CS-439

Project 2 Cella Rule 150

Team CMM

Members:

Carlos Meza

Tasks

1. Understand the rules for Conway's Game of life.
2. Develop a deterministic state automaton that represents the movement of the Turing Machine head. (Developed a circular/spiral movement to determine next cell generation)
3. Read reference for p5.js, which is a external library that simplifies drawing on a HTML canvas.
4. Implement Conway's Game of Life in Javascript
    a. Create 2D grid canvas
    b. Create Functions
        i. Setup()- initializes html canvas, initializes 2D array that represents population, initializes state diagram.
        ii. makeGrid()- creates 2d grid which is used in setup() function
        iii. createStateTable()- initializes state table for TM transitons
        iv. drawStateTable()- displays State Diagram on another html canvas separate from grid.
        v. draw()- main function that performs TM operations for creating game of life and performs canvas transfromation.
        vi. blink()- creates flash animation for representation of given state
        vii. keyPressed()- function that when key is pressed it the Grid generates a random population and performs game of life.
5. Complete project documentation for project 3
6. Turn in project

Progress Board <Project Game of Life> 4/20/19

Team: CMM = Carlos Meza

Ready:

Tasks 1-3 are completed (Basic understanding of Conaway's Game of Life along with p5.js which is an external library that simplifies canvas drawing.

Working:

So far nothing code wise is working just understanding of tools needed to implement project.

Done:

Tasks 1-3 are completed mainly basic understanding of project.

Issues:

Fine tuning state diagram to represent movement of TM head.

Also, there is difficulty understanding animation implementation for state diagram when TM is in given state.

Verified:

#1-3 Accomplished understanding of Conway's Game of Life, and it's rulesets. Along with tools needed to display population on to HTML canvas. (QA by Carlos Meza, Wk by Carlos 4/19)

Progress Board <Project Game of Life> 4/27/19

Team: CMM = Carlos Meza

Ready:

Task 4 is completed the animation of Conway's Game of Life.

Working:

Conway's Game of Life is working on html canvas.

Done:

Tasks 1-4, the main implementation of project, are complete.

Issues:

Also, there is difficulty understanding animation implementation for state diagram when TM is in given state. Was able to show state animation but at great cost to performance, meaning the screen would of freeze with this feature present.

Verified:

#1- 4 Accomplished and completed the implementation of Game of Life on html Canvas. (QA by Carlos Meza, Wk by Carlos 4/26)

Complexity Report

Input Size/ Memory:

- 2D Tape N x N.
  - N = 30
  - M =30
- stateTable = K.
  - K = 11

Main Operations:

1. **Setup()-** initializes html canvas, initializes 2D array that represents population, initializes state diagram.
2. **makeGrid()-** creates 2d grid which is used in setup() function
3. **displayGrid()-** displays grid onto html canvas.
4. **createStateTable()-** initializes state table for TM transitions
5. **drawStateTable()-** displays State Diagram on another html canvas separate from grid.
6. **draw()-** main function that performs TM operations for creating game of life and performs canvas transformation.
7. **blink()-** creates flash animation for representation of given state
8. **keyPressed()-** function that when key is pressed it the Grid generates a random population and performs game of life.

Time Complexity:

- **Setup()** = 2 + makeGrid() + N^2 + N = O(N^2 + 2N) = O(N^2)
- **makeGrid()** = 2 + (N* 1) = O(N + 2) = O(N)
- **displayGrid()** = $6N^2$ + 6N = O(N^2)
- **createStateTable()** = N + 2 = O(N)
- **drawStateTable()** = createStateTable()+12N + 16 = O(N)
- draw() = displayGrid() + drawStateTable() + makeGrid()+ 280(N^2 + N) =
  - O(N^2 + N + + 280N^2 + 280N) = O(N^2)
- **blink()** = 19 = O(1);
- **keyPressed()** = makeGrid() + N^2 + N + draw() = O(2N +2N^2) = O(N^2)