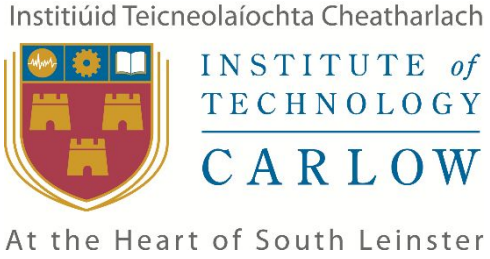


Computer Games Development
CW208

SRS and Project Report

Year IV

Brian O'Neill

C00213756

03/05/2020

Contents

Acknowledgements

I would like to thank the following people who provided me with great assistance before and during my project: over the years, my lecturers have given me the skills to help me tackle this research problem.

I would also like to thank my supervisor Noel O'Hara for offering outstanding assistance, and for his motivation and helpfulness during the year.

Project Abstract

Motivation:

Although Procedural Generation has been around for many years, it is nowadays a particularly useful tool for indie developers and other big name developers. With distribution platforms like Steam, Epic Games launcher, Play Store and Apple's IOS Store, it's now easier to create and publish your games than ever. But the challenge for small developers is no longer distribution, the challenge is content these days. Even small indie developers can produce a vast amount of content through the use of procedural generation, with reasonably low amounts of money and manpower invested. For instance Minecraft is a good example of a small developer making it big by using procedural generation. When it was published the game took the world by storm and it hasn't slowed down since. The development team sold the IP to Microsoft for $2.5 billion dollars[Microsoft n.d]. Since then, many games from larger and smaller developers have started to incorporate these techniques into their games too. Some developers are creating whole games from procedural generation at this point and some are doing very well with this but these techniques are constantly developing and will potentially become so much more than we can imagine right now.

This is what inspired me to pursue a procedural generation thesis, as I find the entire field incredibly exciting.

My Aim:

My goal for this project is to develop an editor in which the player can use procedural generation techniques to create a piece of a game world, including a road network using a directional tool, buildings, vegetation and rocks. Showing how procedural generation will allow you to build varying game assets each time and also showing how small teams of developers may be able to use this object generation method to build highly replayable games.

Method:

I approached this topic by deciding on which engine I wanted to use.I was mainly stuck between two engines either Unreal or Unity. After researching Unreal Engine, I had learned that Unreal Engine was a fantastic engine for studios and teams of people but as I am working alone on this project I decided to use Unity which is a much more solo developer friendly engine. So using this information I decided to use unity for visualizing my project.

After that I decided to look at games that used these techniques and after looking through several different games and genres I found that many games were using procedural techniques to build their game terrains. With this knowledge, I explored more ways of using procedural generation for other parts of a game world. Since it seems to be a very common asset to use procedural generation in.

As I continued other games I realized that SimCity creates its roads using procedural methods, Creating roads is definitely done less in the industry versus creating terrains. Ithen began to think how I would go about creating something like this in my project. I had created

a test for this by making a very simple way of making a curve and then adding points to the end of it to try to simulate the method used in SimCity, However I ultimately scrapped this idea as I began to see through the use of this method that it is not the feel I was looking interested in for my project. However I kept one of the techniques used and that one was the directional tool for creating the road in a particular direction.

After this testing and seeing the way it felt using that method, I decided to use a tile-based method and upon testing this is exactly what I was looking for.

Then I started looking at other possible items that I could bring into the scene to enhance the look and overall appearance of the environment and I came to the conclusion after studying various technologies that are commonly used in industry. One of the technologies I came across was SpeedTree, which is a well known tree generator and I decided that trees would be the perfect addition. AS the more objects that I can create and the more diversity in my world the better.

So I began looking into possible methods I could use and one jumped out at me and after further research into it I realised using this same method I could create grass and this was perfect, So I ultimately decided to pursue this method.

However the placement of the trees around the map would be very important as you don't want them to just spawn randomly throughout the scene and so I to look into possible solutions to this and one method I found was Poisson Disk Sampling which seemed to me to be an effective method at solving the case that I had previously mentioned.

