## 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).

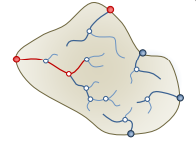


Figure - Génevaux et al. 2013. Fractal Network [digital image] [viewed 22 March 2019]. Available from:

<https://www.researchgate.net/profile/Eric_Guerin2/publication/248703095_Terrain_Generation_Using_Procedural_Models_Based_on_Hydrology/links/00b7d53c52bcd657e3000000.pdf>

### 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]

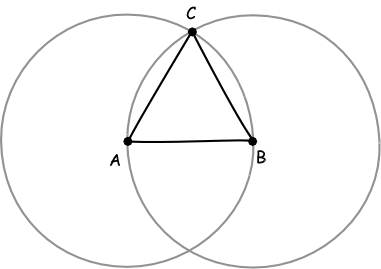


Figure - NORTON, J., 2006. Euclidean Geometry [digital image] [viewed 22 March 2019]. Available from: https://www.pitt.edu/~jdnorton/teaching/HPS\_0410/chapters/non\_Euclid\_fifth\_postulate/index.html

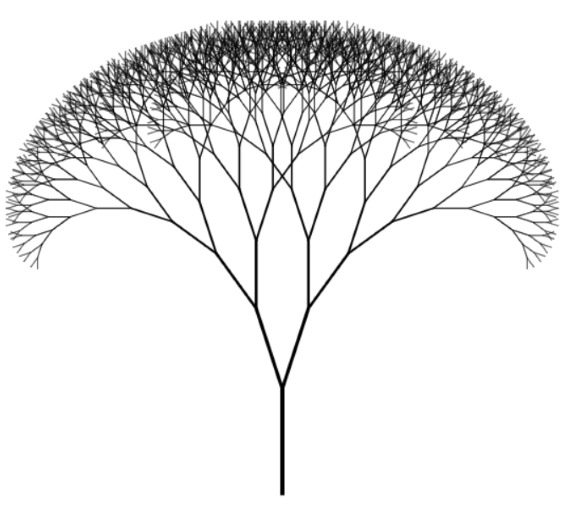


Figure - Seemann, M., 2017. Fractal Tree [digital image] [viewed 22 March 2019]. Available from: https://blog.ploeh.dk/2017/06/06/fractal-trees-with-purescript/

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.

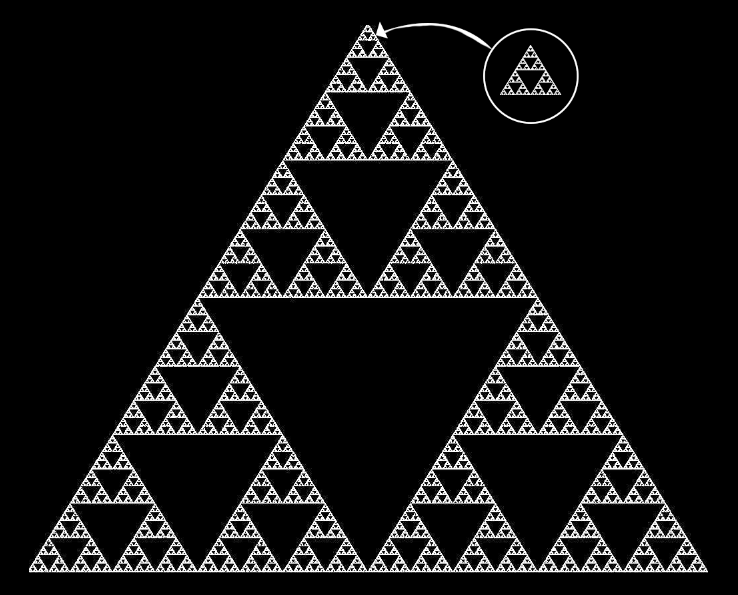


Figure - Sierpinski Triangle

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.

