## Sprint 6 – Studying fractals and fractal implementation for future using in hydraulic erosion 22/03/2019 – 05/04/2019

### Abstract

This sprint is going to cover the process that the author went through when studying and creating the fractals. The reason why the author is studying fractals, lies on the fact that these geometrical figures are wildly found in nature. For example: For creating a river that go through the mountain is necessary a fractal design that creates different river nodes, where in the end these nodes create a river network, like the one in the image bellow [Figure 1] (Génevaux et al. 2013).

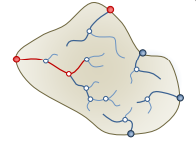


Figure - River Network, Example of fractal use

### Research / Implementation

#### What is a Fractal?

##### Introduction

The term fractal derived from the mathematician Benoit Mandelbrot in 1975. Mandelbrot (1983) defined a fractal as

“a rough or fragmented geometric shape that can be split into parts, each of which is (at least approximately) a reduced-size copy of the whole”

Fractals are a collection of distinct objects with non-integral Hausdorff dimension (JOHN E. HUTCHINSON 1981).

This dimension was introduced by a mathematician named Felix Hausdorff in 1918 and is a measure of roughness or smoothness (Tilmann Gneiting, Hana Ševčíková and Donald B. Percival 2012).

Different from the Euclidean geometry [Figure 2], fractals simulate the geometry of nature [Figure 3]

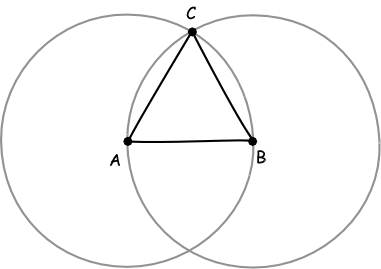


Figure - Euclidean geometry

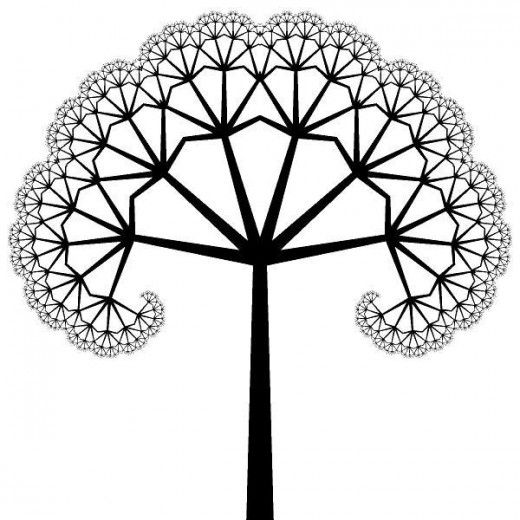


Figure - Fractal Tree

According to Falconer (1990) fractals should be only characterized by the following features:

##### Self-Similarity

Mandelbrot (1983) stated that each part of a fractal is a “reduced-size copy of the whole”. In the [Figure 4] it is possible to see the Sierpinski triangle, where not only every part of this fractal is repeated inside of itself but also the smallest section is an identic copy of the Sierpinski triangle, this is known as self-similarity.