I had learned in previous college years that perlin noise was a procedural method used to create believable terrains. I wanted to implement this inside my editor but I did not want to use it for a map so I decided that maybe using it to create an object would work well. So I began to think which objects would work well with this method and one that would be possible for me to use came to mind and that object was rocks since they can be jagged and are very rarely smooth this jaggedness could work well with perlin noise.

Then I decided that to add to the assets I have already mentioned I believe that creating buildings alongside the road as you move the road building tool would be a worthwhile addition to the project. This allows you to create an interesting stretch of road with all these assets being created around it.

I strongly believe that with these I can show just how procedural generation of game assets is a worthwhile practice in video games.

Project Introduction

I'm developing an editor using Unity3D, My Editor will allow the player to create a small area of a game world which will be populated using both player input and also using procedurally generation to decide where to build and also how to build each of the assets.

The player input will affect things like the direction of the roads, the size and age of the trees and the spacing and density of the grass.

The reason I chose to create this project using Unity was because I really wanted to get some exposure to creating in 3D as over the last few years I have not been exposed to it a lot. The reason I chose this research topic was due to my own interest on how games such as minecraft,diablo and borderlands are able to keep a loyal audience even after 10 or more years and still surprise players by the sheer number of possibilities that are possible within these games.

I find it fascinating that even after such a long time of playing a game that new possibilities are still yet to be discovered due to their procedural generation of game content. This is the reason I believe that my project is an important example of how you can use procedural generation to create varied areas of a game world to produce interesting and varied areas everytime it is generated.

The potential impact of procedural generation is significant. It can be used industry wide to reduce costs and lower development time drastically. Also the demand for the product will increase as with the amount of content that can be generated is basically infinite. This can be done by separating the world into chunks like minecraft where each time a new chunk is loaded it's completely new as long as that chunk has not been loaded before.

My goal for the outcome of this project is to create a tool that allows for the creation of small pieces of a game world with an editor that has integrated procedural generation techniques.

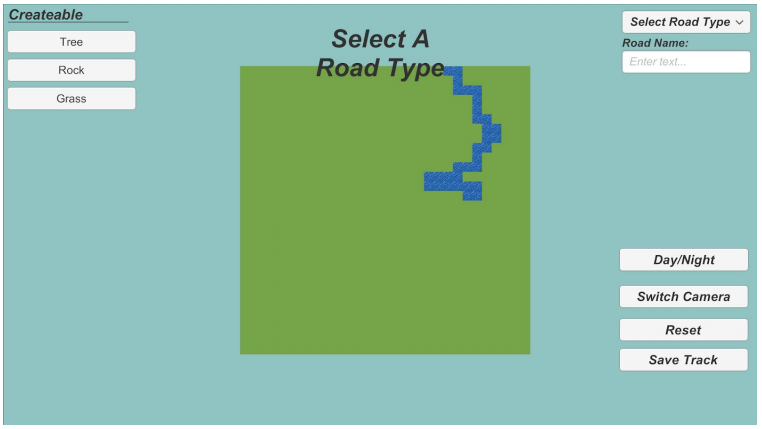
Background

Procedural Generation is the technique of creating content mostly for use in some form of media. The content is created with the use of algorithms rather than a designer or someone of a similar job role creating these assets and laying out the world by hand.These algorithms are created using a mixture of both randomized values and other values that are controlled by the user of these algorithms, this allows for enough freedom for the user to create a near infinite number of assets that all have some degree of difference between them.

