# Abstract

This document is the proposal for the final project and dissertation for Games Software Engineering. Starting with an introduction to the procedural planet generation idea and moving on to detailing all the individual components that would need to be created, in order to state the scope of this project. A literature review is then done to find techniques to complete these components and look for similar work within this field. Following that, is a brief methodology of how the project will be completed and detailed Gannt chart showing the exact timeline and plan for the project.

Contents

[Abstract 1](#_Toc95379412)

[Introduction and Rationale 3](#_Toc95379413)

[Aims and Objectives 3](#_Toc95379414)

[Literature Review 4](#_Toc95379415)

[Methodology 5](#_Toc95379416)

[Project Plan 5](#_Toc95379417)

# Introduction and Rationale

The purpose of the project, detailed below, is to conduct research into the most efficient and highest fidelity way of creating procedurally created planets, that are also a realistic scale. The finished product should allow for the fast generation of scale detailed, randomly generated 3d planets. These planets will have features such as: multiple biomes, level of detail system, atmospheres and procedurally textured surfaces. This area is being investigated because, although all these areas have been individually researched and refined, they have yet to be combined into a single library/experience.

# Literature Review

The first area that was investigated was the best method for creating the vertices for a sphere in a 3d virtual space. This was necessary as a base mesh is required for the creation of the planets. In this specific field, there are many methods for sphere generation. Some of the most popular ones are UV spheres, normalized cubes, spherified cubes and icosahedron (Cajaraville 2019). Unfortunately, all these different techniques have both positives and negatives associated with them. An ideal technique would be where the distribution of points is even, and close to as possible to the points on the actual sphere. Although all these techniques would work for the implementation, the distribution is still not as even as preferred. As such an alternative was found, called the Fibonacci sphere(S 2022). This is known as a “a well-known approach to generate a very uniform sampling of the sphere” (Keinert et al. 2015), and as such would be perfect for generating procedural terrain, if wrapped and triangulated correctly.

The next area that was investigated was the issues that would be experienced with floating point numbers at the intended scale. This is because floating point values would begin to loose precision at 1000km (O’Neil 2022). Using some of the techniques outline by Sean O’Neil in his article, such as manipulating the view matrix, scaling the planets and using doubles, this issue should be resolved.

The next major part that would require some research would be, how to generate the terrain of the planet and all its features. From research, the most standard method for generating terrain is to use some version of noise in order to generate the terrain (Fischer et al. 2020). Of this noise there are 5 main types, which all have their positives and negatives (Vitacion and Liu 2019). But as seen in Vitacion and Liu’s paper, although this noise does provide a good base, noise alone will not allow for a detailed multi environment terrain. In contrast, the work done by Fischer et al. demonstrates how to use this noise as a base, and add some climate simulation on top of this to define biomes and more detailed terrain features (Fischer et al. 2020). The only downside of this implementation, is that is designed to work on a flat piece of terrain and as such, would have to be adapted to work on a spherical world.

Another major part that required some research was the creation of atmospheres. Although there a several methods used in the industry, a variety of them would be unsuitable for the application being created here. One of these insufficient methods would be volumetric ray marching, as this method only allows for the rendering of the atmosphere from outside of the planet (Elek 2009). Fortunately, a method described in papers both by Elek and Schafhitzel et al., will allow for the creation of a mostly scientifically accurate atmosphere with good performance, as the first step (computing the scattering integral) is precomputed before runtime operation(Schafhitzel et al. 2007; Elek 2009).

The final aspect that needed to be investigated was how to implement a level of detail system. This is crucial to the ability to run the intended executable, as the scale of the planets would not allow for a full detail model to be displayed due to the shear number of triangles being rendered. As such being able to switch between different complexities of mesh or dynamically change the mesh’s complexity would be necessary. Based on the research done, there are 3 main things that can be done to the simplify and arbitrary mesh. These are: coarsen the mesh outside of the view frustrum, screen space geometric tolerances (measuring surface deviation from the original model then refining/coarsening the model based off an error value), and surface orientation (coarsening geometry not in view)(Hoppe 1998). Additionally, this technique described by Hoppe, allows for the smooth transitioning between these levels of detail using geomorphs.