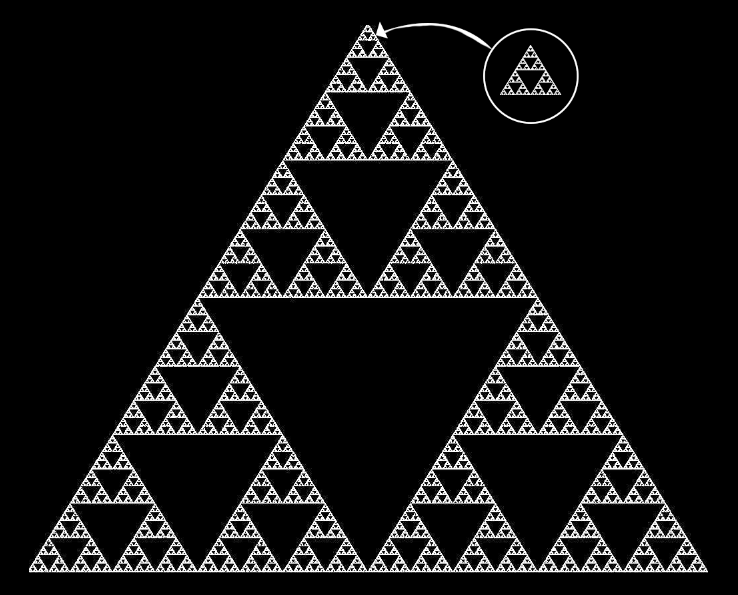


Figure - Sierpinski Triangle

In the example above it is possible to see a perfect self-similar fractal, however fractals do not have to be always perfectly self-similar.

It is possible to see in [Figure 5] the Mandelbrot set, this set produces similarly complex fractal shapes, even if they are similar, some of the shapes are distorted and degenerated, creating this random look.

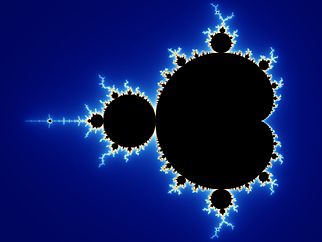


Figure - Mandelbrot set

##### Fine structure at small scales

This feature is essential in a fractal, making the fractal built out of probabilities and randomness, this is known as a stochastic fractal. While self-similarity is important to define a fractal, it cannot be defined only by it. For example, a line is self-similar, but it is not a fractal; fractals are characterized by having a fine structure at small scales, practically they cannot be described with Euclidean geometry.

A good example of a stochastic fractal is the stock market graph. In [Figure 6] and [Figure 7] is an example of two stock graphs with different timelines, even if they look practically the same, without a label around the graphs is impossible to tell the time scale. They can be over one year, one month or one day, so by zooming into the stock market graph is possible to find fluctuations and randomness. In this case [Figure 6] is one day and [Figure 7] One month.



Figure – Stock Graph 1



Figure – Stock Graph 2

##### Recursive

One important feature of a fractal it that it needs to be recursive. Every fractal has a recursive definition, practically it keeps growing and getting small for each recursion until it reaches a point it cannot shrink anymore, and that it is the exit point.

#### Coding a Fractal

After a quick search, I found that for drawing basic lines I needed to learn how to use the LineRender in unity (Unity School 2016) [Figure 4].

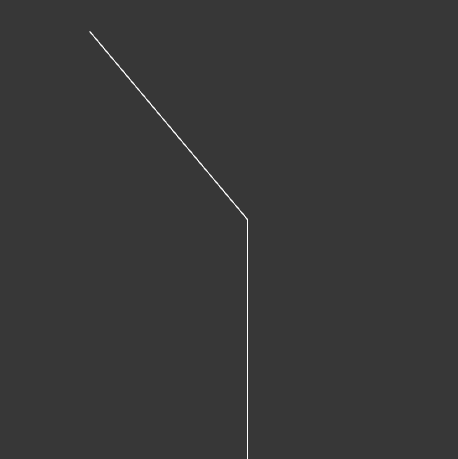


Figure 8 - Fractals Step 1

##### 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 5).

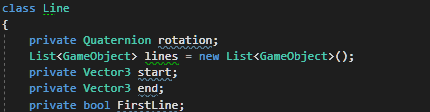


Figure 5 - Line Class Variables

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 6).

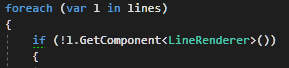


Figure 6 - Check if the line is already drawn

On the branch function 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) (Figure 7).

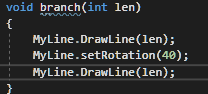


Figure 9 - Line Drawing

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.