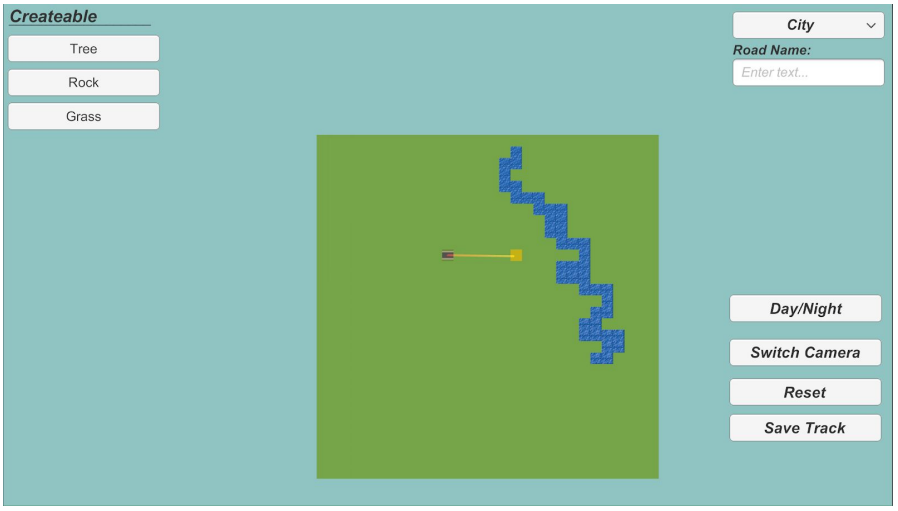
Procedural generation has been especially prominent in the creation of video games since the 1970's but even before this it was being used for other work using computers. However procedural generation can be seen also in nature so in this way it has been around much longer than you would think as it can be seen all around us today . In fact most of the world around us right now is created using a natural version of procedurally generation, the trees around us no two trees are exactly the same there is always some difference. Islands created by erosion due to the sea over millions of years,Coastlines being constantly battered by waves are all examples of this.[Blatz and Korn 2017]

Project Description

When the player starts up the editor, They are greeted by the editor UI and also a blank area along with an area of water running through it. If the player does not like the placement of this water you are able to reset this area to get another more agreeable one.

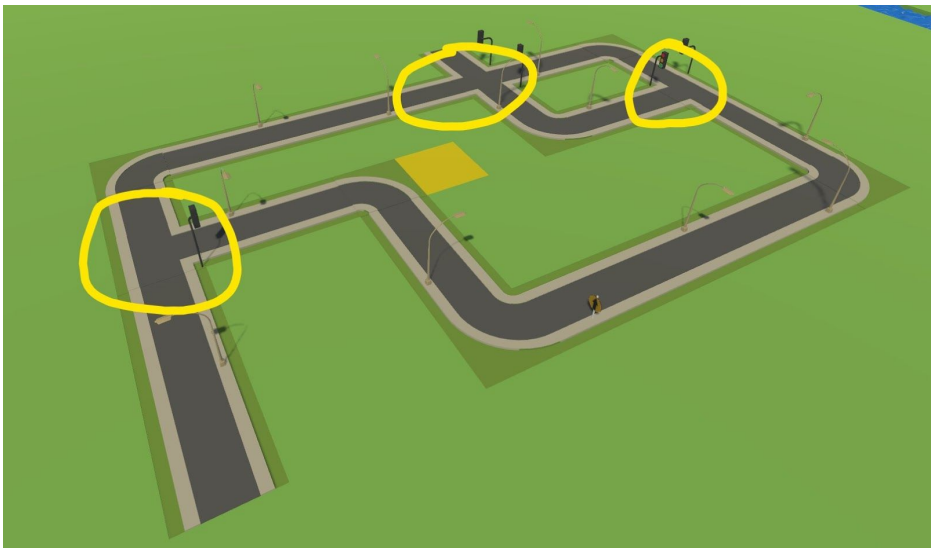


Once the area is suitable then if you look at the upper middle of the screen the player will receive a message telling them to choose a road before proceeding as they will not be able to do anything until the road type has been chosen.



Once the initial point and road type has been chosen, This is when the creation really begins, from the initial point of the road the player drags the cursor and when they move the cursor a line is created between this the initial point and the cursor. This line acts as a directional tool for the player to know in which direction they will move in. Once the player has chosen a direction and releases the mouse button then the angle is sent to the algorithm and from here it decides on the correct road to create for that given situation. As the player creates the road you may begin to overlap on some of the tiles and if that happens under certain circumstances then the editor will decide that a tile should be replaced with another tile that suits the current situation. so for example if two roads are perpendicular to each other and they intersect then

this point of intersection is changed to an intersection tile or if the player decides they want a turn on a previous created road then just move back to that road and then create the new corner road and if this happens this point is change to a T-Junction tile to create a seamless looking road. As you can see in the circled points below.



Along the roads that are made I am spawning roadside objects which range from road signs to lamp posts as well as traffic lights at certain points. These are there for both visual and in some cases functional uses. For example the lamp posts actually light up the game if the player decides to switch the scene to night mode.

