



Map Generation in Games

Ain't no mountain high enough!

Map Generation

Maps in video games were generated using different methods

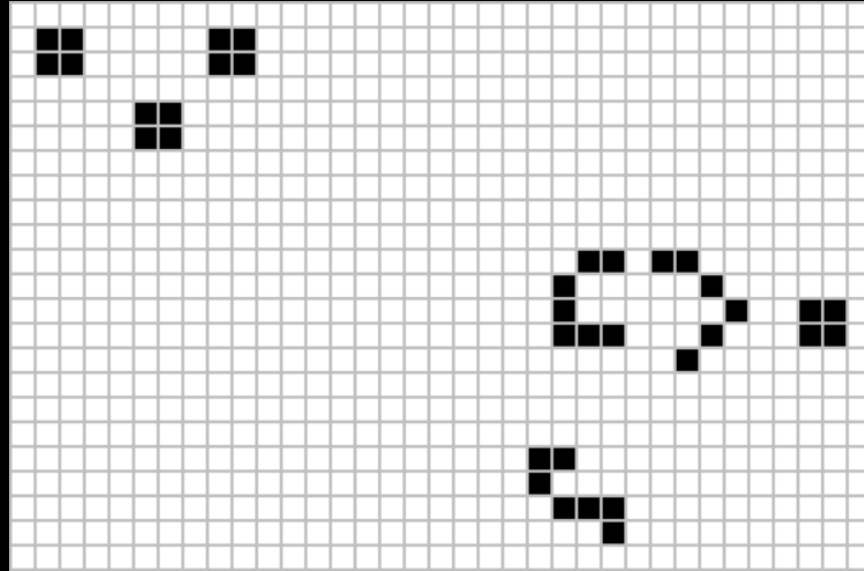
- Manually designing the terrains (time-consuming and challenging)
- Automated generation (The game generates it)



We came across an idea to use a popular "Cellular automation", i.e., Conway's game of life and tried to generate random terrains.



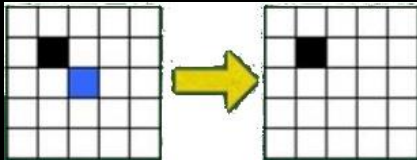
Conway's Game of Life



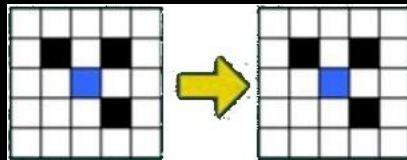
Conway's Game of Life

Rules:

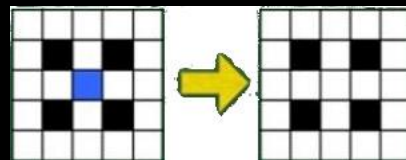
1. Any live cell with fewer than two live neighbours dies, as if by underpopulation.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overpopulation.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.



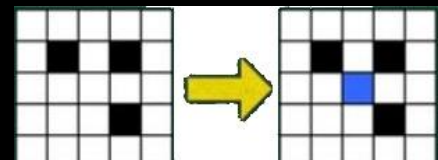
Rule 1



Rule 2



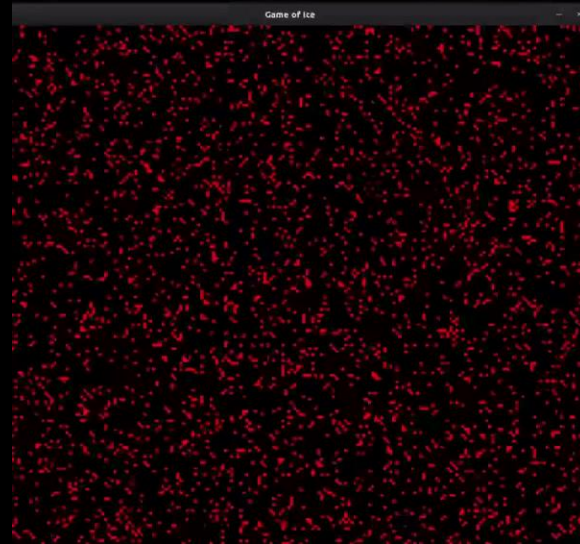
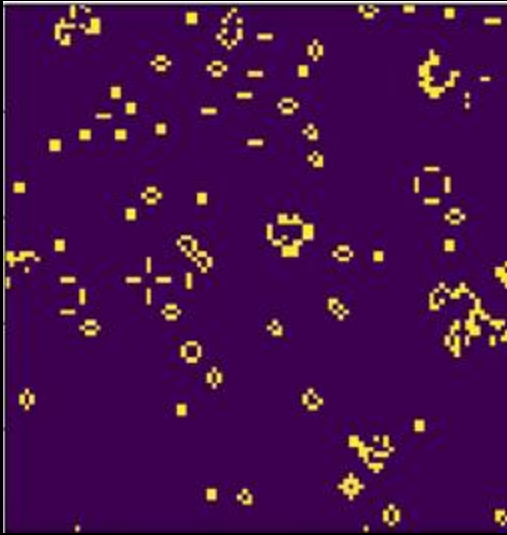
Rule 3