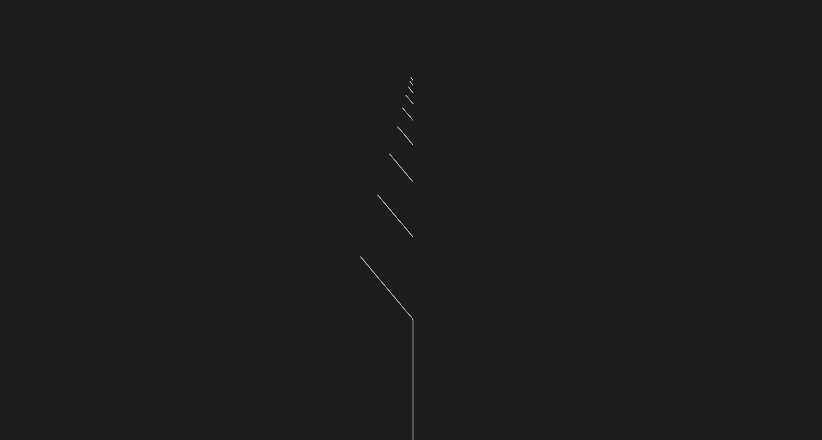


Figure 10 - Error when calling the function with recursion

The problem seems to be on the position for each line, to solve this problem I set the object position equals to the last object position multiplying by the last object rotation.

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.