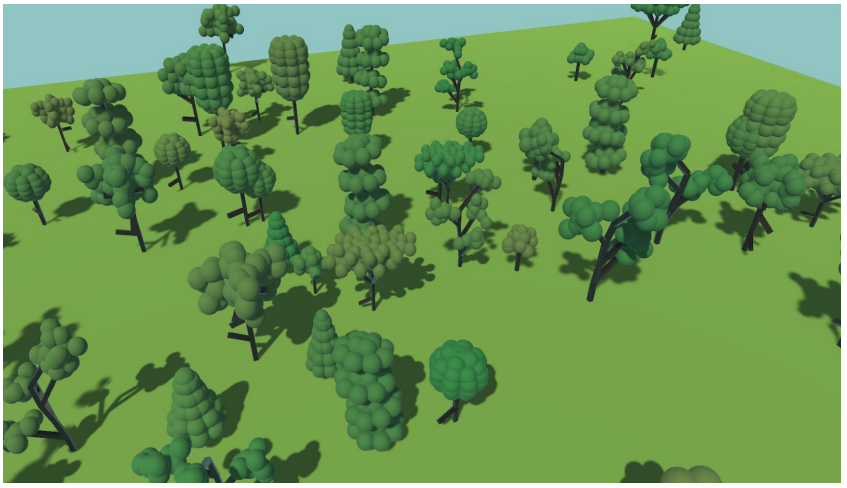
As the roads are made also along the side buildings are created using an algorithm to create the floors of the houses. The walls for the houses are selected using another function which decides which type of wall to create at that point be it a normal complete wall or one of the window variants and then once this is done it selects the door. The door is created on the front of the house and deletes one of the walls and replaces it with the door.

The roofs of these houses are created using procedural runtime mesh generation. Which is creating a custom mesh at runtime based on the parameters of the house and then placing these custom meshes at the correct points on the tops of the houses. I have several types of roofs and when the house spawns the algorithm decides which roof to use in that situation.

On these roofs depending on the roof type other roof add-ons can be added to include as seen in the image above dormer windows.

Below is an example of a completed road.

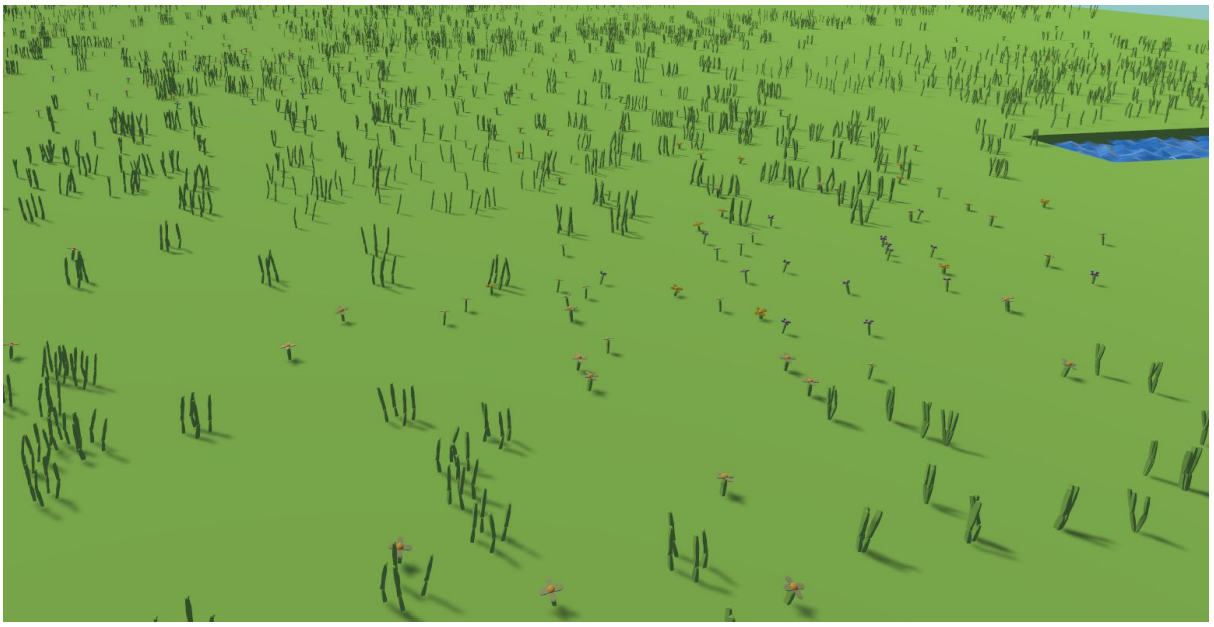
Once the player has completed this. They are then able to create other objects throughout the space. Using the menu on the left hand side of the UI the player is given the choice between Trees,Rocks and grass objects. The trees are place around the scene using an algorithm known as Poisson Disk Sampling which takes a point and the picks a random direction and then moves in that direction and checks if its outside the radius and if it's not then the point is not added to the list otherwise its added to a list and returned. Once the position has been decided then the L-System creates the tree from a choice of a few different rulesets that I created.



