# Sprint 7 –hydraulic erosion 06/04/2019 – 01/05/2019

## Abstract

The goal of this sprint is to implement the hydraulic erosion algorithm to simulate real water erosion on the terrain mesh. Not only this is one of the largest sprints but also one of the most important ones. Optimisation techniques are discussed within this sprint.

The end goal of this sprint is the ability for the user control the quantity of erosion created by the water particles and a terrain mesh with visible erosion effects.

## Research

The algorithm that the author of this project created is inspired in the water erosion algorithm create by (Volynskov, 2011). It is a based particle simulation in which single water particles are placed into the map, where each particle has the capacity of eroding the terrain and deposit sediment.

### Algorithm

#### Water Particles

Each particle uses the gradient function to get a direction. If the gradient is zero, a random direction is generated.

The drops are simulated in 2.5D which means the drop is always considered to be at ground level, and only saves its position in two dimensions.

* Particles do not interact
* Do not use the physical principles
* Particles move the same distance every simulation step, although they are not bound to the grid.
* The simulated time per step is not consistent.
* Not suitable to simulate fluids visually.

##### Understanding Water Erosion

Water erosion is composed by two different steps, the first step is erosion and the second is deposition.

In the first step the flow of water starts its erosive activity when going through the land in a steep gradient. When the gradient starts to get a value near zero or the water flow reached the limited amount of soil that can carry then the water starts the deposition process. Normally it is possible to see the most amount of erosion when the flow of water it is at highest speed and the amount of water have a low quantity of sediment.

This amount of soil depends on the following variables:

* Surface Slope
* Amount of water
* Speed of the flow

Untitled picture.png 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Figure 1 - Water Flow

###### Erosion

* + If the quantity of soil is less than the maximum quantity of soil, then erosion happens.
  + This process removes soil from the terrain and adds it to the flow.

###### Deposition

* + If the water flow carries more than the maximum quantity then deposition happens, dropping extra carried soil as sediment.

##### Formula

* V = Water velocity
* W = Water amount
* Ch = constant value

### Erosion Class

#### Erosion method

When sketching the erosion class, the author wanted to create a particle inside the erosion method [Figure 2]. Each particle is created inside a for loop that checks for the number of iterations given by the user. Each particle consists on the following:

* Vector2 Position
* Vector2 direction
* Float Water Velocity
* Float Water amount
* Float Sediment

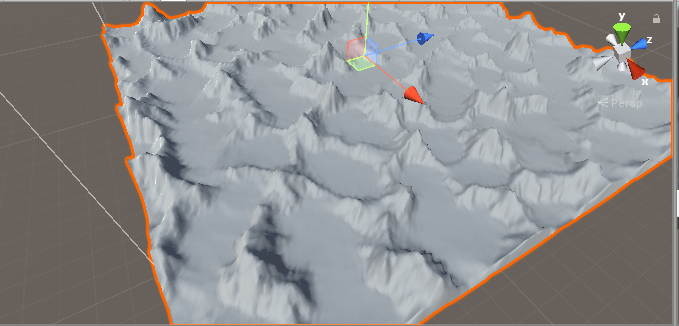


Figure 2 - Erode function Sketch

Multiplying the weighedErodeAmount for the amounToErode and a Random Range between 0.01f and Radius give a better end result.

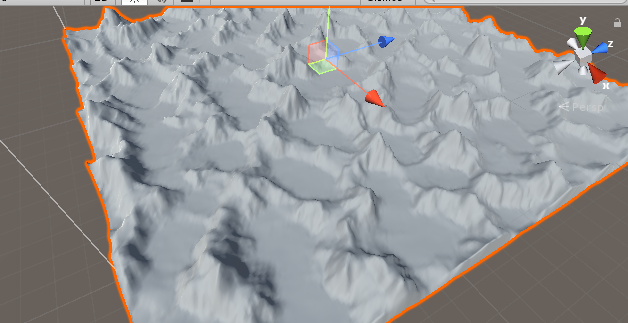
Small differences but they are noticeable and creates more a natural looking effect inserting more randomness.

Before





After





#### Gradient Method

To implement the gradient function the first thing was studying how to do linear interpolation, bilinear interpolation and contour maps [Figure 3].