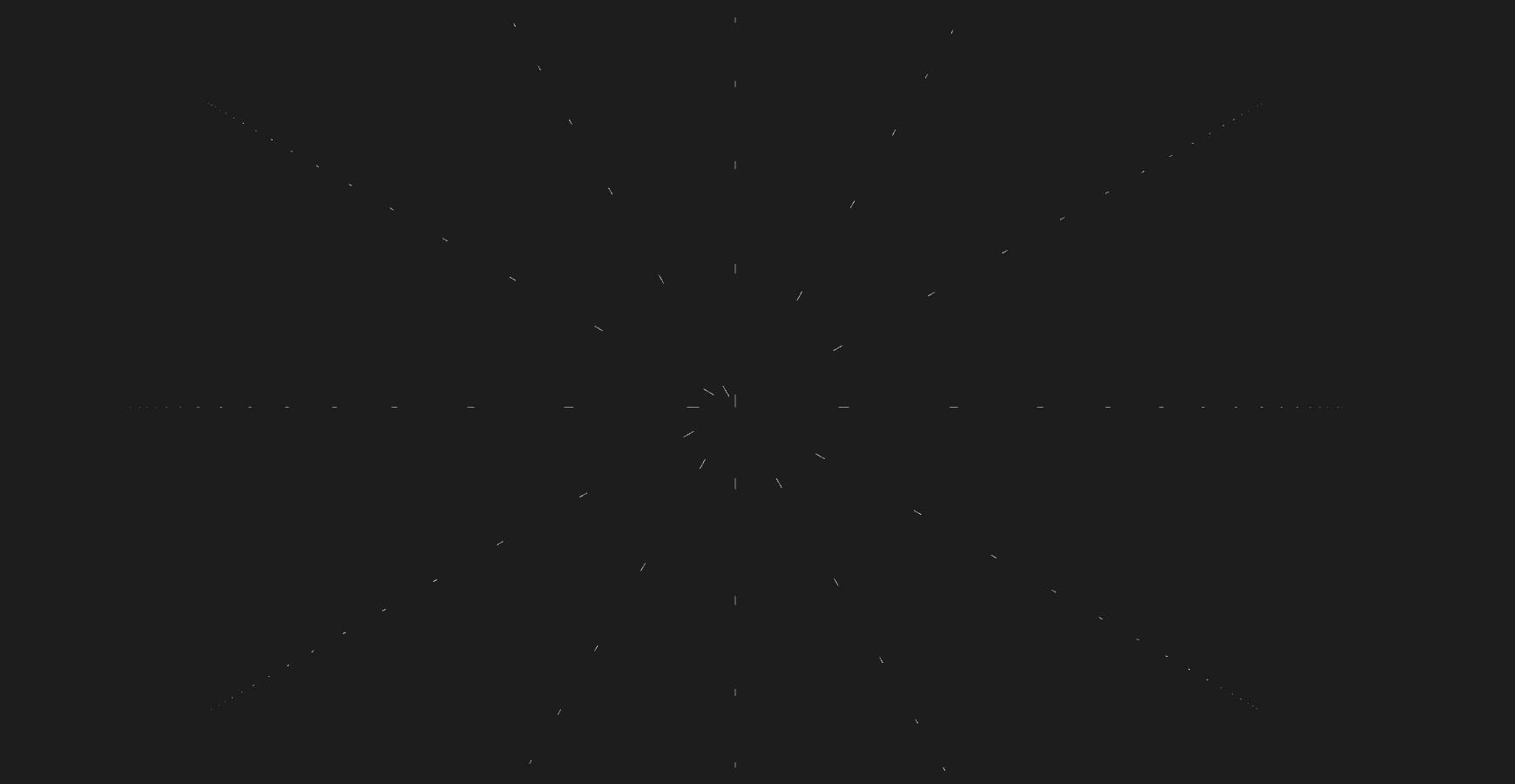


Figure 11 - Random Fractal by Flavio Fiori

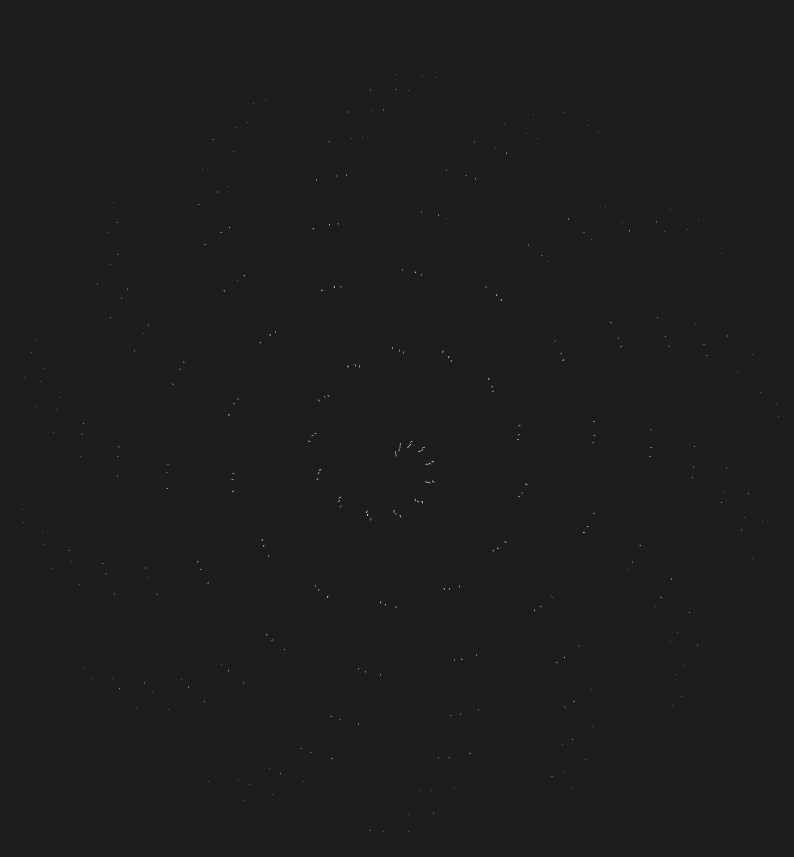


Figure 12 - Random Pattern 2

##### Fractal Tree

Not happy with the result from the previous attempt to generate fractals, I took a different approach.

I decided to create an empty game object that is going to have a cube has a child and create a script for the empty game object.

###### Fractals Script

This script is responsible for the recursion of the game object, and to achieve this result, different from the previous time where I was using a line renderer, now I just instantiate a copy of the game object itself [Figure 9].

