Suyoung Park

Professor Ryan Kaufman

AMS 325

19 Dec 2024

Game of Life Simulation Algorithm Report

Game of Life is a game algorithm created by John Conway that simulates literally a "life" in which life and death are repeated along a certain pattern. With discrete distributions along the grid, each cell is expressed as alive ('X') or dead ('.'). This grid develops according to four rules.

1. Living cells die if there are less than two neighboring cells.
2. Living cells survive if there are two or three neighboring cells.
3. Living cells die if there are more than three neighboring cells.
4. Dead cells come to life if there are exactly three neighboring cells.

When trying to understand this rule intuitively, it seemed that the cells' life and death, with each cell having an average of three neighboring cells, were repeated. I didn't understand exactly how the algorithm could sustain that cycle, but I eventually found that by placing a predefined pattern in the center, the grid could repeat itself in multiple iterations, discovering it or drawing the pattern for fun.

To perform a part of each algorithm, we performed the following steps.

1. Initial setup: The grid starts by placing a specific pattern in the center. Because we used integr division, which is not exactly central, we need to set a sufficient number of horizontal and vertical compartments on the grid to prevent "index out of range" errors.

2. Evolution: I made it follow the logic described above. The thing I want to brag about in the code is that I used the dy and dx variables to process each condition branch at once using a repeat statement.
3. Handling of boundary conditions: We made sure that the code worked well on very small grids and large iterations. We also added conditions so that the cells on the edge are not compared to the cells on the outside.

Through this, we successfully implemented the Conway game life algorithm. First, we finished by plotting graphs automatically (every time we close) at 10-frame intervals, saving each frame as a video. The problem of dealing with the grid was like a graph theory problem. It was nice to be able to understand the interaction with the grid more and to use matplotlb to visualize the grid. I hope it will be a great help when solving graph theory problems such as BFS or DFS to others later.