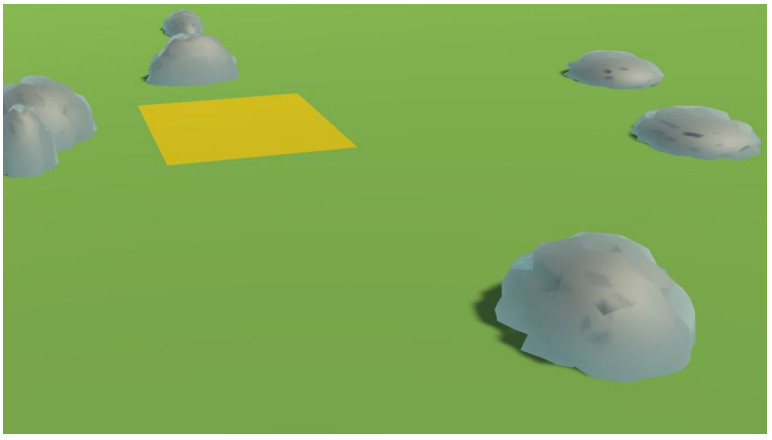
Using these two methods I am able to create completely different areas each time as the points are almost never the same. The trees also have an age statistic that changes both the size and color of the tree depending on the age.

Grass is another object I have created using L-Systems, the grass has a few different variants and along with that the grass also has some flowers mixed in with them to add more

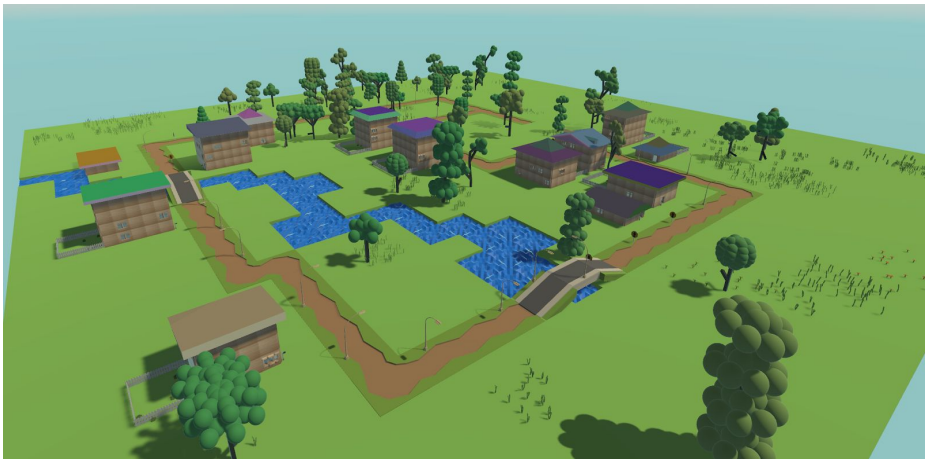
variation. The edible parts of the grass in the UI are the density of the grass objects which dictates the number of grass blades there are in that area and also another that spaces out the blades of grass to have them cover a larger surface area.



The rocks do not have any editable variables, but they are created using a simple perlin noise alterion to their meshes. I take their meshes and apply perlin noise to the rocks vertices this creates an irregular shaped mesh which creates the rocks.