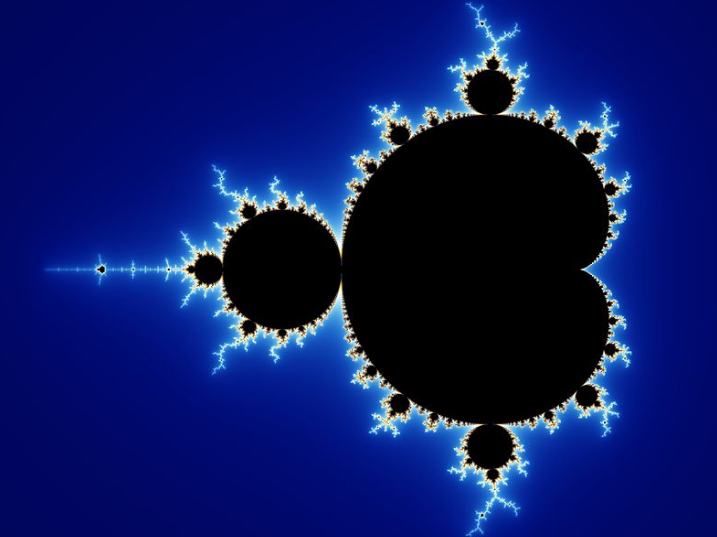


Figure - Beyer, W., 2013. Mandelbrot set [digital image] [viewed 22 March 2019]. Available from: <https://en.wikipedia.org/wiki/Mandelbrot_set#/media/File:Mandel_zoom_00_mandelbrot_set.jpg>

##### 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 from Apple, 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 year and [Figure 7] one month.



Figure - Apple Inc., 2019. Market Summary of 1 Year [digital image] [viewed 28 March 2019]. Available from:

https://www.google.com/search?safe=off&sa=X&rlz=1C1CHBF\_enPT776PT776&tbm=fin&q=NASDAQ:+AAPL&stick=H4sIAAAAAAAAAONgecRoyi3w8sc9YSmdSWtOXmNU4-IKzsgvd80rySy



Figure -Apple Inc., 2019. Market Summary of 1 Month [digital image] [viewed 28 March 2019]. Available from:

https://www.google.com/search?safe=off&sa=X&rlz=1C1CHBF\_enPT776PT776&tbm=fin&q=NASDAQ:+AAPL&stick=H4sIAAAAAAAAAONgecRoyi3w8sc9YSmdSWtOXmNU4-IKzsgvd80rySy

##### 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

So, the author decided that for creating a fractal he needed to learn how to draw a line in unity. After a quick search, he found that for drawing basic lines, learning how to use the LineRender in unity was necessary (Unity School 2016).

##### Line Class

The author started creating a Line Class, this class have a method for Drawing a line, each time this method is called, it creates a game object and it stores it inside a list. The struggle here was not only drawing the line and being able to draw the next line saving the position from the line before but also apply a specific rotation if needed. To be able to do this the author created two vector 3, a quaternion and a Boolean variable [Figure 8].

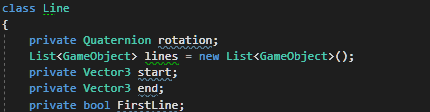


Figure - Line Class Variables

Afterwards he 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 it was necessary to find a way to check if the line was not drawn yet, for this an if statement was implemented for checking if the game object had the line Renderer component, if not then draw the next line [Figure 9].

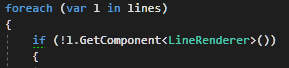


Figure - Check if the line is already drawn

When compiling the code, it is possible to see that the lines are created upon the length variable defined in the parameter of the branch method [Figure 10]. Then the next lines of code are responsible for setting the rotation and redrawing another line. This creates the result that can be seen in [Figure 11].

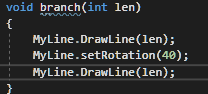


Figure - Fractals Step 1

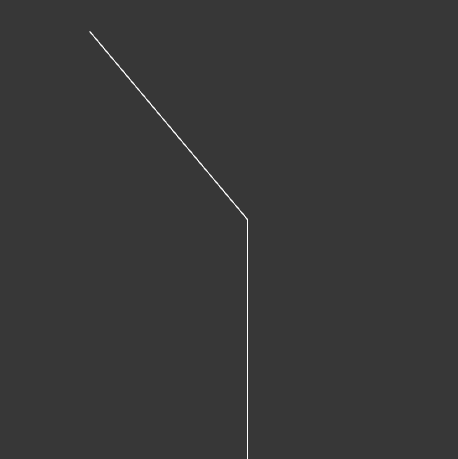


Figure - Line Drawing

When the function is called recursively, a strange problem appears [Figure 12], it seems that the lines are not being translated in a proper way. For this it is necessary to find the exact position where the line before ends and pass it to the new line starting position.

