Initial Plan

Procedural Terrain Generation Engine

Author: Teodor Nikolov

Supervisor: Frank C Langbein

Moderator: Bailin Deng

Module: CM3203 – One Semester Individual Project – 40 credits

Project Description

In this project I will build a software library written in C# for Unity Engine. My software will deal with **generating 3D procedural landscapes** which will be rendered by Unity's rendering engine, having provided a mesh. The "Engine" part of my project involves the generation of **landscapes with different forms and features** based on how the parameters of the engine are set. The possible diversity can include mountainous terrain, canyons carved inside a smooth surface, caves, and others.

Prior to writing this report, I had already made some progress on the project. So far, I have got a **chunk-based, hierarchical system** of the world, where individual chunks can be generated with their coordinates considered. The generation is **reproducible**, in the sense that a seed is used every time to initialize the world. Using the same seed will always result in the same landscape.

For the moment, I am only generating a heightmap of the terrain using 2D Perlin noise. In the future, I will introduce other noise generation algorithms, such as 3D Perlin / 3D Simplex noise, Perlin Worms, and other algorithms, carefully combining them to enable the above-mentioned terrain diversity. I am currently **extracting the mesh** data using **Marching Cubes**.

The procedural generation element involves two bits: the generation and the management of the terrain chunks. For the former I will investigate **parallelizing** this process to enable faster generation of the terrain/mesh using a task-based system to avoid thread creation overhead. For the latter, management means that chunks of terrain will be **dynamically loaded and unloaded** depending on player location and view distance. To assist that, I will implement **quadtrees**.

If I have enough time, I will try to introduce biome-like generation, so that the terrain could be even more diverse. I will also try to better parallelize the generation process by parallelizing the Marching Cubes algorithm and will also look at terrain modification and saving/loading the world so that all mechanics will enable greater customization of the world.

Project Aims and Objectives

The primary aims of my project are:

- Procedurally generating reproducible landscape data using noise algorithms such as 2D and 3D Perlin Noise
- Extracting mesh data from the landscape data using Marching Cubes
- Splitting the terrain into manageable chunks and procedurally generating/destroying chunks as deemed necessary with the assistance of quadtrees

The secondary aims of my project are:

- Make the terrain generation biome-based, which will consider what terrain needs to be generated for what biome (desert, ocean, mountains, fields, etc.)
- Make the terrain modifiable (placement and removal of terrain data)
- Saving/Loading of the terrain to/from a file

Work Plan

1. Milestones

- **Week #8**: Develop a parallelized chunk manager responsible for the generation/destruction of new/old chunks controlled by player position and view distance.
- **Week #12**: Develop a terrain generation system that creates complicated terrain formations and shapes by combining different noises and applying modifications on the raw data.

2. Weekly Workplan

Week number	Week starting	Aims and Objectives
2	05/02/2018	- Investigate how quad trees can be implemented in a chunk-based system and how it can assist the management of the landscape - Further investigate how to achieve more realistic-looking terrain and what techniques are currently out there that achieve that
3	12/02/2018	- Begin implementation of the chunk manager - **Chunk Manager:** Introduce simple parallelism to the generation process (Ex. 8 threads – 1 chunk per thread)
4	19/02/2018	- **Chunk Manager:** Introduce a simple, dynamic chunk management system (chunk generation/destruction)

5	26/02/2018	- **Chunk Manager:** Implement and adapt quadtree data structure to the chunk-based hierarchical system
6	05/03/2018	- **Chunk Manager:** Continue Implementation
7	12/03/2018	- **Chunk Manager:** Alter chunk manager to work with the quad tree data structure to manage the chunk-based world
8	19/03/2018	- **Chunk Manager:** Continue Implementation - **MILESTONE 1** - Review meeting with Frank
9	26/03/2018	- Begin Implementation of advanced terrain generation system - **Terrain Generator:** Begin development of a noise stacking and noise blending system to enable rich terrain features
10	2/04/2018	- **Terrain Generator:** Continue Implementation
11	9/04/2018	- **Terrain Generator:** Continue Implementation
12	16/04/2018	- **Terrain Generator:** Continue Implementation - **MILESTONE 2** - Review meeting with Frank
13	23/04/2018	- Write final report
14	30/04/2018	- Write final report - Submit report on 07/05/2018

For the Chunk Manager, I have got some ideas of how it could work. However, I am not yet fully sure how I would go about implementing quadtrees, so I left myself more room for the actual implementation. Regarding the terrain generation system, it heavily combines a noise-generation system and then the usage of the said noise-generation system. Because the system can grow in complexity as more features are added, I left myself plenty of time to develop this essential module of the project.

The Easter holidays I will aim to work on the primary components of the application, hopefully leaving myself some extra room from Week 9 onwards to develop some of the secondary goals for this project. Every week I will have a scheduled meeting with Frank.

3. Deliverables

- Source code of the program
- A final, stable build of the program to be ran on the Unity 3D engine
- The final report