Once the player has completed using all these features then the once blank area has now become an area filled with procedurally generated objects. In this way no two generations of the area will ever be the same.



All these mechanics are crucial for the effectiveness of the editor.

Bringing all these features and mechanics together and implementing them into an editor I feel answers the question I put forward at the start of my project and that is procedurally generating game assets in unity.

Overview

My Editor Design Goals

My Editor is not going to impact the research space in any fundamental way. However it will serve as a good basis for others looking into procedural generation and how it can be used. Some of the methods I have incorporated into this project however are currently used in other applications.

My Editor will work only using the unity engine as at the start of my research I had decided that unity was my best option to create this editor since I believed that unity would help me both achieve the level of clarity that I was looking for and also allow me to experience and improve my skills using a 3D engine.

What is the Game?

My project is not exactly a game, it's an editor that you can use to create a small area of a game world. Using my Editor you will be able to create a road network that can create buildings on the fly. Then after you have finished the roads and you are happy with their layout, then you can move onto using the other options to create other objects to add more variety and more impactfulness to the area.

Reason for creating this editor?

The reason I have decided to create this editor was because I have loved the idea of procedural generation since I was young and playing minecraft back in 2009 and since then I have been keen to try to understand how this could be achieved. So I decided that using these methods I could create a more interactive experience for the player If I incorporate them into an editor for the player to create something.

Where does the editor take place?

My editor does not take place in any specific place, it's more so about the player creating an area and then allowing them to decide what to do with this area. So if the player wants to name the area they have made or use it in a game world they have created.

What do I control?

In my editor the player will control the placements of the roads in the area and also the creation of the objects inside the scene like the trees,grass and rocks by using a rectangular area.When they are creating objects the player is given control of some of the variables that can change the objects size,shape and color. The player is also given the option to change the

time of day to night and day. AS well as these options the player is given the option to change the camera to a fully controllable camera that the player can move around the scene in and using this camera if an object is created and the player decides they do not want it there then they can delete it.

What is the main focus?

The main focus of my editor is to create a small scene using player input and also using procedural generation techniques to create varied landscapes.

Define the Application

The product I am creating is an editor that will allow the user to create part of a game world using only a few button presses. This editor will incorporate procedural techniques to make it so that no two parts of the world that are created are every exactly the same and that it is always varied.

What is the application supposed to do/ Main Features

1. **Road Creation:** This feature is the core functionality of the editor without this there would be no roads created, no houses and no roadside objects. So I believe that this is the most important feature of my editor.
2. **Building Generation:** The buildings I believe once they have been added to an area they add a good visual impact as well as adding more generated objects to the scene. Also the building's creation is much more complicated than the other objects.
3. **Vegetation(Trees,Grass):** The vegetation is another feature that I believe is critical for the visual impact and also the varying landscapes that my editor creates.
4. **Ui Controls:** The Ui is very important to my project as my project is an editor. Editors depend on the user interface to make them work.
5. **Combining Meshes:** Extremely important feature, Since without it there would hundreds of single objects in the scene making up all the trees,houses and grass. But since this feature is in the editor it allows all those object meshes to be combined and then delete the single objects to improve performance dramatically.
6. **Directional tool:** This directional tool is a very important part of the road creation and Ui, So I counted it in its own feature since it falls under two categories. Without this tool it would make creating the roads in the way you want very difficult.

Who is going to be using this application

The user groups that would be taking advantage and using my editor would be people who have an interest in procedural generation and want to see how it works and how beneficial it can be.

If given more time to fix the problems that occurred with the saving functionality the editors user groups would change slightly and the editor would then be also used by small developers who are looking to make a game.

Metrics

For my project, Since it is very hard to acquire metrics and data for the effectiveness of procedural generation I will gauge how successful my editor is by using the visual differences between two different instances of the area since this can show that they are created using these techniques and that both areas are varied and different from one another.

