# Computing Project Technical Plan

**Name:** Jonathan Mills

**Course:** BSc of Computer Games Development

**Supervisory Team:**

## Title

Procedurally generated terrain using a game engine.

## Summary

This project intends to create a game engine with the ability to procedurally generate (procgen) terrain using parameters given by the user. Weather effects and a day night cycle will be attempted to be implemented into the game engine, if everything else has been implemented properly. The benefits of undertaking this project will be a further understanding of how game engines are created and furthermore, prototypes. Another benefit would be further learning of how graphics are implemented into game engines. I will be using to agile methodology while working on this project as it allows for flexibility, when obstacles occur during the development process.

## Deliverables

I will be submitting a project solution with all the source files. Another artefact that will submitted is the demonstration video, demonstrating the key features of my project. The final artefact to be submitted is the project executable to allow the marker/users to run the project themselves to understand how it works.

## Constraints

A major constraint on this project is the time management aspect of the project timeline in relation to the fixed deadlines, as well as having several months to complete the project. A way to overcome this constraint is to set up a schedule to follow and to stick to it as much as possible. A method that will be used to set up and follow this schedule is a KanBan board, the one that I will use is Trello®, as this has many features that will enable me to stick to the schedule that I have set up.

## Key Problems

One problem that needs to be overcome is the designing of an intuitive user interface to allow the user to easily understand what each control does to the program. This problem will be overcome by the implementation of the imGUI library as this has been specifically designed for user interfaces and has a free software license.

Another problem is the efficient implementation of the Perlin Noise algorithm within my project. This is the main problem to overcome within my project as this algorithm is the key to my project succeeding. However, I do not think that this will be a hard problem to overcome.

The third problem to this project is the creation of a 3D graphical environment, luckily I was taught last year how to use DirectX11 as the graphics library to do this, and have based this project off of the ‘skinning’ lab project from Year 2. This lab will be used for now until I fully understand how to setup a DirectX11 environment to the specifications that will be required.

## System and Work Outline

There are several key requirements within this project: the creation of a very basic game engine type program that will be able to procedurally generate terrain based off of the user input, an intuitive user interface, and the algorithm that will be used to generate this terrain. I will be using visual studio 2019 as my development environment with DirectX, imGUI, TinyXML and potentially assimp as external libraries to help me develop this project. I have checked the licenses for these libraries and have included the necessary licenses within my project.

I will begin development of this project by building the outline of the game engine and including the basic needs of the program. After this is done, I will be implementing the Perlin noise algorithm in my project to generate coordinates for the terrain that will be created. Once this is finished the finishing touches of the program will be done by designing and implementing the User Interface to allow the user to perform the function of the project. If I have enough time after this, I will be implementing the Midpoint Displacement algorithm as an additional option for the user, to understand how the different algorithms generate terrain. Doing it in this order will allow to me to deal with any problems/errors that occur, quickly since I will be doing everything in parts and this will also allow me to adapt to any further changes required in my project.

## Project Activities

## Risk Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Risk | Severity | Likelihood | Action |
| Lose of access to resources required | Extreme | Very Low | Restart Project with another idea. |

## Options

Lifecycles:

One of the first Lifecycle options that I considered was the waterfall method because it is the easiest to understand and work with. This methodology has several advantages such as having a clear structure to follow and the end goal is determined early on within the project. However, I have chosen not to use this methodology due to the fact that it is not adaptable to new problems that can arise during the lifecycle of the project and that it does not permit testing until the project has been completed.

The second lifecycle option that I considered was the agile methodology.This was considered because of how popular it is among software developers in the industry. This popularity is warranted due to the several advantages this lifecycle provides. One of the big advantages, is how flexible it is to changes/problems that occur during the development lifecycle as the methodology works by breaking the lifecycle into 4-week segments/scrums and at the end of every scrum a review is done to see if anything new needs to be worked on. One disadvantage of this methodology is that there is no clear end goal to work towards as these 4-week scrums could go on indefinetly. The reason I chose to use this methodology is due to the flexibility as discussed earlier.

Development Tools:

There were two options I considered using for providing the graphics libraries to my engine, OpenGL and DirectX 11. I will go through there pros and cons before deciding which one to go with.

OpenGl was considered because it is easily portable to other platforms such as web browsers. This would make it easy to provide my engine’s services to more customers. However, there are several disadvantages to using OpenGl within my project, one the of the big disadvantages is the lack of proper documentation that is readable without confusing the reader. This will not be workable for me as I would be learning OpenGL from scratch and this would delay the project timeline significantly. Another disadvantage is that the implementations of OpenGL vary between graphic card manufactuares, such as Nvidia and AMD.

