EROSION ALGORITHMS FOR BETTER TERRAIN SIMULATION

In graphics, erosion algorithms are used to simulate terrains accurately. Erosion is the process of natural terrains being worn out and being transported away by natural forces. In real-life, the way terrains look is due to different types of **erosion.** In graphics, the following are the main ones used:

- Hydraulic Erosion: This is caused by water flowing over terrain and making river grooves or slopes. As water flows down, it carries all the eroded materials such as sediments with it. When it settles down, all the sediments are released creating a sediment deposit.
- Thermal Erosion: This simulates the effect of gravity over different iterations on the sediments that form the terrain.
- Wind Erosion: Simulates the effect of wind especially in arid regions. Some examples can be dunes, plateaus or basically any other terrain where the wind shifts loose materials.

There are several multi-erosion algorithms that combine the effects of the above-mentioned erosions.

Using different height values on a 2D grid, we can create a terrain without grooves or caves that looks sufficiently good. Most games (for example, Minecraft) layer different types of coherent noise such as Perlin Noise and Simplex noise. These terrains don't form grooves or caves, but at a large scale these things are not noticeable.

Adjusting the amplitudes and focusing on the details, we can create a rough pattern called the fractional Brownian motion. There are also different ways of creating a better and smoother looking terrain using noise but when compared to real life, they don't hold up as well.

So, if we were to choose only one of the algorithms to make better and more accurate-looking terrains, the best algorithm to use would be hydraulic erosion. The reason why I chose this algorithm is because of its versatility and that's due to the following reasons:

- Creates grooves, slopes, and overhangs that mimic most of the real terrains.

- Creates sediment deposits that give a better texture making the terrain more visually appealing.
- Since this algorithm dictates how the water interacts with the terrain, it creates much more dynamic and versatile terrains that create rivers, river beads, canyons etc.

Hydraulic erosion also had a lot more resources and tutorials (from what I could find) as compared to the other types, making it more accessible and easier to approach.

References

Dandrino. (n.d.). GitHub - dandrino/terrain-erosion-3-ways: Three Ways of Generating Terrain with Erosion Features. Retrieved from https://github.com/dandrino/terrain-erosion-3-ways.

Sebastian Lague. (2019, February 17). Coding Adventure: Hydraulic Erosion- YouTube [Video file]. Retrieved from https://www.youtube.com/undefined.