# Sprint 6 – Implement Hydraulic Erosion 22/03/2019 – 31/03/2019

# Abstract

# Research

After researching for a few hours, I came across an application named Instant Terra, this is a 3D application made in C++ for Procedural Terrain Generation. I used this application and got an idea in what I should expect from the hydraulic erosion in my terrain (Figure 1, Figure 2 and Figure 3).

The first step when reading the journal (Terrain Generation using Procedural model Based on hydrology), I figure out that I needed to learn how to create fractals. I am starting to learn how to draw some basic lines on unity (Figure 4).

## Line Class

First, I started creating a Line Class, this class have a method for Drawing a line, each time is called it creates a game object and it stores it inside a list. The struggle here was to draw the line and being able to draw the next line saving the position from the line before and apply a specific rotation if needed. To be able to do this I created two vector 3 variables, a quaternion and a Boolean for checking if this is the first line (Figure 6). I defined that the game object position is going to be equal to the start position and, in the end, when the line is created, the start position is going to be equals to the end position. When going through the list I had to find a way to check if the line was not drawn yet, for this I did an if statement to check if the game object had the line Renderer component, if not then draw the next line (Figure 5).

For the result in code check (Figure 7), the first line draws a line with the length defined in the parameter, the next code sets the rotation for the next line and after draws another line (Figure 4).

When I am trying to call this function recursively, I am getting this strange problem (Figure 8), it seems that I need to find the exact point were the line finishes and apply the translation from that point.

After many hours spent on the attempt to create a fractal, I ended up with two interesting fractal pattern using only lines in unity (Figure 9)(Figure 10). My end goal is to be able to create different patterns for example a tree, this technique is going to allow me to draw the water paths for the hydraulic erosion.