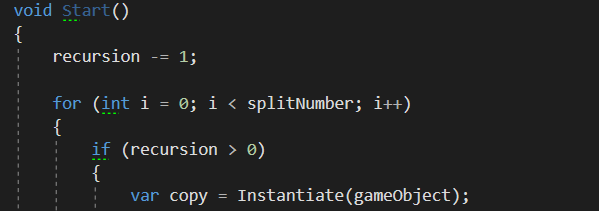


Figure - Recursion for the game object

When the copy variable is created with the game object, the goal is to create another variable to store the script where I am going to use a method named Send Message, this method allows to call any method in a class that derives from MonoBehaviour (Unity 2018) [Figure 10].



Figure 14 - Using Send Message Method()

###### Expander Script

This script contains the method that is called from the SendMessage() method that is used on the Fractal Script [Figure 10].

This method is named Created and have a RecursionScript has a parameter, this script is responsible to store the value from “i” in the for loop. For example, let’s say that the splitNumber value is 2, then the “i” can be 0 or 1. When the “i” is 1 then the rotation is positive when the “i” is 0 the rotation is negative due to the fact that multiplying 0 by \* 2 is 0 then subtracting 1 is -1. [Figure 11].

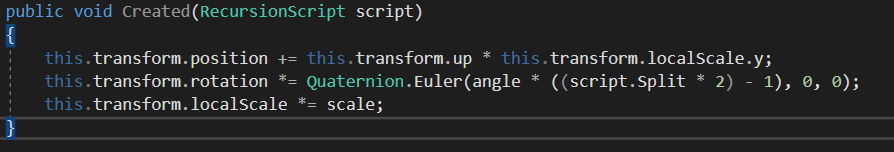


Figure 15 - Created Method

###### Recursion Script

This script is only responsible to hold the value from split and then use it to know the direction that is rotating.

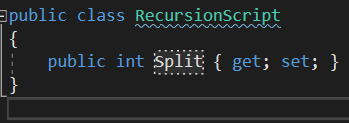


Figure 16 - Recursion Script

###### Conclusion

Now that the code was working has intended the result using the cubes started to look like a tree, I called this a fractal cube [Figure 13].

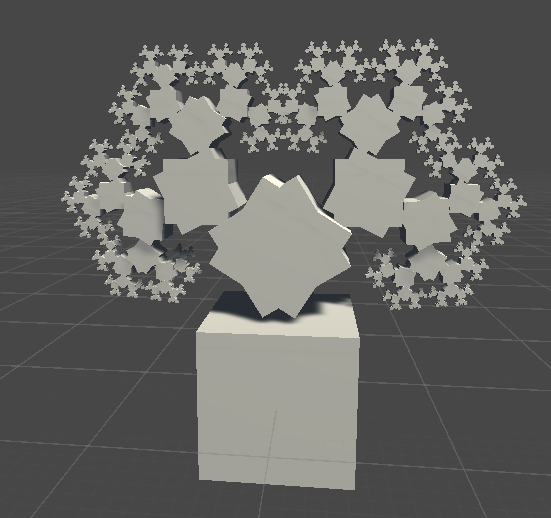


Figure 17 - Fractal Cube

I decided to rescale the cube to look alike a branch from a tree, using the cube scale has follow: x = 0.1, y = 1 and z = 0.1



Figure 18 - Tree Branch

Now that I created the tree branch, I compiled the code once again, the result was far from what I expected [Figure 15].