Is there a precedent for this application? (Your inspiration):

The inspiration for one of the features in my game was the road creation in SimCity. SimCity influenced the way I decided to build my path, particularly with a guidance tool that I thought was a fantastic idea when I saw it and would work extremely well inside my editor. My method differs in the way the road is generated. My road is generated in a tile based way rather than the SimCity approach which created the road along the length of the tool and also changes its shape depending on the angle of the directional tool. So in that way my road editor is different to the method used in SimCity.

Project Milestones

October 2019:

I began coding in an early idea for how the roads would be made. My initial idea was an idea using splines which is basically a curve that connects two or more points. I believed that this would allow me to create believable and smooth curved roads. However as I had a working piece of spline code that would allow me to show off the curve itself I believed that this was not the style I was looking to use. So I opted to try another method one that was more in line of the type of road I wanted to create. The method I was looking to try was to create a more tile based road creator that would use tiles instead of adding points to a pre existing line to add more road surface. So I began creating this and started with creating a square of connecting roads that knew its size by the sizes the player inputted but as I was testing this method and as I got other people to test it. the same conclusion was usually reached. That it was a very rigid system. So I went back to the drawing board with the actual creation of my tile based road.

November 2019:

At this point I had successfully created a road builder that I was happy with the player would choose the direction that they wanted to move and then using the angle of the player's mouse from the previous point a new road would be created and would slot into place alongside the other roads. The road was able to create corners,T-Junctions and intersections if any roads overlapped.I held another test and the results of this test where the total opposite of the earlier method, This one was much more well received and that this method was much more flexible in creation and allowed the user more freedom.

It was at this point that I decided to look into a way to create my trees and thanks to my research I had a great start with the knowledge I had gained about L-Systems I decided that this would be a good place to start and began working on creating a string based rule system to help me create my trees.

December 2019:

By this time I had a working L-System that was able to generate based on the rules input by me however I did not have a working rule that would create a tree with believable branches. It was just a basic rule that created blocks along the lines to simulate the trunk and branches of a tree being created. On the topic of the roads this month I decided to add a new feature to the roads that allowed you to move the roads up off the ground like a bridge which would come into play next month when I would be able to add in the asset to make the bridges make sense. By getting the L-System operational was a great milestone for my project as it meant that tree generation could be done.

January 2020:

Over the christmas break I implemented a water tile that is just a plane and im manipulating it vertices using a wave like motion to simulate water moving and as I implemented last month

this is where the bridges come into play they will allow you to make roads over these rivers. At this point this process is still manual as I need to be able to test this functionality wherever I am. I started working on UI and began with fields that allow you to choose the style of roads be they Dirt roads or city roads and also a way to name the track you've made so that then using the new save button which calls a new script that saves the newly created environment and places it into the a folder so it will be available for reuse.

February 2020:

At this stage I was happy enough with progress on the roads, So I decided to start work in the houses and I began working on the basic house where you could choose the size of the house,number of floors and the windows in the house. Then I began placing them along the sides of the road. I made a change that allows you to swap cameras and move down into the environment to let the user get a good look at what has happened and what's been created.

I also made minor changes to the UI that now allow you to completely reset the starting area.

March 2020:

At this point in the project I began creating the roof for the houses, Which was the hardest part of it for me. I Started with just using objects I had created in blender but this was not working out quite how I would like so I began redoing it again but this time I decided to create it using procedural mesh generation instead so I created the roof from vertices and triangles I created at runtime. This was still early days in this as I had to make more types of roofs.

April 2020:

I had talked about creating a roof using procedural mesh generation and this had developed a lot over the early part of april. This was a great milestone for the building generation for me and from there I continued to make more roofs.After I had made enough variations of roofs I then moved onto changing the L-System to allow me to create my own grass using the L-System and also while doing this I began creating new rules for the tree generation so that there would be more variations in the trees. I began updating the way that the trees would be positioned on the map, I began using poisson disk sampling to place the trees around in a more natural way. At this stage I began encountering some performance issues since I had hundreds of objects in the scene, So to counter this I implemented a way to combine all the meshes of an object to a single mesh to help with the performance and then I deleted the other objects after combining them all. I began fixing the manual way I have the roads moving over the water to an automatic approach.

Earlier in my development I had found a way to save the created area into a folder, However this feature began to show some bugs after changing to the the roof types and also since I was now combining all the objects meshes at runtime this meant that the meshes were being lost once the game was not running anymore this lead to errors with the saving, So I decided to leave it in the project for now to show that it was possible to save the objects.