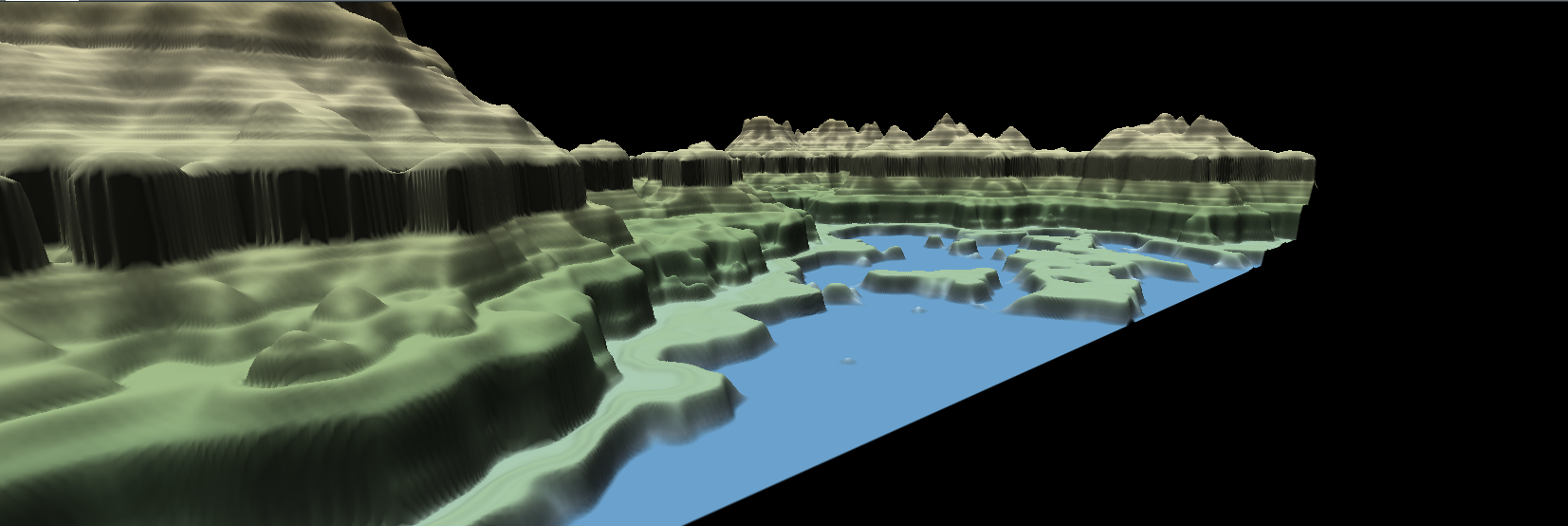


Figure - Instant Terra, terrain without hydraulic erosion

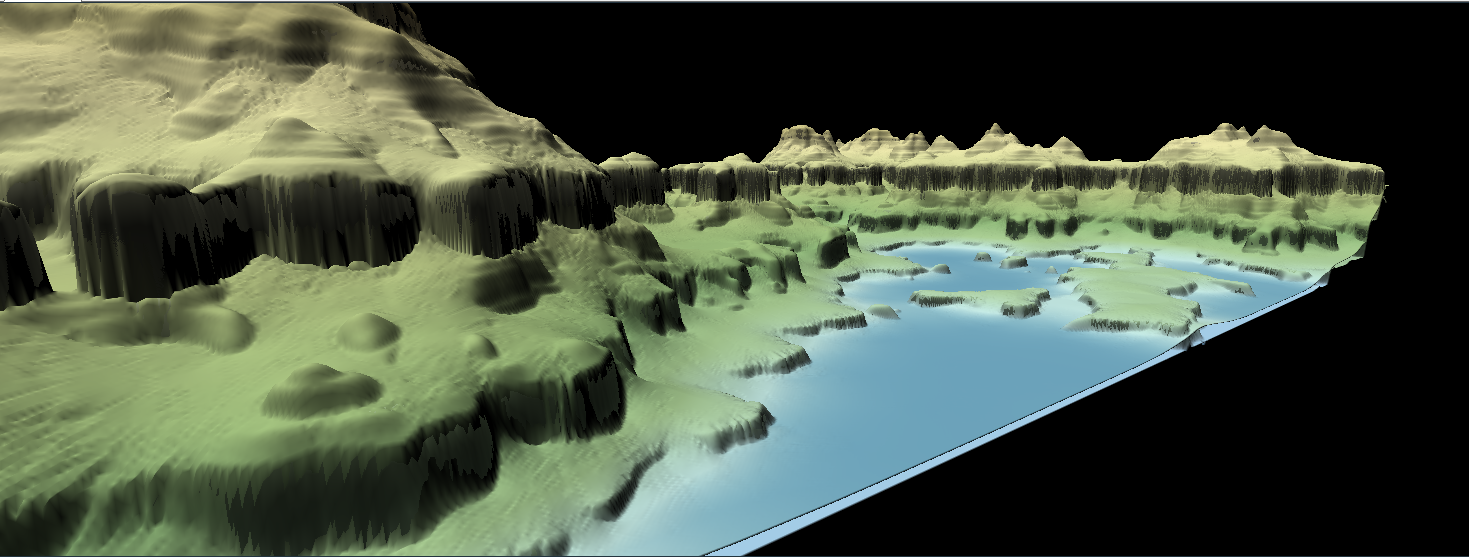


Figure - Instant Terra, terrain with hydraulic erosion

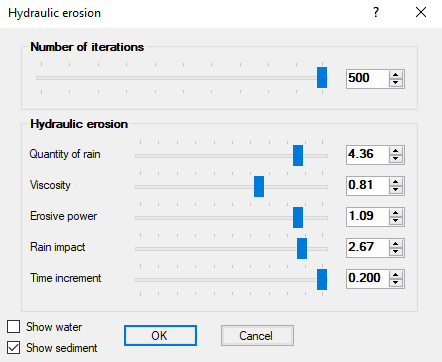


Figure - Instant Terra, hydraulic erosion settings

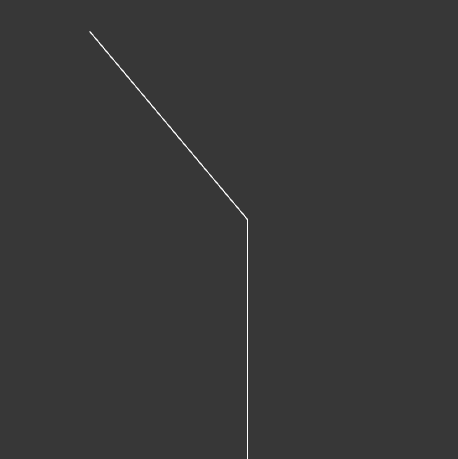


Figure 4 - Fractals Step 1

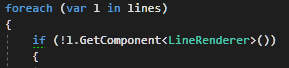


Figure 5 - Check if the line is already drawn

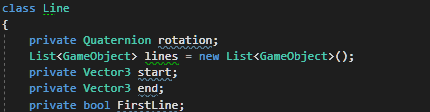


Figure 6 - Line Class Variables

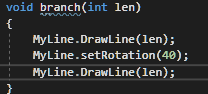


Figure 7 - Line Drawing

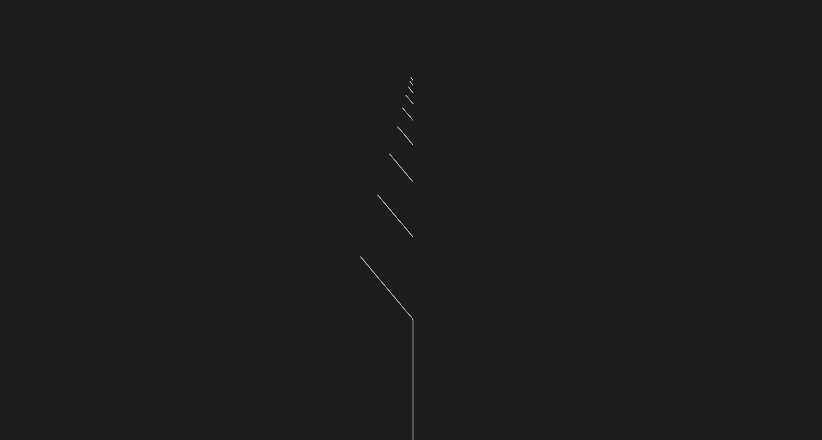


Figure 8 - Error when calling the function with recursion

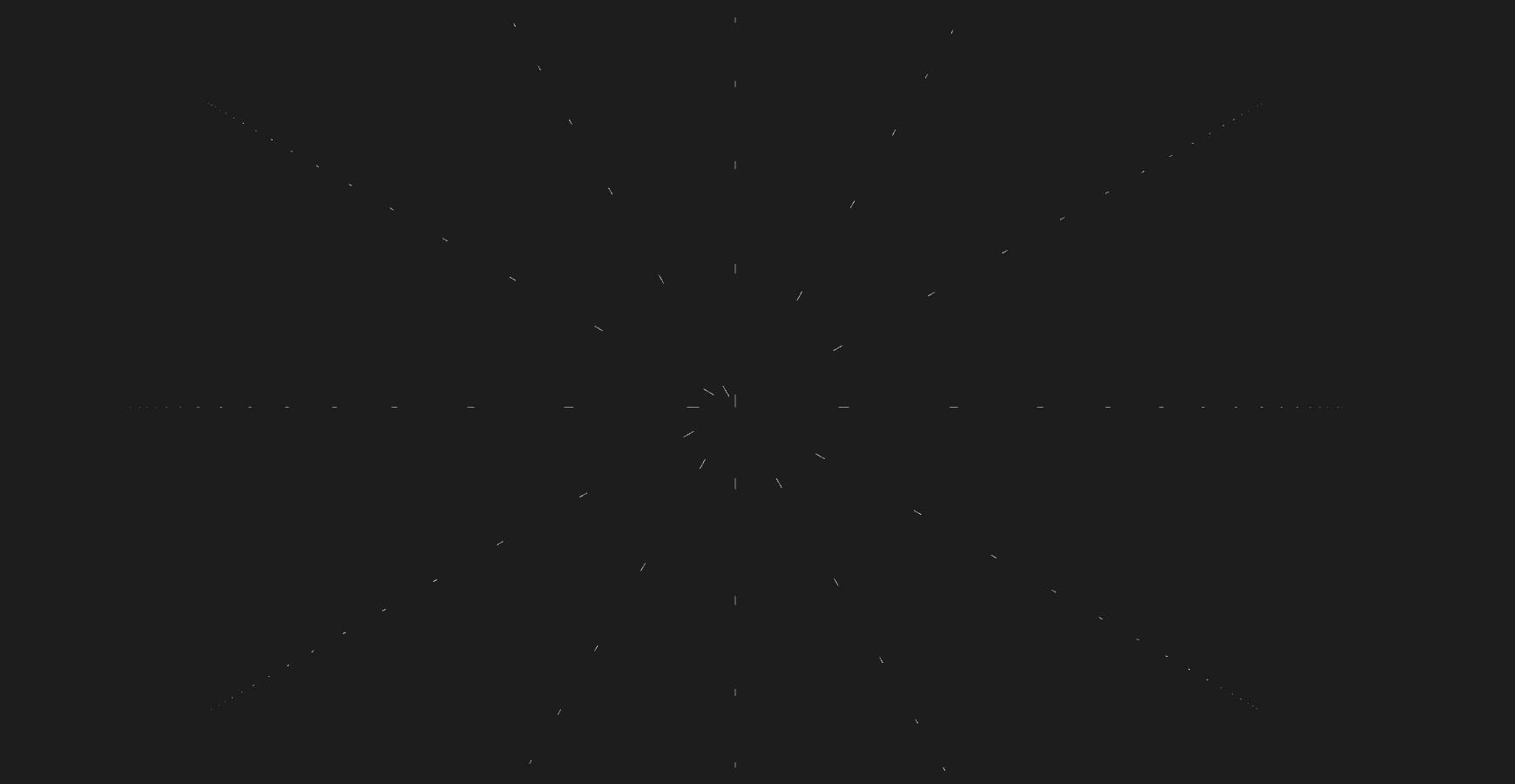


