# Sprint 3 – Finishing the Perlin Noise method and research in terrain synthesis techniques 21/02/19 – 26/02/19

# Abstract

The goal of this sprint is to finish the Perlin Noise class implementation, research ways to improve the terrain synthesis techniques, implementing some program design like the UML class diagram that describes the structure of the class.

# Research

During the research for improving the Perlin Noise algorithm a journal named Improving Noise by Ken Perlin (Perlin, 2002) explained two deficiencies present on the algorithm; one of those was the fade method, this method is responsible for easing the curve and the other one was the gradient calculation.

## Fade function

The first thing was improving the fade function by changing the formula, so the author replaced the method present in [Figure 1] to the new method present in [Figure 2].



Figure - Original Fade Function

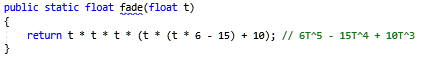


Figure - Fade function improved

Just by simple changing the fade function it is possible to perceive the increased level of detail on the right image, the sand is creating a better pattern and it is possible to perceive that now the water is present in places that before was just sand. This is due to the fact that now the values get even closer to integral coordinates, in other words closer to 1 and 0 [Figure 3].

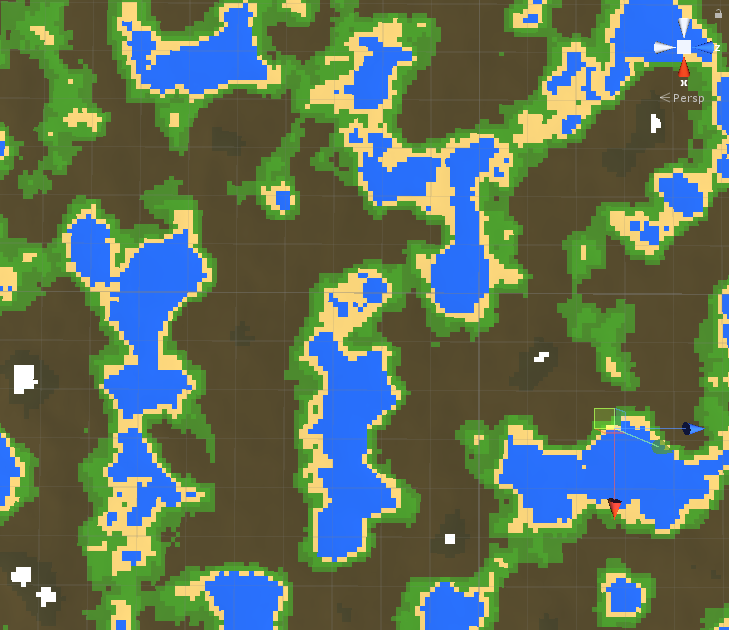
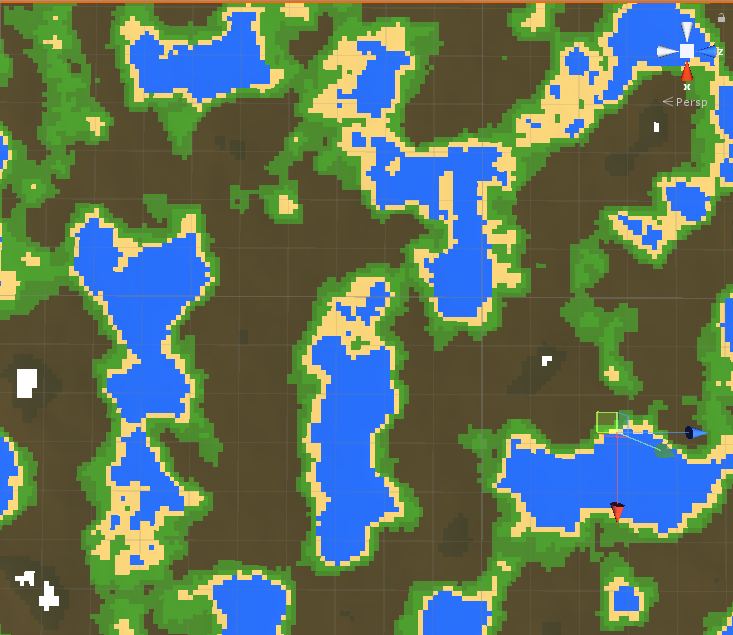


Figure - Original Fade Function left, Improved fade function on the right.

