Cellular Automata for Procedural Level Generation

Hendrikus Bimawan Satrianto 13514066
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514066@std.stei.itb.ac.id

***Abstract*—This paper presents a reliable method to procedurally generate cavelike level maps based on the self-organization capabilities of cellular automata (CA).**

Keywords—games; cellular automata; procedural generation

I. Introduction (*Heading 1*)

A process of level design is crucial in every game development. Unfortunately currently vast and open game is the trend. This means a lot more resources needed to be put for the level design. To overcome this problem procedural generation can be used within level design process.

The use of procedural generation in game level design has been used as early as 1980 by a video game known as *Rogue*. It used procedural generation to create random tile-based dungeon layout made out of ASCII. The procedural generation would define rooms, hallways, monsters, and treasure that the player would find. Later on game with procedural generation for tile-based level would be known as its own genre Roguelike.

Modern use of procedural generation can be found in 2015 game *Galak-Z:The Dimensional*. In this game player would assume control of a space fighter pilot navigating cavern within asteroid where space pirates hides. Procedural generation technique called *cellular automata*. With this technique it is possible to generate various hollow space of surrounded by a wall akin to a cave. It is this *cellular automata* technique that will be discussed further in this paper.

II. Theoretical Basis

A cellular automaton consists of a regular grid of *cells*, each in one of a finite number of states, such as *on* and *off*. 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. Cellular automaton works by each cell observing its neighborhood and changing its state according to what state its neighbor currently is.

With cellular automata it is possible to generate cave-like structures. The basic idea is to create a grid with cell of two possible state, *wall* or *floor*. The procedural generation start by filling the grid's cell with randomly selected state, then repeatedly apply this 4-5 rules:

1. If there is less than 4 walls in the neighborhood the cell becomes floor.
2. If there is more than 5 walls in the neighborhood the cell becomes wall.
3. If there is exactly 4-5 walls in the neighborhood the cell state stayed the same.

III. Implementation

The cellular automata procedural generation is created within Unity3d game engine.

The scene itself only contain:

- Camera
- Light source
- mapGenerator
    - Mesh filter
    - Mesh renderer
    - Script

The level created by converting 2D array, which comes from cellular automata, into a mesh or a 3D model.

The cellular automata itself works according with this algorithm:

```
void GenerateMap() {
  map = new int[width,height];
  RandomFillMap();

  for (int i = 0; i < 5; i ++) {
    for (int x = 0; x < width; x ++) {
      for (int y = 0; y < height; y ++) {
                int neighbourWallTiles =
GetSurroundingWallCount(x,y);

        if (neighbourWallTiles > 4)
          map[x,y] = 1;
```

```
            else if (neighbourWallTiles < 4)
              map[x,y] = 0;

          }
        }
      }
}
```

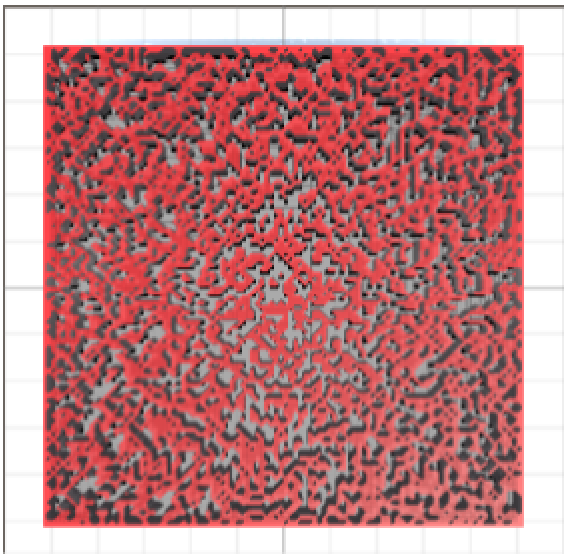

Figure1. Random seed of a map

What happen is first each cell in the 2D array or grid are assigned a random value of 1 or 0 with 1 means it is a wall and 0 means it is a floor. Next, for 5 cycle every cell in the grid is iterated with 4-5 rules:

1. If there is less than 4 walls in the neighborhood the cell becomes floor.
2. If there is more than 5 walls in the neighborhood the cell becomes wall.
3. If there is exactly 4-5 walls in the neighborhood the cell state stayed the same.

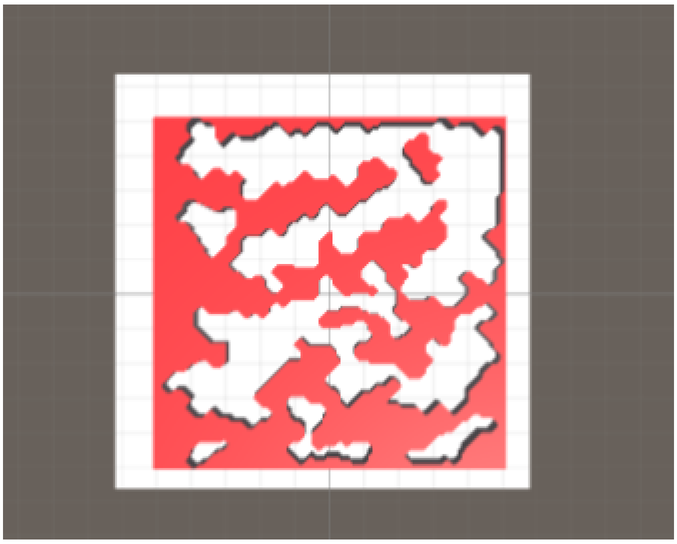

Figure 2. Map after applied cellular automata

IV. Conclusion

Cellular automata can be used to create a randomized corridor-like level map.

Acknowledgment

I would like to Dessi Puji Lestari ST,M.Eng.,Ph.D., Dr. Ir. Rinaldi Munir, MT, and Dr. Eng. Ayu Purwarianti, ST.,MT for your teachings throughout the semester in IF3280 course.

References

[1] Johnson, L., Yannakakis, Georgios N., and Togelius J., "Cellular automata for real-time generation of infinite cave levels"

Statement

I hereby declare that this paper is my own work and not a copy, translation, nor plagiarism of somebody else's work.

Bandung, May 5 2017

Hendrikus Bimawan Satrianto 13514066