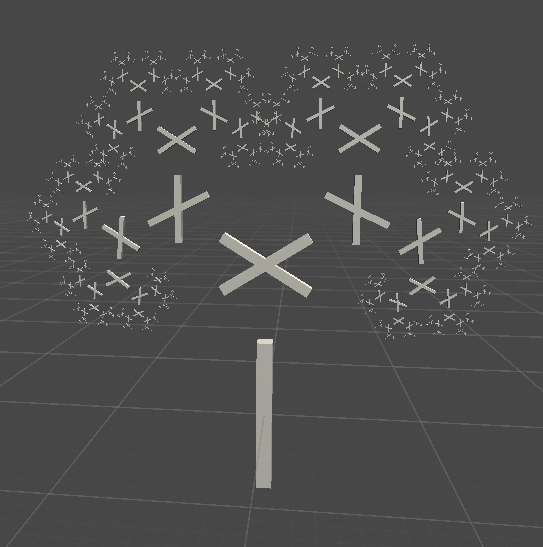


Figure 19 - Tree Generation Failed

After a while looking for a solution, I found that changing the position from the cube inside the parent was improving my results, I changed the Cube Position to Y = 1 [Figure 16].

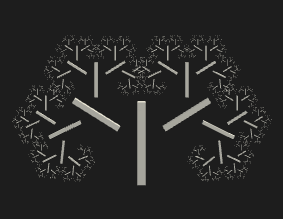
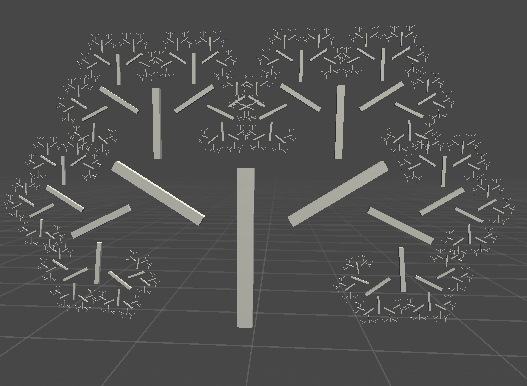


Figure 20 - Cube Y Position changed to 1

Not satisfied with the results, after playing for a while I found that the perfect Position for Y was 0.5, and the result was exactly what I was looking for, the fractal tree [Figure 17].

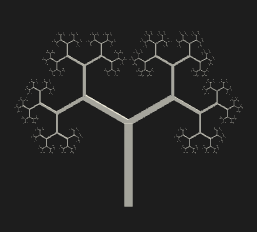
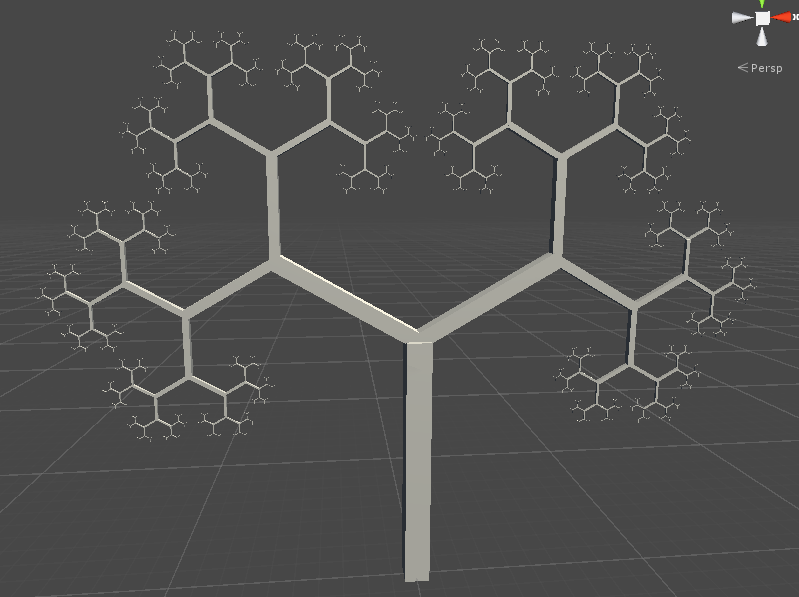


Figure 21 - Fractal tree

## Sprint Review

## WBS

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

## Reading List

Bottom of Form

BEYER, H., 2015. *Implementation of a method for hydraulic  
erosion*, TECHNISCHE UNIVERSITÄT MÜNCHEN

