# Computing Degree Project Proposal

**Student Name:** Jonathan Mills

**Course:** BSc of Computer Games Development

**Project Title:** Procedurally generated terrain using a game engine.

## Project Context

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 implement into the game engine. The benefits of undertaking this project will be a further understanding of how game engines are created and furthermore, game prototypes are created. Another benefit would be further learning of how graphics are implemented into game engines.

Procgen has been used to many different effects in video games, *Crusader Kings 2* has used it to create rich and detailed family trees dynamically and *Shadow of Mordor* has used procgen to create new and exciting enemies that would remember the player if they were injured by them (Moss, 2016). Procgen has also been used for generating new terrain in many games as this is its main purpose, one of the best examples of this is Sid Meier’s *Civilization* series (Moss, 2016).

A few researchers have gone through the many different techniques of generating noise and the different categories while discussing their benefits and cons (Lagae, et al., 2010). The algorithm I chose to implement is the Perlin noise algorithm, a terrain smoothing technique to overcome a rough field (Ginting, Sari, Fadhilah, & Yusra, 2019), to generate the values required to set the locations of the textures in the scene. The workflow to be used/implemented for generating perlin noise is described in Figure 3.1 in Erstu’s Perlin Noise Generator (Erstu, Sell, & valli, 2017, p. 5).

## Specific Objectives

* Basic Game Engine.
* Procedurally generated terrain based on user entered parameters.
* Different weather effects

## 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 most likely not the case and that I will be able to use this algorithm within my project. 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.

## Resources

* DirectX 11
* Visual Studio 19
* Programming Language: C++, HLSL
* Github or bitbucket (version control)

## Potential Commercial Considerations - Estimated costs and benefits

The estimated time to complete this project is about 4 months of work, there aren’t many costs for this project as this will be done individually and not to 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 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.

## Proposed Approach

The first step would be to try and create a 3D game engine, that will focus on the generation of terrain, this will take the longest of all the tasks required as this is the biggest task to complete. The engine will need to be intuitive for the user to use and will provide the user with parameters to control the terrain generation. I estimate that this task will take about 1-2 months of work. The next task will be to create a perlin noise algorithm, commonly used for texture generation in games (Hendrikx & Iosup, 2013).This will take an estimated 1-2 weeks of work. Another task will be to create the procedural generation algorithm to generate the terrain using the game engine, I estimate that this will take about a month of work to complete. The final task will be to add weather effects and/or a day/night cycle, this task is optional and will be attempted if the three other tasks are completed. I estimate that this additional task will take a month to complete. The main tools that will be used is visual studio community 2019 and DirectX 11.

# References

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.

Moss, R. (2016, January 01). *7 uses of procedural generation that all developers should study*. Retrieved from Gamasutra: https://www.gamasutra.com/view/news/262869/7\_uses\_of\_procedural\_generation\_that\_all\_developers\_should\_study.php