Rule 4

Conway's Game of Life - Output

When all these rules are applied, we get this:



Map Generation in Video games

Two ways

Automatic:

Used in games that require continuous map generation. Eg: Minecraft, Terraria.



Pre-designed:

Used in games that have a constant map. Eg: CS:GO, GTA.



Map generation using Conway's game of Life

- We ran 100 different independent simulations of Conway's game of life.

Map generation using Conway's game of Life

- We ran 100 different independent simulations of Conway's game of life.
- Each simulation ran for a 100 *epochs / iterations* (generations).

Map generation using Conway's game of Life

- We ran 100 different independent simulations of Conway's game of life.
- Each simulation ran for a 100 *epochs / iterations* (generations).
- Then, we added these 100 different final outputs.

1	2	3		0	2	4		1	4	7
4	5	6	+	4	6	8	=	8	11	14
7	8	9		6	8	9		13	16	18

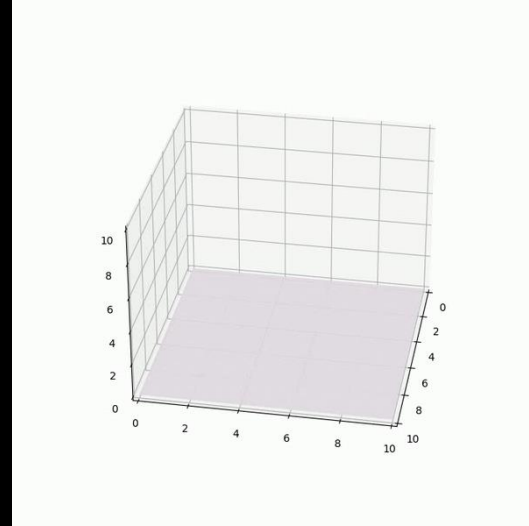
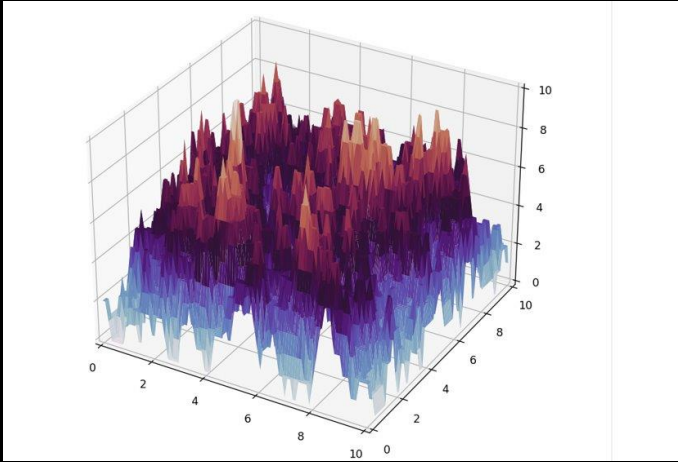
Map generation using Conway's game of Life

- We ran 100 different independent simulations of Conway's game of life.
- Each simulation ran for a 100 *epochs / iterations*.
- Then, we added these 100 different final outputs.
- This is what we used to generate our map
 - The higher numbers represented peaks / mountains
 - The lower numbers represented the valleys / trenches
 - The middle-ranged numbers represented plains / middle ground

