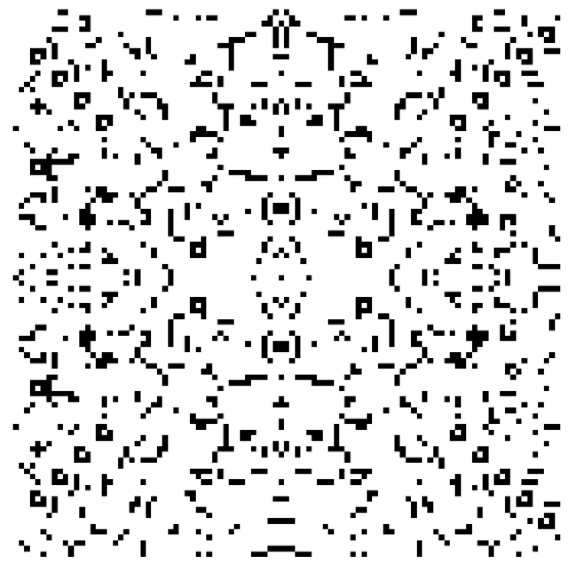


Figure 2. Normal Game of Life Ruleset

  
Figure 3. Our new Disease Survival ruleset

As you can see the change is drastic, and the seed no longer seemed to be confined to the middle of the universe and instead spread and multiplied. When we saw this we wanted to continue to explore the other possible modifications that could be made to the game of life.

The next modification that we implemented was the ability for cells to communicate and have an effect on their states from a longer distance so we ended up implementing a way for cells to interact with cells that were not just one cell away but two. The results of this are shown in figure 4 using the same seed as in figures 2 and 3.

  
 Figure 4. Our wide range interaction ruleset

Lastly we experimented with spontaneous life and death of cells based on a certain time period which could be synonymous with the lifespan of a living thing. We ended up picking arbitrary time periods of 20, 40 and 60 iterations before the possibility of spontaneous life can occur with a change of 5, 10 and 20 percent chance respectively. Figures 5 and 6 show the spontaneous life ruleset running in conjunction with the original rules for Conway’s Game of Life using the Beacon seed. The spontaneous life has just begun to work in Figure 5, and the cluster in the middle of the frame is where the beacon used to exist. In Figure 6, the spontaneous life has been active for some time.

