Ray Hu

CSE355

Representing Darwinism in Cellular Automata

https://www.youtube.com/watch?v=EBIzegoNChE&ab_channel=RayHu

Introduction:

For my extra credit project, I explored topics from biological evolution such as mutations, natural selection, and genetic drifts through a cellular automata. I got this idea from my biology class while learning the same subject. In this simulation, the environment is a giant grid where each colored square represents a cell that belongs to a colony of that same color. Every colony of cells will undergo evolution based on their immediate environment and base conditions to achieve the same goal of dominating the entire board. For this objective, every cell will have certain properties inspired by their real life counterpart.

Implementation:

Global Variables:

The global variables in this simulation act as the base conditions for the cells. In this project, three variables are used.

- MUTATION_RATE: This probability value sets how often cells will mutate from one generation to the next.
- BIG_MUTATION: This probability is the rarer version of the MUTATION_RATE variable and will have special conditions attached to it.
- AGE_LIMIT: This value sets the age limit of all cells inside the simulation.

Classes:

Cells: Every cell is an object of the Cell class. This class contains five attributes:

- Strength: A value representing how strong the cell is.
- Reproduction: A value representing how likely the cell will reproduce.
- Colony: a variable that holds the reference to a colony object.
- Age: a value that represents the current age of the cell.
- AgeLimit: a value that represents how old the cell can grow before it dies.

Colony: The colony object holds the starting and basic information of every cell object part of the colony. Each colony object contains six attributes:

- initialStrength: The strength set to 127.
- initialReproduction: The reproduction value set to 127.
- Color: A random RGB tuple that gets generated on initialization.
- Count: A value that represents the current count of cells within the colony.
- totalStrength: A value that represents the added strength value from all cells within the colony.
- totalReproduction: A value that represents the added reproduction value from all cells within the colony.

Simulation Mechanics:

Movement: Each cell will have a chance to move in either of four directions: UP, DOWN, LEFT, RIGHT. In the situation where an existing cell already exists in the direction the cell is trying to move, two outcomes can happen.

- The other cell belongs in the same colony: The current cell will stay in place and not move this round
- The other cell belongs to another colony: The two cells will fight to determine who will own that spot next round.

Combat: This mechanic happens when two cells from different colonies try to mov into each other's territory. The rules for this are very basic. The two cells will compare their strength values and the cell with higher value will win the engagement. The losing cell dies and its spot will be taken over by the winning cell.

Age: Each cell has a counter that represents their age which increases by one per round. Once the age of the cell reaches their age limit, they will die and free up their space.

Reproduction: Each cell will have a chance at the beginning of the round to reproduce. This probability is based on each cell's reproduction value. If the cell decides to reproduce, it can create another cell in either of five places: UP, DOWN, LEFT, RIGHT, and STAY. If there is space available in any of the four directions, a child cell of the current cell will be created in that place. However, if there are no spaces left or the cell reproduction is decided in STAY, then the current cell will die and its child cell will take its place.

Reproduction Mechanics

When a child cell is created, it will take in its parent's strength value, reproduction value, and colony reference as parameters to form the cell's base values.In the constructor of the cell class, there is a chance, depending on the mutation global variables, to mutate those base values.

Regular Mutation: The new cell will randomly select a new strength or reproduction from a range of values based off the parent values

Strength:

$$0 <= parentStrength - 10 < childStrength < parentStrength + 10$$

Reproduction:

$$0 <= parentReproduction - 10 < childReproduction < parentReproduction + 10 <= 255$$

Big Mutation: The new cell gains a massive strength increase from their parent values but also suffer drawbacks for their reproduction value and age limit.

Strength:

$$childStrength = parentStrength + 100$$

Reproduction:

$$childReproduction = parentReproduction - 50$$

AgeLimit:

$$ageLimit = AGE\_LIMIT / 2$$

Results:

In my simulation, I was able to see different emergent behaviors that were representative of the Darwinistic attributes. Each colony of cells was able to have their average values changed over generations to better improve their chances of survival against each other. The values we see at the end will be very different from the ones from the beginning. Furthermore, by changing the initial conditions in the beginning, we also see differences in where colonies prioritize their values. Overall, the outcomes were both expected and unexpected as each new run presents results different from the last.