Project Review and Conclusions

In this section I plan to review my project and discuss the development of my project process as well as what went right and also what went wrong. Along with that some things I would do differently now looking back on the project.

A few things that went well I thought in the project were the creation of the roads and the way that I have the roads moving automatically over the water. Since when I first tried to do this I had some trouble trying to determine when I had hit a ground tile after being over the water. Another feature I am very happy with was my L-System which I implemented to a degree of functionality that I was incredibly happy with.

A few things I was less than happy with in my project include my implementation of the rock generator as I feel like given more time I would have been able to flesh out that feature much more but due to time issues I had to prioritize the other features over that feature.

Another problem that occurred early on in development, Was that I had begun looking into using bezier curves to begin my development on the roads. My plan was originally to create the road mesh alongside the bezier curve and then just add points to the curve which would in turn continually add the mesh. While I had a working prototype of the curve itself with no mesh around it, I had then realized that this was not the way I wanted to approach this after this investigation.This looking back now was something that I had approached wrongly at the start and I should have given more time think about what I wanted the road editor to look like.

Early on in the project I had found a method through my research into unitys capabilities to save an object created at runtime. At that point in the project when I had not incorporated all the systems into a single scene I only had placeholder objects for the trees and the buildings it worked like a charm and would save the object out as a prefab, however once I combined all the systems into the one scene and then implemented my new combine mesh function it cause that to cease working as since I was now created meshes at runtime and deleting the previous ones to save performance this lead to the mesh not saving with the prefab and thus losing them after saving. I began looking into ways to tackle this problem and one way I had come up with was to save the whole area and then save the positions of everything on the area into a script or file and then once the player drags the saved object into a script just use those positions to recreate the area. But due to the time frame left in the year, I decided that I would prioritise other tasks in the project before this one and if I had time at the end I would revisit it.

If I had more time to develop the project the saving feature is definitely one I would revisit and try to complete as I think it would be a very good feature to have in the editor. Another thing I would like to revisit would be the performance of the editor as at the moment it is still not as smooth as I would like it to be. Although I had implemented my mesh combining into the project I still feel like there are many other ways that I could optimise the project to run even smoother. Another feature I would implement would be a slow build mode where the

objects that are built are created slowly so you can watch them build from the ground up to show the player them being built at run time.

If I had to start this project I would definitely have spent more time researching into the different methods I could use into building the roads, By the end I was happy with the method I had chosen but I believe that had I taken more time to think about what the completed version of my project would look like I would have reached my current method much faster. I would also have done weekly refactoring as by the end some of the code had become rigid and could most definitely have been more flexible.

To others attempting a similar project in the future I would say to get a good idea of the main assets you wish to generate first and see if there are any other objects that you could incorporate into the project that would elevate that main one up even higher.

I believe that my choice of technology was actually a great choice as the amount of literature on how to use unity helped me tremendously through my development and also with its visuals I was able to produce a much better looking game than doing my project in 2D.

Even so, I believe that my decisions were for the good of the project as a whole throughout and I had to make some difficult decisions and some sacrifices had to be made to deliver this project. But I can honestly claim that this has made my project a success, and I'm very pleased with that.

References

Number	Referenced Publication	Citation	Reference
1.	Website	(Microsoft n.d)	Microsoft. (n.d). *Microsoft purchases 'Minecraft'*[online]. (https://news.microsoft.com/announcement/microsoft-purchases-minecraft/).(Accessed 1 December 2020).
2.	Website	(Blatz and Korn 2017)	Blatz,M. and Korn,O. (2017). *A Very Short History of Dynamic and Procedural Content Generation* [Online]. (https://www.researchgate.net/publication/315863952_A_Very_Short_History_of_Dynamic_and_Procedural_Content_Generation). (Accessed 13 March 2020)

Faculty of Science

Open-Book and Remote Assessment Cover Page

Student Name: Brian O'Neill

Student Number: C00213756

Lecturer Name: Philip Bourke

Module: Project II

Stage/Year: 4th Year

Date: 03/05/2020

Declaration

This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.