Map generation using Conway's game of Life

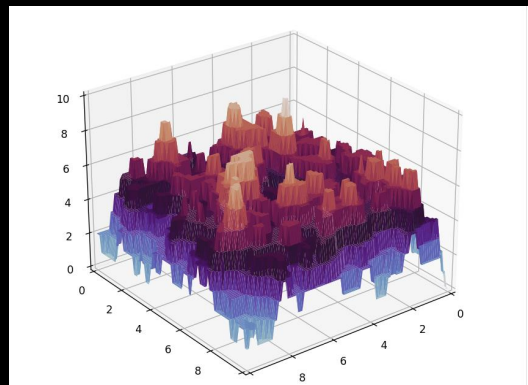
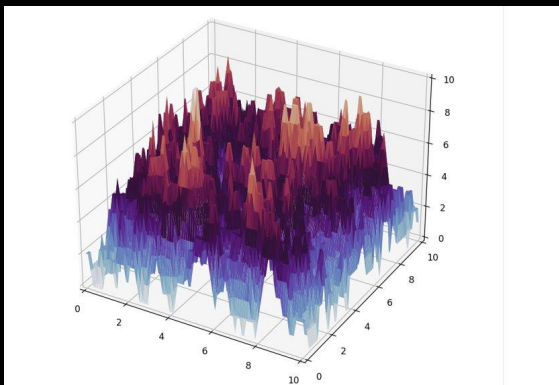
- We ran 100 different independent simulations of Conway's game of life.
- Each simulation ran for a 100 *epochs / iterations*.
- Then, we added these 100 different final outputs.
- This is what we used to generate our map
 - The higher numbers represented peaks / mountains
 - The lower numbers represented the valleys / trenches
 - The middle-ranged numbers represented plains / middle ground

Map generation using Conway's game of Life

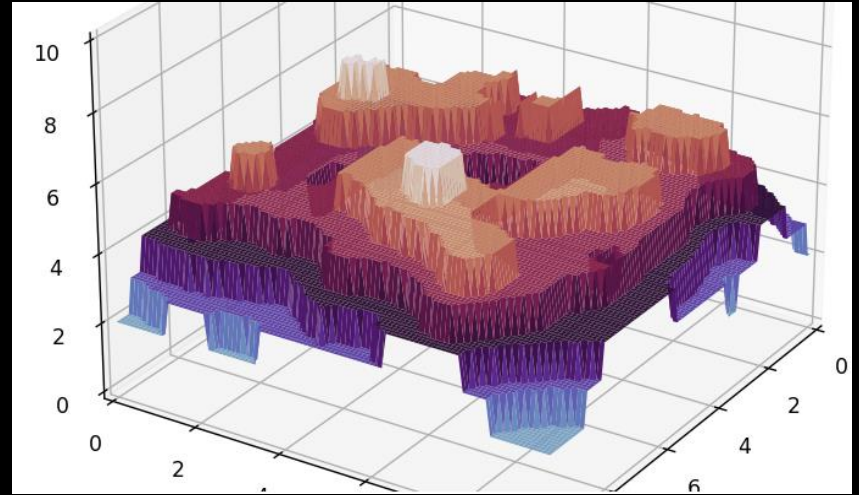
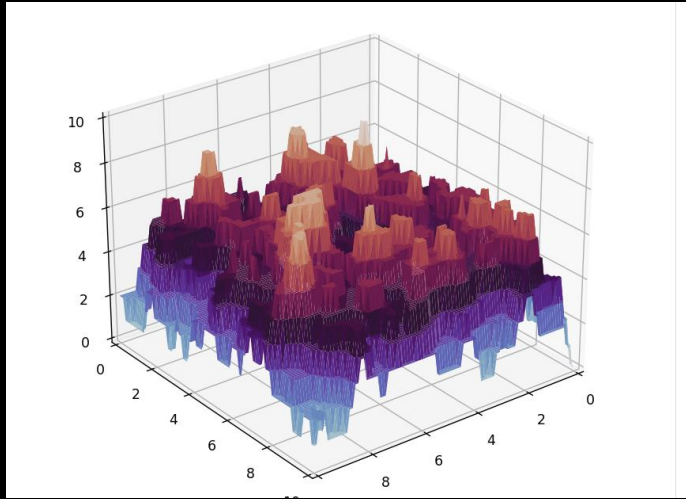


Map generation using Conway's game of Life

- You may notice that there are a lot of irregular peaks and trenches. This looked a bit weird to us as well, so we decided to look into ways to smoothen the generated terrain.
- We looked online and found a mathematical technique called Gaussian Smoothing that helped us achieve this exact outcome.



Maps generated after smoothing



Demostration of the map

Resources

- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
- <https://www.8bitmen.com/procedural-generation-a-comprehensive-guide-in-simple-words/>
- Find the code here: <https://github.com/MEC-Enigma/map-game-of-life>
- <https://github.com/IceCereal/The-Game-Of-Life>

Thanks for tuning in!