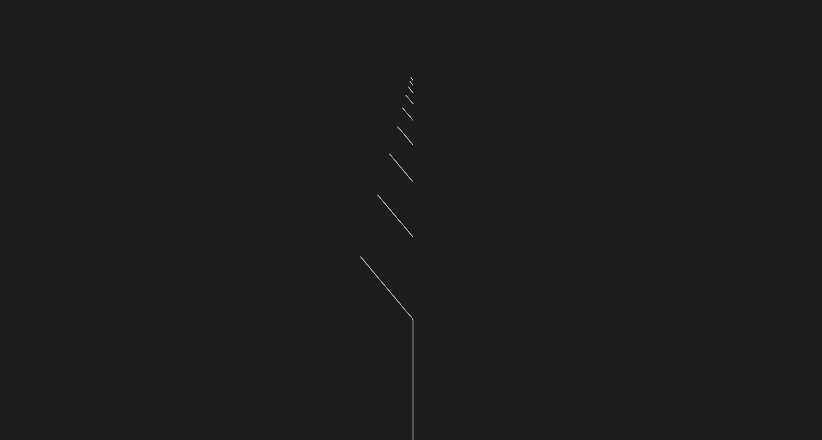


Figure - Error when calling the function with recursion

The problem was found on the position for each line, the solution was setting the object position equals to the last object position and then multiplying by the last object rotation.

After many hours spent on the attempt to create a fractal, the author ended up with two interesting fractal patterns using only lines in unity [Figure 13] [Figure 14]. The end goal was to be able to create different patterns, for example a tree. This technique is going to allow the drawing of water paths when developing the hydraulic erosion algorithm.