GÉNEVAUX, J.*et al.,* 2013. Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics (TOG),*32(4), 1-13

GRZEGORZ ROZENBERG and ARTO SALOMAA, 1980. *The Mathematical Theory of L Systems.*US: Academic Press

*Erosion by water,*2012 Directed by Kari KINKEY. Youtube: . 25 Oct,

ROAST, K., 2012. *L-Systems Turtle Graphics Renderer - HTML5 Canvas*[viewed 05/04/ 2019]. Available from: <http://www.kevs3d.co.uk/dev/lsystems/>

ŠT'AVA, O.*et al.,* Jul 7, 2008. Interactive terrain modeling using hydraulic erosion. Eurographics Association, pp.201-210

*8.1: Fractals - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.2: Fractal Recursion - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.3: Fractal Recursion with ArrayList of Objects (Koch Curve) - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.4: Recursion with Transformations - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.5: L-Systems - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*Coding Challenge #14: Fractal Trees - Recursive,*2016 Directed by The Coding Train. Youtube: . May 30,

UNITY, 2018. *GameObject.SendMessage*[viewed 05/04/ 2019]. Available from: <https://docs.unity3d.com/ScriptReference/GameObject.SendMessage.html>

UNITY SCHOOL, 2016. *Drawing Lines with LineRenderer*[viewed 01/04/ 2019]. Available from: <https://unity.grogansoft.com/drawing-lines-with-linerenderer/>

*Generating Fractal Trees in Unity  
,*2016 Directed by World of Zero. Youtube: . 26 Sep,

*How to simulate hydraulic erosion in Instant Terra,*2018 Directed by Wysilab. Youtube: . 7 Mar,

## References

GÉNEVAUX, J.*et al.,* 2013. Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics (TOG),*32(4), 1-13

UNITY, 2018. *GameObject.SendMessage*[viewed 05/04/ 2019]. Available from: <https://docs.unity3d.com/ScriptReference/GameObject.SendMessage.html>

## Bibliography

BEYER, H., 2015. *Implementation of a method for hydraulic  
erosion*, TECHNISCHE UNIVERSITÄT MÜNCHEN

GÉNEVAUX, J.*et al.,* 2013. Terrain generation using procedural models based on hydrology. *ACM Transactions on Graphics (TOG),*32(4), 1-13

GRZEGORZ ROZENBERG and ARTO SALOMAA, 1980. *The Mathematical Theory of L Systems.*US: Academic Press

*Erosion by water,*2012 Directed by Kari KINKEY. Youtube: . 25 Oct,

ROAST, K., 2012. *L-Systems Turtle Graphics Renderer - HTML5 Canvas*[viewed 05/04/ 2019]. Available from: <http://www.kevs3d.co.uk/dev/lsystems/>

ŠT'AVA, O.*et al.,* Jul 7, 2008. Interactive terrain modeling using hydraulic erosion. Eurographics Association, pp.201-210

*8.1: Fractals - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.2: Fractal Recursion - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.3: Fractal Recursion with ArrayList of Objects (Koch Curve) - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.4: Recursion with Transformations - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*8.5: L-Systems - The Nature of Code,*2015 Directed by The Coding Train. Youtube: . 11 Aug,

*Coding Challenge #14: Fractal Trees - Recursive,*2016 Directed by The Coding Train. Youtube: . May 30,

UNITY, 2018. *GameObject.SendMessage*[viewed 05/04/ 2019]. Available from: <https://docs.unity3d.com/ScriptReference/GameObject.SendMessage.html>

UNITY SCHOOL, 2016. *Drawing Lines with LineRenderer*[viewed 01/04/ 2019]. Available from: <https://unity.grogansoft.com/drawing-lines-with-linerenderer/>

*Generating Fractal Trees in Unity  
,*2016 Directed by World of Zero. Youtube: . 26 Sep,

*How to simulate hydraulic erosion in Instant Terra,*2018 Directed by Wysilab. Youtube: . 7 Mar,