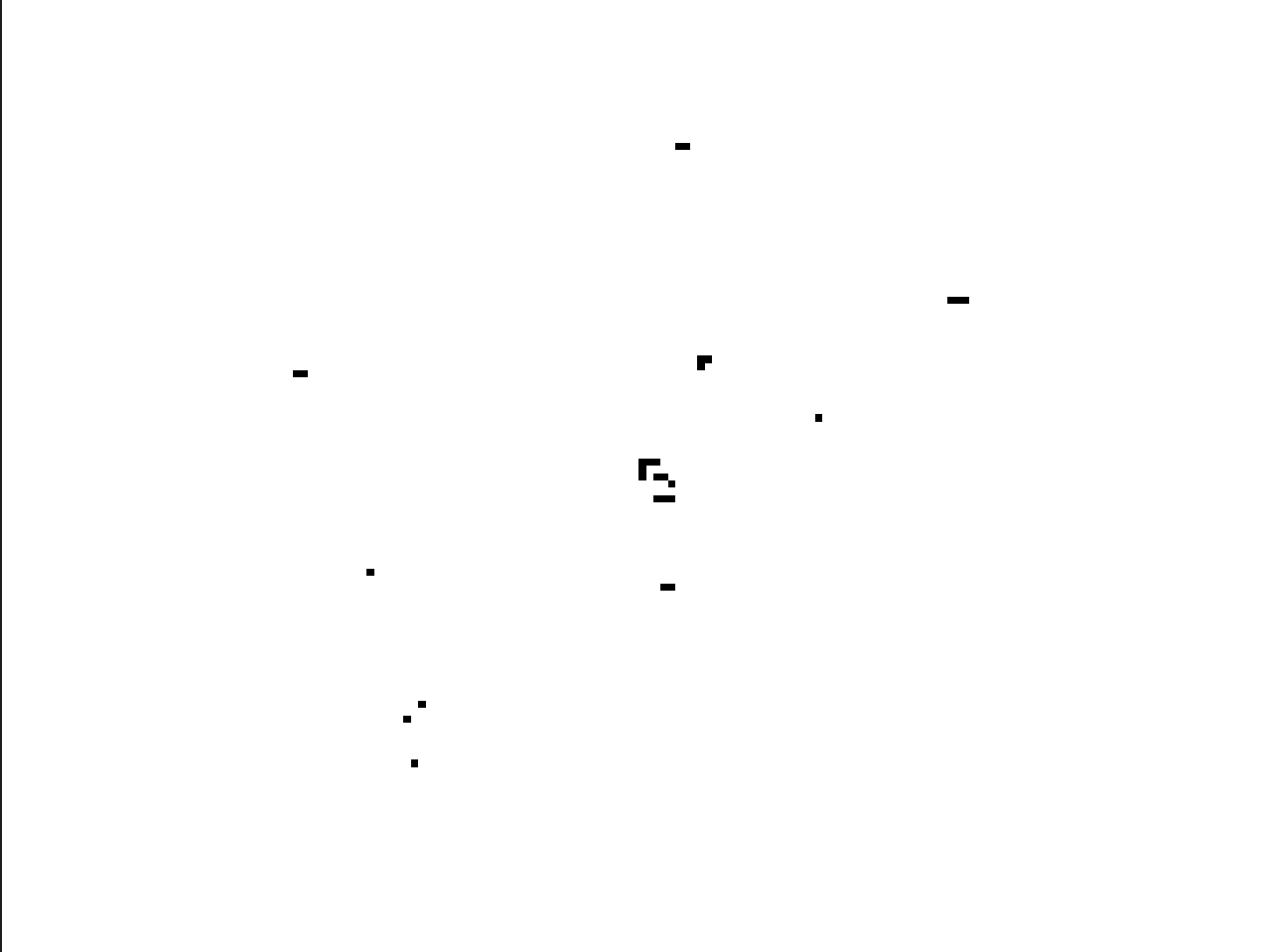
  
 Figure 5. Spontaneous life ruleset running with all other basic Game of Life Rules



Figure 6. Spontaneous life ruleset running with all other basic Game of Life Rules

Spontaneous death is also shown in Figure 7 running alongside the disease survival. In the original disease survival with the sun seed we see static boxes form that never leave the screen but with spontaneous death these boxes break apart and the elements in the universe continue to shift and move.

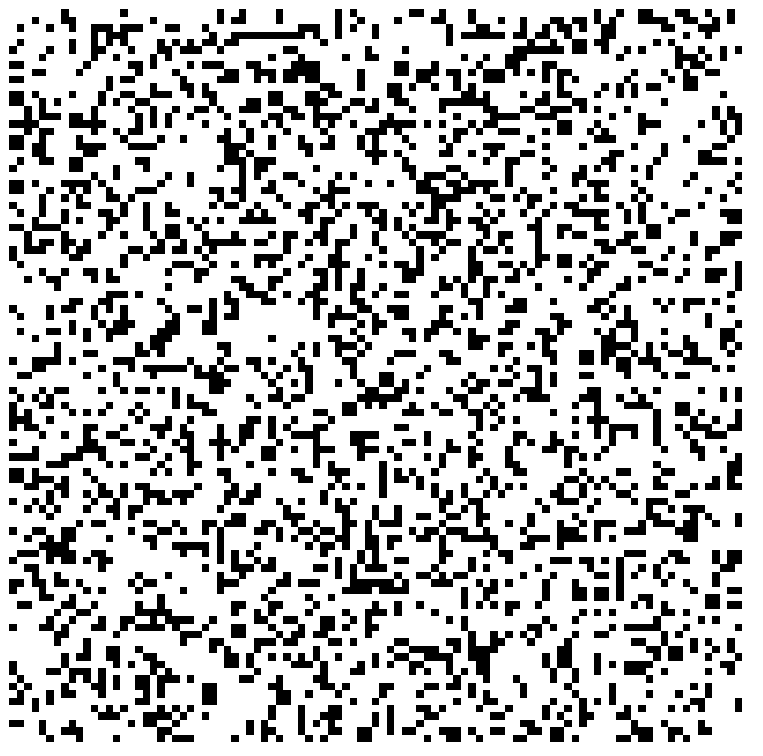


Figure 7. Spontaneous death ruleset running with Disease Survival rules

Upon running the Python code we have assembled, you will find a simple Gif in the working folder in Visual Studio Code.