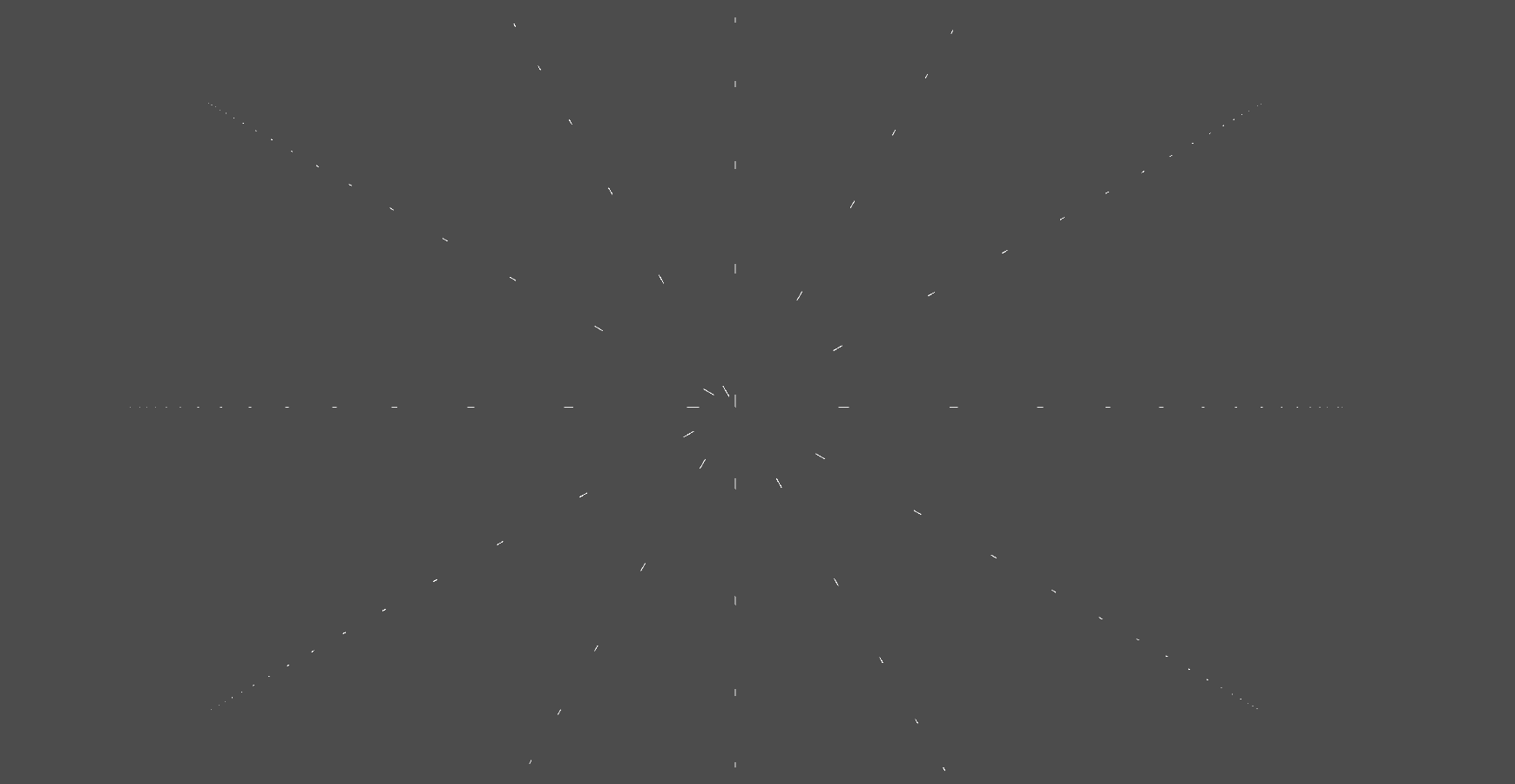


Figure - Random Fractal by Flavio Fiori

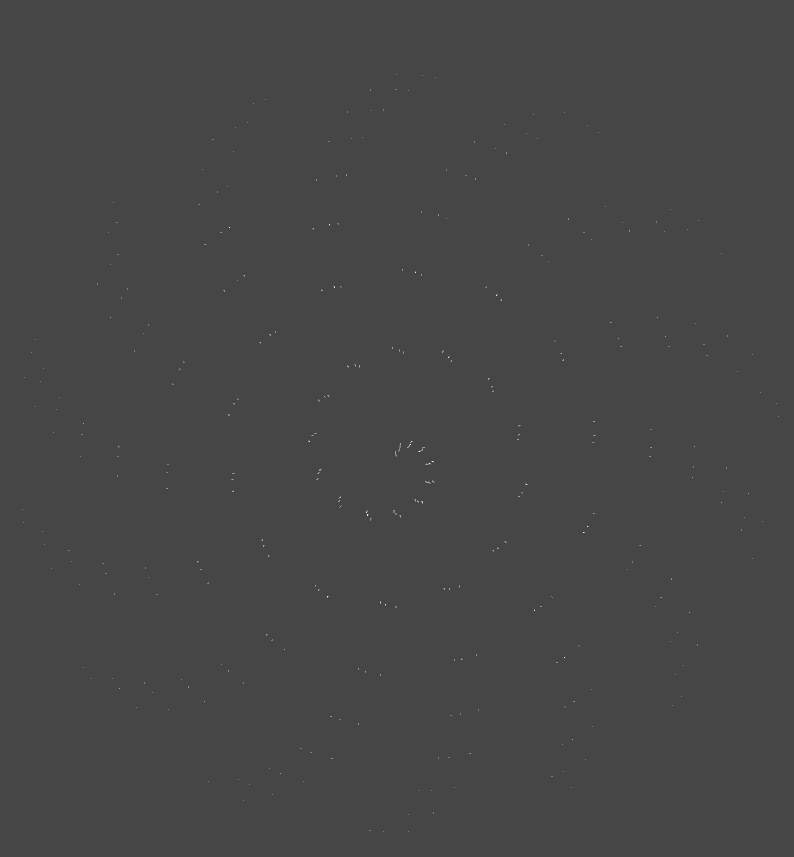


Figure - Random Pattern 2

##### Fractal Tree

###### Introduction

A fractal tree is created by calling reclusively a method to creates a branch of the tree itself.

Frongillo, Lock and Brown (2007) defined the fractal tree as the following:

“A fractal tree can be loosely defined as a trunk and a number of branches that each look like the tree itself, thus creating a self-similar object. Often, these appear strikingly similar to real trees, and hence are used frequently as tree models.”

To be able to create the fractal tree the author decided to split the creation in three classes, the Fractal, Expander and Recursion script.

###### Fractals Script

The first idea when creating the fractal script was to think in how to approach the recursion of the object. The author decided to remove 1 value from the recursion variable when the script starts, so this means if the value of recursion is 1 the code is not going to run, the reason why, lies on the fact that there is already one object on the scene, so to see any visible results it is necessary at least two recursions.

For each value of recursion bigger than 1, a copy of this game object is instantiated and saved inside a variable. [Figure 15].

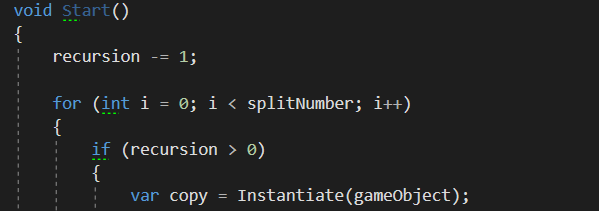


Figure - Recursion for the game object

The second step is to get the script named fractals from the game object and create a new class named Recursion [Figure 16].

