Game Of Life

Algorithms, Data Structures & Software Engineering mini-project documentation

Christian Tsalidis

June 10, 2022

Contents

1 Introduction

My goal with this project was to code the behaviour for Conway's Game Of Life with a high resolution. In terms of software I used Unity [1] with C#. For increasing the resolution of the grid I wanted to utilise the graphical processing unit (GPU) for calculations, therefore I used high level shader language (HLSL) for creating an algorithm based on parallelism in separate threads to increase the performance and resolution. In order to achieve this, I delved into learning how to use compute shaders in Unity. All the code we used for this project is available at my GitHub repository [2].

2 Conway's Game of Life and Related Work

Cellular automata (CA) have origins around the 1950s but became popular with John Conway's Game Of Life in 1970 with the Scientific American article from Martin Gardner [3]. This report follows the explanation of this algorithm following said article, and using Carter Bays' introduction to cellular automata [2]. It is standard to describe CAs rules in two dimensions with Moore neighbours [2], as follows: given $E1, E2, .../F1, F2, ...$ where the environment (E) specifies the number of live neighbors required to keep a cell alive, and fertility (F) represents the number to have a cell come to life. In Conway's Game of life. this rule is 2,3/3.

The Game of Life consists of a two-dimensional orthogonal grid of squared cells. Each cell has two possible states: either dead or alive. Each cell has eight corresponding neighbouring cells directly adjacent to it: vertically, horizontally and diagonally. The first generation of cells placed in the grid make up a pattern and seed for the following generations. These generations change based on aforementioned rules that apply to each cell, listed as follows.

- **Underpopulation**: any live cell with fewer than two live neighbours dies.
- **Overpopulation**: any live cell with more than three live neighbours dies.
- **Survival** Any live cell with two or three live neighbours lives on to the next generation.
- **Reproduction**: any dead cell with exactly three live neighbours will become alive.

Bays also explores the possibility for candidates for 3D game of life rules, with the most appropriate being 45/55 and 57/66 [1].

3 Implementation and results

In Unity, I created a two dimensional grid of cells. Each cell was assigned with a cube. Then, each frame, I would apply the game of life's rules. Listing 1 illustrates counting each cell'e neigbours and enforcing said rules.

```
int neighbors = CountNeighbours(grid, i, j);
// reproduction
if (state == 0 && neighbors == 3) next[i, j] = 1;
// underpopulation / overpopulation
else if (state == 1 && (neighbors < 2 || neighbors > 3)) next[i, j] = 0;
// survival
else next[i, j] = state;
```

Listing 1: Conway's game of life cell rules.

[1] https://unity.com/
[2] https://github.com/DidStuffStudio/GameOfLife

4 Results

The following figures illustrate my implementation of Conway's game of life in both two [Figures 1 and 2] and three dimensions [Figure 3], with varying grid resolution.

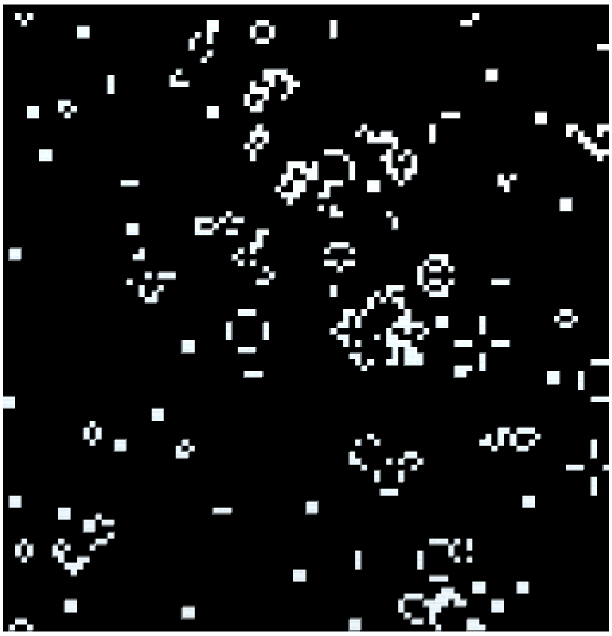

Figure 1: Game of life in 2d on a 100x100 grid.