DirectX 11 was considered because of my experience using it during Year 2, I found directX 11 to be powerfull and understandable during this time. DirectX 11 is very similar to OpenGL except that the implementation doesn’t change between graphics and the documentation being easier to understand. I have chosen to use DirectX 11 in my project.

Algorithms:

There were two algorithms that were being considered for this project, Perlin Noise and Midpoint Displacement. The midpoint displacement algorithm is efficient for the generation of height maps used in terrain generation, However I have chosen not to use this algorithm as my main algorithm, instead opting to add it as an option to the user, allowing them to see the difference between the two algorithms. Losh (2016) goes into detail on how this particular algorithm works, however he is using Wisp to demonstrate the code used, therefore I will be using his work to understand the theory behind the algorithm.

The Perlin Noise algorithm has been chosen as the main algorithm used, because there are more sources available online to help me understand how it works and it generally generates smoother terrain.

## Potential Ethical or Legal Issues

If this project were planned to be sold commercially, one legal issue that would arise is copyright status of this project and whether end users would be able to use the engine to create their own piece of works, the solution to this problem is to decide how I would want the product to be licensed and whether I will allow end users to create commercial products with this project. An ethical issue that might occur with this project is the type of content that can be made with engine by the end user. An example would be content made that is too inappropriate for certain communities. This will not be an issue for this project as the content that can be made from the engine will be restricted to terrain and weather effects.

Another legal issue that can occur is whether I will be able to use the Perlin Noise Algorithm in this project, for example, whether I will need to pay a certain fee for the use of the Perlin Noise Algorithm. This looks like it is not the case and that I will be able to use this algorithm within my project without having to pay anything and will not be required to produce a license. A final issue that I have found that could arise is the ethical issue of prohibiting users to use my project - if it is to be sold - through not allowing accessibility features.

## Commercial Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Factor name | Description | Is this a cost or a benefit | Estimated Amount | Estimate of when paid |
| Salary | The average salary for a software developer in England | Benefit | £4,800 | Monthly until the end of the project |
| Price | The price of the product if sold commercially | Benefit | £300 | After project, if customer can be found. |

The estimated time to complete this project is about 4 months of work. There are not many costs for this project as this will be done individually and with the intention of the project to not be sold. If this was to be sold commercially, the main cost of this project will be the workforce required to complete this project. The average monthly salary for a programmer in the UK is £4,800, therefore the calculated cost for the working hours of one programmer for this project is £19,200. This project would be planned to be developed to support a company, providing them with the tools to be able to develop different game worlds/levels for different scenarios. This project could also be altered to affect different areas/systems in a video game. i.e. Weather systems, loot found in chests. if this project was done commercially, I would package it as its own product and propose its use to several game companies, providing it as a package that could be implemented within their other game engines.

## Employability Contribution

This project will improve my technical skills since I am working on a back-end supportive engine that will enable computer games to have the ability to create new terrain on the spot with different parameters.I have needed to research the Perlin Noise algorithm in order to create this project, giving me the knowledge to understand how the algorithm works. I can now use this knowledge to more easily understand how other algorithms are designed and meant to function. By going through with this project, my understanding of programming techniques has improved significantly as I worked through the problems that arose. I would be able to bring these techniques to where I will end up working.

## References

# Bibliography

Archer, T. (2011). Procedurally Generating Terrain. *Midwest Instruction and Computing Symposium* (pp. 1-15). MICS.

Erstu, E., Sell, J., & valli, S. (2017, May 27). Perlin Noise Generator.

Ginting, A., Sari, K., Fadhilah, C., & Yusra, R. N. (2019). Application of the Perlin Noise Algorithm as a Track Generator in the Endless Runner Genre Game. *Journal of Physics Conference Series*.

Gregory, J. (2009). *Game Engine Architecture.* Wellesley: A K Peters, Ltd.

Hendrikx, M., & Iosup, A. (2013). Procedural Content Generation for Games: A Survey. *ACM Transactions on Multimedia Computing Communications and Applications*.

Lagae, A., Lefebvre, S., Cook, R., DeRose, T., Drettakis, G., Ebert, D. S., . . . Zwicker, M. (2010). A survey of Procedural Noise Functions. *COMPUTER GRAPHICS forum*, 1-20.

Losh, S. (2016, February 19). *Terrain Generation with Midpoint Displacement*. Retrieved from Steve Losh: https://stevelosh.com/blog/2016/02/midpoint-displacement/

Lucid Content Team. (n.d.). *The pros and cons of Waterfall Methodology*. Retrieved from Lucid Chart: https://www.lucidchart.com/blog/pros-and-cons-of-waterfall-methodology