## Gradient Function

This function is used to calculate the dot product of a randomly selected gradient vector in 8 different locations. The original function [Figure 4] made by Ken Perlin’s uses complicated and confusing bit-flipping code to calculate the dot product and due to that fact, the author was not satisfied, he then decided to begin a research for an easy and more understandable solution.

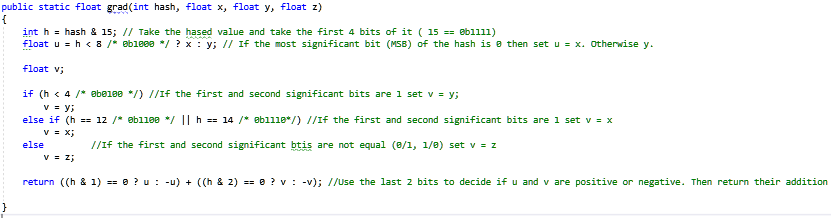


Figure - Gradient Original Function

After a few hours of researching the author found a great solution (Riven 2010) that seems to have the same effect from the original one, not only is easier to understand but is twice as fast from the original due to the fact that uses a switch statement and gets all the 16 possible outcomes from the gradient [Figure 5].

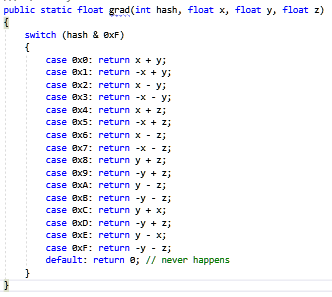


Figure - Simple Gradient Function

To prove that the function gets the same result in [Figure 6] is possible to see the two different functions side by side, and that the result is totally identical.

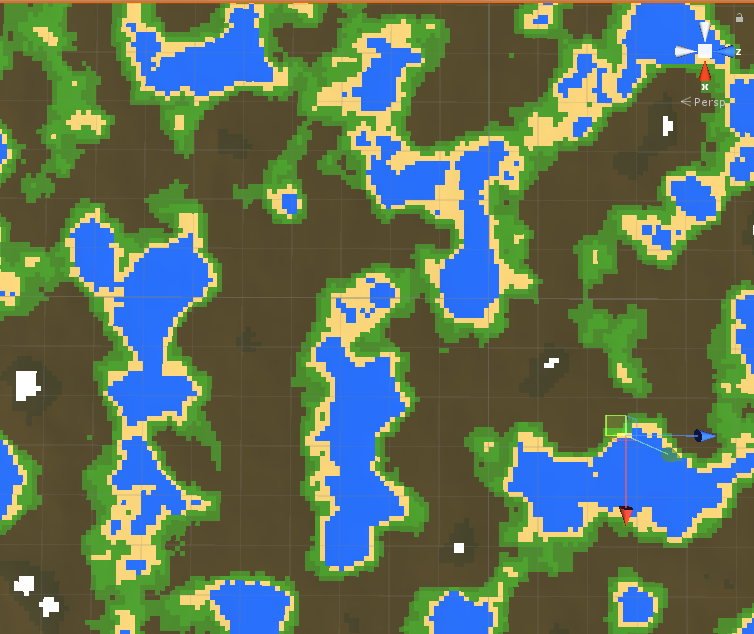


Figure - Original Gradient function | Simple Gradient Function

# Sprint Review

# WBS

1. Improve the Perlin Noise method (30%) (9 hours)
2. Create a class diagram (14%) (4.2 hours)
3. Create a flow chart (14%) (4.2 hours)
4. Implement a Colour Map using the noise map (14%) (4.2 hours)
5. Create a colour lerp method (14%) (4.2 hours)
6. Hydraulic Erosion (14%) (4.2 hours)