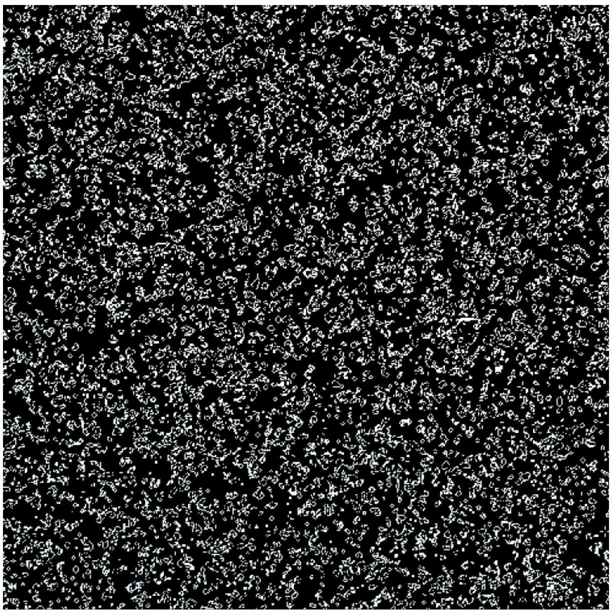

Figure 2: Game of life in 2d on a 500x500 grid.

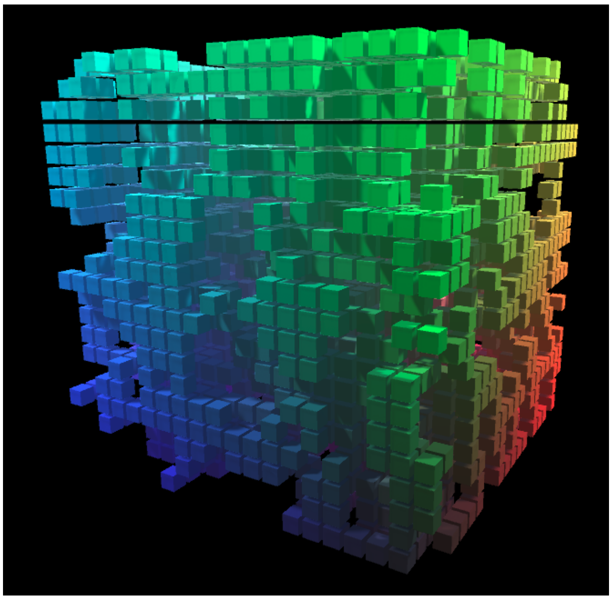

Figure 3: Game Of Life in 3D on a 20x20x20 grid.

5 Conclusion and future work

In the scope of this report, I was able to successfully implement a cellular automata that follows Conway's game of life rules. However, it is worth noting that this wasn't implemented with compute shaders, which I would like to to in order to be able to increase the resolution of my system. Moreover in terms of future work, I would like to generate music based on this celullar automata behaviour, as well as playing around with the rules for generating the game of life in 3D. The latter is due to the fact that I would like to be able to see more known patterns in the 3D visualization, as the known 45/55 and 57/66 rules did not give me appropriate results.

References

[1] Carter Bays et al. "Candidates for the game of life in three dimensions". In: *Complex Systems* 1.3 (1987), pp. 373–400.

[2] Carter Bays. "Introduction to cellular automata and Conway's Game of Life". In: *Game of Life Cellular Automata*. Springer, 2010, pp. 1–7.

[3] Mathematical Games. "The fantastic combinations of John Conway's new solitaire game "life" by Martin Gardner". In: *Scientific American* 223 (1970), pp. 120–123.