

# Variations of the Game of Life in Processing

Daniel Elliott, Chenkai Shen, Daniel Murdoch

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

Period 6

June 8, 2023

## 0 Description

In this project, we created a base grid, with the capability to randomly generate starting values in all cells, and iterate from those starting values by following a given rule set. From there, we create three variations: Fire world, Wireworld, and Rainbow Game of Life by adjusting the rule set and generation parameters. We allow further manipulation by the ability to change the starting percentile, which adjusts the random generation drastically, and the ability for the user to input their own cell values as the generations iterate, further changing the created grids.

## 1 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 “zoom” into and out of the grid with plus and minus keys and scroll wheel
- Ability to modify the starting size of the grid
- Ability to modify the frame rate

### 1.1 Base Game

- Follows the rules of the 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 an initializer that fills the grid with the given percentage of alive cells at the start. Can calculate the number of alive neighbors of any given cell
- Keeps count of the number of alive vs dead cells.

### 1.2 Wire World

- 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

### 1.3 Fire model

- Creates forest with given percentile of alive trees
- Generates random grid
- Currently 40 percent all but guarantees no spread, 50 - 68 is either way, and more then 71 percent is guaranteed spread
- Each cell currently on fire checks the four cells to its north, west, south, and east, and sets them ablaze if they are a tree

### 1.4 Rainbow

- Dead squares are all black
- Alive squares are colored in a gradient, with each alive square's color being affected by its neighboring alive squares
- Smoothly fades to black with time

### 1.5 Start Menu

- Allows user to set starting percentile alive, starting display size, and max size for zoom
- Allows user to choose which game they wish to play

### 1.6 Console

- Displays text
- Can have computer prompts or handle user input
- Each enter press creates a new line
- Resets when filled
- Has get methods for the last user input

### 1.7 Button

- Can display on the screen at arbitrary position and size
- Have labels and values
- Have changeable values
- Have get methods for labels and values
- Have get methods for if button is clicked
- Visually display when clicked

## 2 Log

- Chenkai Shen
  - Added alive/dead cell counter, worked on counting neighbors, and debugged in base game
  - Added w a s d movement (later scrapped), + - zoom, and click input to grid
  - Added rainbow game of life
  - Added alive/dead cell counter in FireGame
  - Worked on all of the documents
  - Helped debug WireWorld
- Daniel Elliott
  - Added WireWorld
  - Added FireGame
  - Added intro screen
  - Added click modify, base rules, and ability to run base game
  - Added randomly generated, array system, and display for grid
  - Wrote "How does it work"
- Daniel Murdoch
  - Added scroll zoom functionality to grid
  - Fixed zoom screen problems
  - Added click-dragging feature which had to be scrapped later on
  - Made all of the UML diagrams
  - Helped debug rainbow game of life

## 3 UML Document

Attached separately

## 4 How does it work?

### 4.1 From intro screen:

- Click on each of the top three buttons:
  - Each time you will be prompted, when you are, enter a value (For best performance: 10-100 for size, 0 - 100 for percent, and 5-10 max size (Higher is likely to cause extreme lag)
- After entering the value, hit enter and see if the value is displayed in the console
- If you made an error in typing, click the button twice, wait to be prompted, and type again
- Once successful, move onto the next button
- Following the successful setting of the top three buttons, pick one of the four games, and select done

### 4.2 In all games:

- You can use the scroll wheel or +, - keys to move in and out, to see more or less of the game

### **4.3 Base game:**

- Objective: mess around with adding and removing squares
- You can click to change any given square between alive and dead, to see what you will create

### **4.4 Fire Simulation:**

- Objective: see what percentiles do what, or just lag your computer for fun
- Here, play around with the percent alive (i71 percent is all but guaranteed to propagate, j50 percent all but guaranteed to not)

### **4.5 WireWorld:**

- Objective: emulate your computer on your computer
- Press 0 to place conductors, which will propagate electric charge, press 1 to palace an electron, which will propagate over conductors, and see what you create(press 2 for an eraser)

### **4.6 Rainbow world:**

- Objective: Look at pretty colors
- Functions the same as base game