Figure - Random Fractal by Flavio Fiori

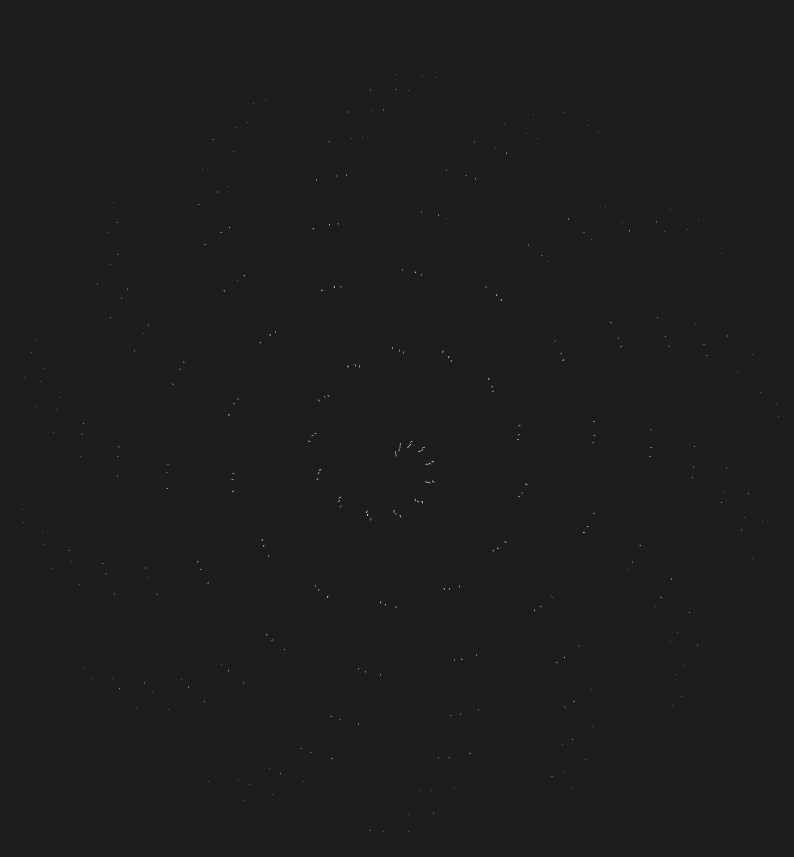


Figure - Random Pattern 2

# Sprint Review

# WBS

1. Research (40%) (22 hours)
2. Implement graphical user interface (10%) (5 hours)
3. Implement the level of detail functionality (50%) (27 hours)

# Reading List

Bottom of Form

Erosion Water Simulation-https://www.youtube.com/watch?v=xzVBFkpD94E

Instant Terra APP - <https://www.youtube.com/watch?v=jW3RpEHRyJs>

L-Systems <https://en.wikipedia.org/wiki/L-system>

<http://www.kevs3d.co.uk/dev/lsystems/>

<https://www.youtube.com/watch?v=f6ra024-ASY>

Drawing line with line Renderer <https://unity.grogansoft.com/drawing-lines-with-linerenderer/>

How to draw a line using a script <https://answers.unity.com/questions/8338/how-to-draw-a-line-using-script.html>

**Fractals** {

<https://www.youtube.com/watch?v=-wiverLQl1Q&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef>

<https://www.youtube.com/watch?v=s3Facu6ZVeA&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef&index=2>

<https://www.youtube.com/watch?v=_BOtJncHCVA&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef&index=3>

<https://www.youtube.com/watch?v=RWAcbV4X7C8&index=4&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef>

<https://www.youtube.com/watch?v=f6ra024-ASY&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef&index=5>

<https://www.youtube.com/watch?v=0jjeOYMjmDU&index=6&list=PLRqwX-V7Uu6bXUJvjnMWGU5SmjhI-OXef>

<https://www.youtube.com/watch?v=VXegg-HGT0s>

}

# References