

Variations of the Game of Life in Processing

Daniel Elliott, Chenkai Shen, Daniel Murdoch

Team Name: $\frac{d}{dc}$

Period 6

May 2023

0 Description

In this project, we are building a base grid, on which we can visualize different versions of the game of life by slightly modifying the game rules and factors such as the size of each cell on the grid, frame rate, or percentage alive at the start

1 Current functionalities

1.0 Grid

- Ability to modify max size(as a multiple of screen size for both width and height)
- Allows for varying strength computers to run simulations
- Ability to move throughout the grid with W,A,S, and D keys
- Ability to “zoom” into and out of the grid with plus and minus keys
 - current max zoom-in of 4x from base
 - * from base size of 40
 - * max zoom in is $\frac{\text{starting size}}{10}$
- Ability to modify the starting size of the grid
- Ability to modify the frame rate

1.1 Base game

- Follows the rule of game of life
- Each alive cell only stays alive if it has 2 or 3 alive neighbors
- Each dead cell becomes alive with exactly 3 alive neighbors Has initializer that fills grid with given percentage of alive cells at the start(best between 0.1 and 0.85) Can calculate the number of alive neighbors of any given cell

1.2 Wireworld

- Models circuits
- Has four types, empty cells, electron heads, electron tails, and conductors
- Empty cells stay empty, electron heads become electron tails, tails become conductors, and conductors become electron heads with one or two electron heads around them'
- Overrides alive neighbor calculator to calculate electron heads

2 Needed functionality

- Click modification to the grid at the start
 - This allows for better wireworld tests, so that we can actually begin generating circuits
- Rainbow game of life
 - gradient colors with smoother transitions
- Tsunami model – more complex but would be really cool
 - Based of off two grids, grid of water and of depth
- Create starting window, to pick model, set initial cells, and set initial rules in a more user friendly manner

3 Problems

- Class allocations. We had problems deciding in which class certain methods would go and we also changed these after the fact, meaning that code had to be rewritten to accommodate these changes
- Our zoom function currently changes the grid size, but this leads to some layering issues and it would be ideal if it did not change the grid size at all, only the display section.
- We had two people push progress at the same which led to progress being overwritten
- In one of our for loops, it never ended, leading to an unstable program that would eventually crash