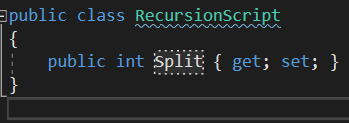


Figure - Recursion Script

This class is going to be only responsible to store the value from “i” in the for loop [Figure 15].

The last step is to use a function named Send Message, this function allows to call inside the same solution any method declared in a class that derives from MonoBehaviour (Unity 2018) [Figure 17].



Figure - 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.

Let’s imagine that the variable defined in the Fractal Script named “splitNumber” as a value of 2, then the Split value can be 0 or 1. When the value is 1 then the rotation is positive, otherwise if is 0, the rotation is negative. The reason lies on the fact that multiplying any number by 0 by is always 0 then subtracting 1, the end result is -1 [Figure 18].

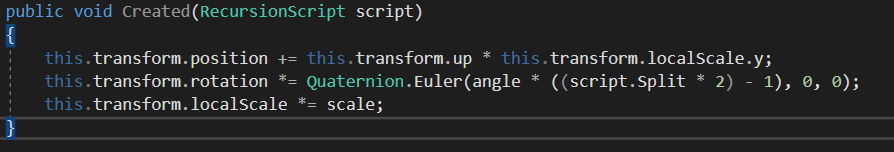


Figure - Created Method

###### Conclusion

The author ran the code in a cube, and a strange tree made of cubes was generated [Figure 19].

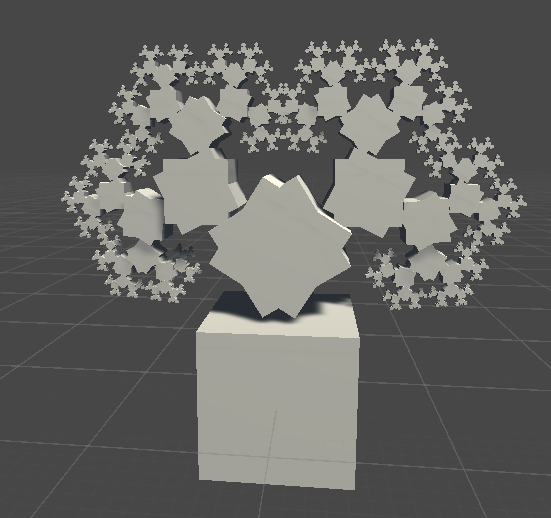


Figure - Fractal Cube

To improve the look he decided to rescale the cube to look alike a branch from a tree [Figure 20], the cube scale was defined as follow: x = 0.1, y = 1 and z = 0.1



Figure - Tree Branch

When the code was compiled once again, the shape was similar to a tree but the result was far from what the author expected [Figure 21].

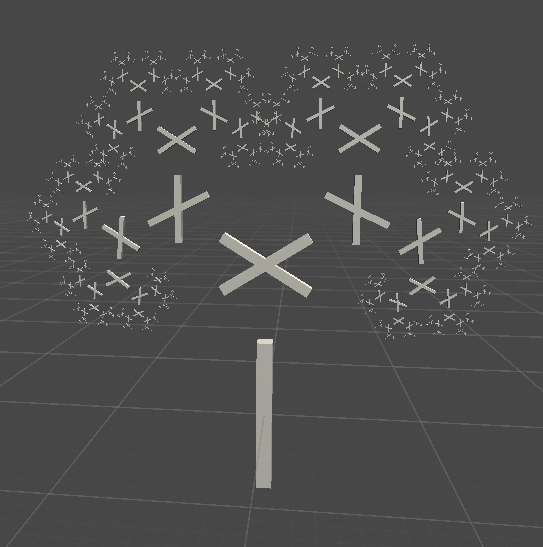


Figure - Tree Generation Failed

After a while looking for a solution, it was found that changing the position from the cube that was present inside the parent it improved the results in a drastic manner, the Cube Position value was set to Y = 1 [Figure 22].