# Methodology

Upon starting this project, the first thing that will be conducted is a further literature review to verify all of techniques discovered in the preliminary view seen above, as well as research any additional methods that would be suitable for this project. Additionally, some initial research will be conducted into the areas of the stretch goals, so that if time for their implementation is available, it can be completed. Next the implementation for generating the sphere mesh using the Fibonacci sequence would be undertaken (Keinert et al. 2015). Along with this initial mesh, the level of detail system will also be implemented, as it too is a required component for the rest of the features. The next thing to be added would be then the actual terrain generation algorithms, based off the ones described in the paper by Roland Fischer et al., implementing and adapting it to work on a spherical world. Then the final major step in completing the main project would be undertaken, the addition of the atmosphere. This will be done using the techniques described in the paper by Schafhitzel et al. Then, if time allows, the stretch features will be completed in the order of multithreaded optimisations, Oceans and then Terrain manipulation.

# Aims and Objectives

The main goal of this project is to create an executable that demonstrates the research done and implementing a complete procedural planet generation system. The final executable will allow the user to set parameters for the generation of the planet, then proceed to explore the generated planet with a flying camera.

Following are all the components and problems that must be solved and researched in order to complete the project. These are the components completely necessary to the project and must be completed before the project deadline.

* Generating initial sphere mesh
  + Creating the vertices and geometry for the planet.
  + Fixing any floating point precision errors that would appear due to the scale of the planet.
  + Setup a level of detail (LOD) system to simplify and coarsen the mesh of the sphere, will allow for LOD on the finished terrain.
* Generated Terrain on Sphere
  + Basic Terrain Generation
    - The basic started terrain shape that would be spread across the entire body of the planet.
  + Biome generation
    - Create the biomes on the planets, and then tweak terrain details to refine the individual biomes
* Planet atmosphere system
  + Creating the sky colour and visible atmosphere, and visualisation of the sun.

In addition to these main components, there are some stretch goals that would ideally be featured in the final version but will only be added if time allows. These additional components are listed below.

* Multithreaded generation
* Terrain manipulation
* Ocean support

# Project Plan

Shown below, and attached with this document, is a Gannt chart outlining the stages in which the project shall be completed, as well as all additional and associated deadlines. This Gannt chart shall be followed step by step to complete the project.

Works Cited

Cajaraville, O. S., 2019. Four Ways to Create a Mesh for a Sphere. *Medium* [online]. Available from: https://medium.com/@oscarsc/four-ways-to-create-a-mesh-for-a-sphere-d7956b825db4 [Accessed 2 Feb 2022].

Elek, O., 2009. Rendering Parametrizable Planetary Atmospheres with Multiple Scattering in Real-Time, 8.

Fischer, R., Dittmann, P., Weller, R. and Zachmann, G., 2020. AutoBiomes: procedural generation of multi-biome landscapes. *The Visual Computer*, 36 (10–12), 2263–2272.

Hoppe, H., 1998. Smooth view-dependent level-of-detail control and its application to terrain rendering. *In*: *Proceedings Visualization ’98 (Cat. No.98CB36276)*. Presented at the Proceedings Visualization ’98 (Cat. No.98CB36276), 35–42.

Keinert, B., Innmann, M., Sänger, M. and Stamminger, M., 2015. Spherical fibonacci mapping. *ACM Transactions on Graphics*, 34 (6), 193:1-193:7.

O’Neil, 2022. *A Real-Time Procedural Universe, Part Three: Matters of Scale* [online]. Available from: https://www.gamasutra.com/view/feature/131393/a\_realtime\_procedural\_universe\_.php [Accessed 4 Feb 2022].

S, A., 2022. *Delaunay+Voronoi on a sphere* [online]. Available from: https://www.redblobgames.com/x/1842-delaunay-voronoi-sphere/ [Accessed 4 Feb 2022].

Schafhitzel, T., Falk, M. and Ertl, T., 2007. Real-Time Rendering of Planets with Atmospheres, 8.

Vitacion, R. J. and Liu, L., 2019. Procedural Generation of 3D Planetary-Scale Terrains. *In*: *2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT)*. Presented at the 2019 IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT), 70–77.