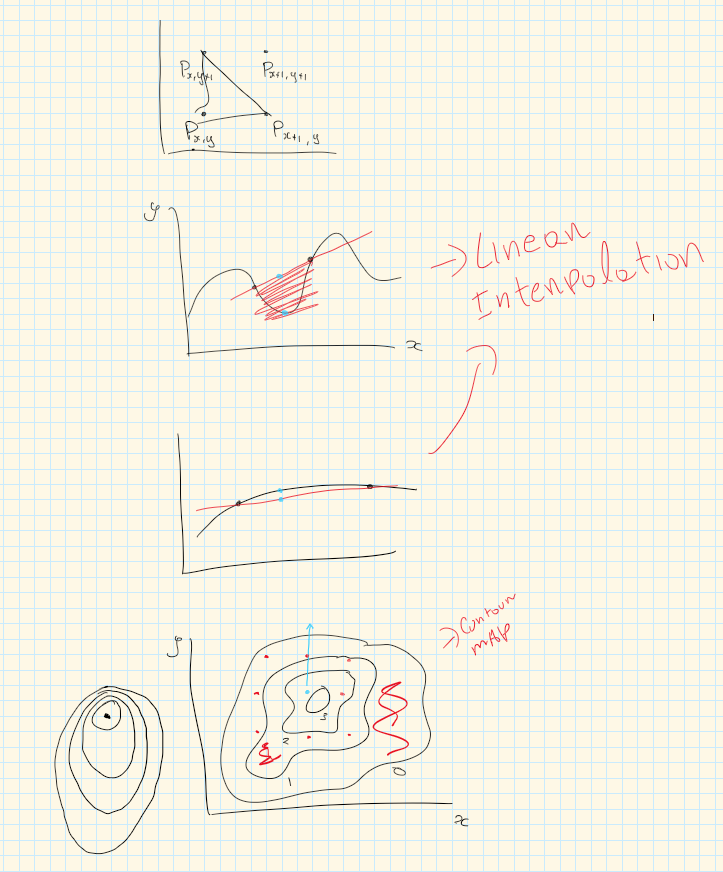


Figure 3 - Droplet gradient calculation sketch

Creating a bilinear interpolation from the four gradients from the map [Figure 2].

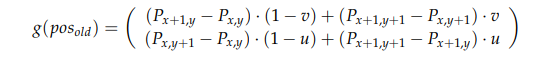


Figure 4 - Bilinear Interpolation for Gradient

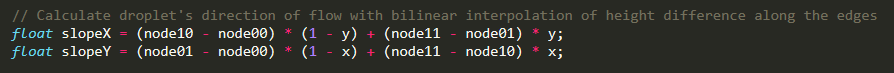


Figure 5 - Bilinear interpolation in code

Untitled picture.png 
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings
Ink Drawings


Figure 6- Gradient Calculation

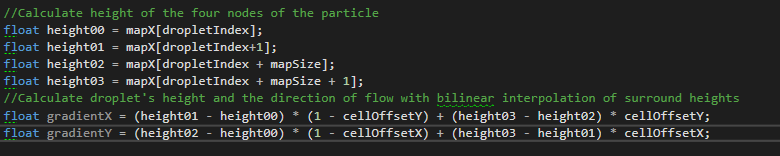


Figure 7 - Gradient Implementation in Code

#### UML Erosion Class Design

A screenshot of a cell phone

Description automatically generated

Figure 8 - UML Erosion Class

## Sprint Review

## WBS

1. Research (60%) (36 hours)
2. Fractal Generation (40%) (24 hours)

## Reading List

Bottom of Form

<http://www.improvedoutcomes.com/docs/WebSiteDocs/Clustering/Clustering_Parameters/Euclidean_and_Euclidean_Squared_Distance_Metrics.htm?fbclid=IwAR3tWg3O_7vWxwPjEzIh3PgmskyoE1PLcYJMNiqHBk741JvO5C7BExpEc2M>

<https://stackoverflow.com/questions/11555355/calculating-the-distance-between-2-points?fbclid=IwAR2ih_elI7nYonLhZ8zp2WVyDSUBMztaFtcorOe4toXT0NuJZZgrZu4uyKY>

<https://orion.math.iastate.edu/dept/links/formulas/form2.pdf?fbclid=IwAR3Bgom9hojgf4dxA8suf3uCRV-VLf3YgKKEliZmle0PEMX43098giW1p6M>

<https://www.khanacademy.org/computing/computer-programming/programming-natural-simulations/programming-vectors/a/vector-magnitude-normalization?fbclid=IwAR2-EO05GIcGXZ0b3o8eoQhL4RwQ1J_W839b3hxFaDW97tUbtO_pfDX7gTU>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/jagged-arrays>

<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/single-dimensional-arrays>

<https://en.wikipedia.org/wiki/Fractional_Brownian_motion>

<https://ranmantaru.com/blog/2011/10/08/water-erosion-on-heightmap-terrain/>

<http://web.mit.edu/cesium/Public/terrain.pdf>

<https://www.redblobgames.com/maps/terrain-from-noise/#trees>

<https://www.firespark.de/resources/downloads/implementation%20of%20a%20methode%20for%20hydraulic%20erosion.pdf>

<http://hpcg.purdue.edu/bbenes/papers/Stava08SCA.pdf>

<https://en.wikipedia.org/wiki/Contour_line>

<https://notch.tumblr.com/post/1409584546/some-work-on-biomes>

<http://pcg.wikidot.com/pcg-games:minecraft>

<https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/operators/arithmetic-operators#remainder-operator->

## References

## Bibliography