Conway's Game of Life

An implementation of List Life algorithm in JavaScript.

Pragya Jaiswal
Cluster Innovation Centre, University of Delhi
Delhi, India
pragya.jswl@gmail.com

Prashant Sinha
Cluster Innovation Centre, University of Delhi
Delhi, India
prashant@ducic.ac.in

***Abstract*— Game of Life is a cellular automaton devised by the British mathematician John Horton Conway in 1970 [1]. Presented here is a JavaScript implementation of List Life algorithm for the Game projected on an isometric grid.**

Keywords—cellular automata; JavaScript; game of life;

I. INTRODUCTION

A cellular automaton consists of a grid of cells that are finite state automatons, that is, the grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood is defined relative to the specified cell [2]. An initial state (time t = 0) is selected by assigning a state for each cell. A new generation is created (advancing t by 1), according to set of fixed rules that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood.

Game of Life is a type of cellular automaton of binary state finite automaton grids.

II. LIFE GAME

A. Game Rules

The universe of the Game of Life is an infinite two-dimensional grid of binary-state cells. Each cell can have either a `alive`, or `dead` state. Every one of the cells can interact within its Moore Neighborhood [3]. The Moore Neighborhood is the set of all eight surrounding cells of a particular cell in the grid.

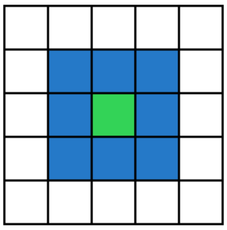

Fig. 1. *Moore neighborhood for Green cell is the set of all Blue cells.*

The interaction function that defines the transition is defined as a function of the number of neighbor cells that are alive. It is given as:

1) Live cell with fewer than two live neighbors dies (under-population).

2) Live cell with two or three live neighbors lives on to the next generation.

3) Live cell with more than three live neighbors dies (overcrowding).

4) Dead cell with exactly three live neighbors becomes a live cell (reproduction).

B. Game Patterns

Various different types of patterns occur in the Game of Life. These include:

1) Still lifes: These do not change between iterations.

2) Oscillators: These oscillate between different repeating patterns with a given period.

3) Spaceships: These patterns translate themselves across the universe of the Game. [4]

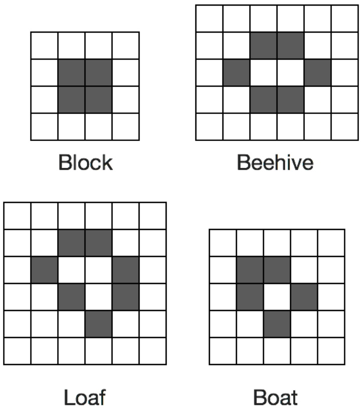

Fig. 2. Still life patterns.

III. IMPLEMENTATION

A. List Life

The List Life algorithm [5] is an array mapped Game board. The optimizations are mainly to allow skipping the dead cells in the list.

1) The universe is represented as a list of the co-ordinates of the live cells in scan-line order.

2) This is scanned by three pointers, for the current row and the rows above and below.

3) One co-ordinate per cell. Instead of storing the Y co-ordinate for every cell in a row, store it once at the start of the row and only store the X co-ordinate of each cell in the row.

4) In order to distinguish between X and Y co-ordinates, give them opposite signs, e.g. all X co-ordinates are positive and all Y co-ordinates are negative.

The implementation in JavaScript follows closely to the original implementation in C. The dead neighbors are estimated by checking the states of following neighbors:

```
[x - 1, y - 1, 1]
[x,     y - 1, 1]
[x + 1, y - 1, 1]
[x - 1, y,     1]
[x + 1, y,     1]
[x - 1, y + 1, 1]
[x,     y + 1, 1]
[x + 1, y + 1, 1]
```

B. HTML5 Canvas

There are two implemented HTML5 Canvas elements. Former one shows the 2D overview of the Game. The later one is used to projects isometric pixels of the Game.

The height and width of the canvas is determined dynamically after the page load using JavaScript directive `{`window.innerHeight`,`window.innerWidth`}` which sizes the canvas equal to the viewport.

The simulation is controlled via runtime semaphore interrupts.

IV. AUTHOR'S NOTES

The implementation and the source are hosted on GitHub at `https://prshnts.in/game-of-life/`.

REFERENCES

[1] Stephen Wolfram, Reviews of Modern Physics.

[2] Lemont B. Kier, Paul G. Seybold, and Chao-Kun Cheng, Modeling Chemical Systems using Cellular Automata: Springer.

[3] Conwaylife. [Online]. http://www.conwaylife.com/w/index.php?title=R-pentomino

[4] Tony Finch. Tony Finch's homepage. [Online]. http://dotat.at/prog/life/life.html

[5] Martin Gardner. (2009, June) Mathematical Games – The fantastic combinations of John Conway's new solitaire game "life". [Online]. http://web.archive.org/web/20090603015231/http://ddi.cs.uni-potsdam.de/HyFISCH/Produzieren/lis_projekt/proj_gamelife/ConwayScientificAmerican.htm