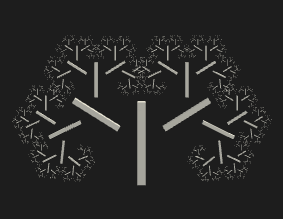
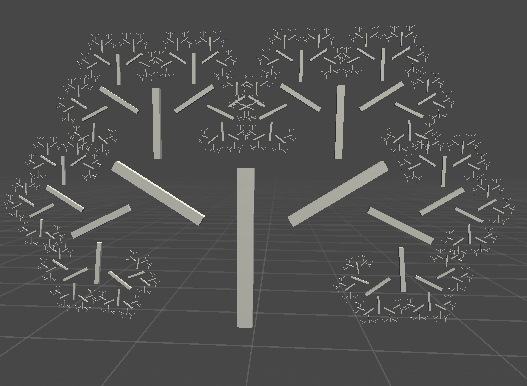


Figure - Cube Y Position changed to 1

The final result was found after playing for a while with the Y Position, the perfect Y value is 0.5f, resulting exactly in the fractal tree [Figure 23].

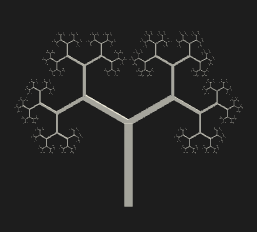
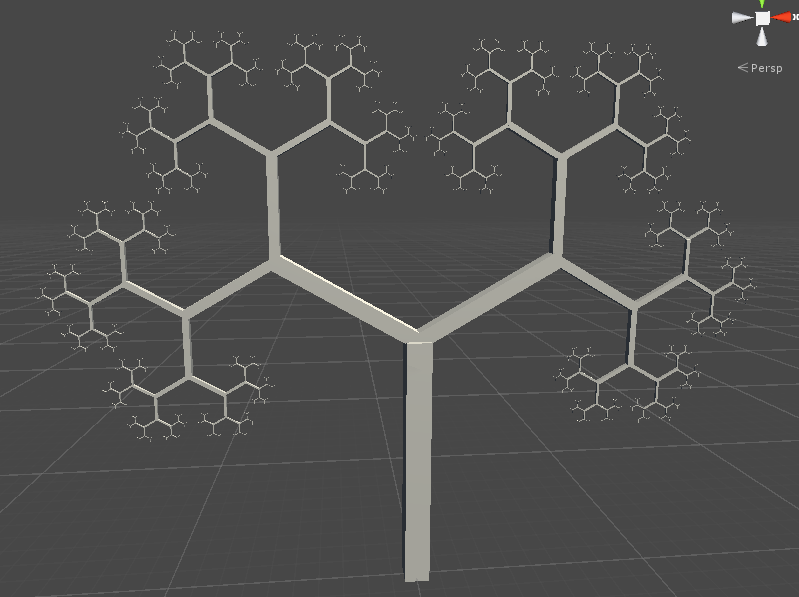


Figure - Fractal tree

## Sprint Review

This sprint was an overall success, not only it gave the necessary knowledge to the author for developing water paths inside the mountains using fractals but also to implement some trees in the map generation if time allows.

## 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

FALCONER, K.J., 1952, 1990. *Fractal geometry : mathematical foundations and applications.*England:

FRONGILLO, R.M., E. LOCK and D.A. BROWN, 2007. Symmetric fractal trees in three dimensions. *Chaos, Solitons and Fractals,*32(2), 284-295

JOHN E. HUTCHINSON, 1981. Fractals and Self Similarity. *Indiana University Mathematics Journal,*30(5), 713-747

MANDELBROT, B.B., 1983. *The fractal geometry of nature.*Updated and augm ed. New York: Freeman

TILMANN GNEITING, HANA ŠEVČÍKOVÁ and DONALD B. PERCIVAL, 2012. Estimators of Fractal Dimension: Assessing the Roughness of Time Series and Spatial Data. *Statistical Science,*27(2), 247-277

## Bibliography

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

FALCONER, K.J., 1952, 1990. *Fractal geometry : mathematical foundations and applications.*England:

FRONGILLO, R.M., E. LOCK and D.A. BROWN, 2007. Symmetric fractal trees in three dimensions. *Chaos, Solitons and Fractals,*32(2), 284-295

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

JOHN E. HUTCHINSON, 1981. Fractals and Self Similarity. *Indiana University Mathematics Journal,*30(5), 713-747

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

MANDELBROT, B.B., 1983. *The fractal geometry of nature.*Updated and augm ed. New York: Freeman

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,

TILMANN GNEITING, HANA ŠEVČÍKOVÁ and DONALD B. PERCIVAL, 2012. Estimators of Fractal Dimension: Assessing the Roughness of Time Series and Spatial Data. *Statistical Science,*27(2), 247-277

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,