A5 Project Proposal
Title: 3D Nature Environment with Procedurally Generated Trees
Name: Liam Ozog
Student ID: 20515121
User ID: lozog

Final Project:

Purpose :

   To render a 3D environment that can be explored, featuring terrain, water, and procedurally generated trees.

Statement :

   For my project, I intend to render a 3D environment with terrain, procedurally generated trees, and reflective water. Much of the inspiration for this project comes from David Whatley's chapter in GPU Gems 2, 'Toward Photorealism in Virtual Botany'. Many of the topics covered in that chapter are beyond the scope of this project, but it did pique my interest in nature scenes. The other inspiration was Ryan Geiss' chapter in GPU Gems 3, 'Generating Complex Procedural Terrains Using the GPU', which got me interested in procedural generation.

   This project is more advanced than any real-time program I have ever attempted, so it will be an interesting challenge to try to achieve the objectives while maintaining a reasonable framerate. I also hope to achieve some interesting, perhaps even life-like trees when implementing L-systems to generate tree models.

   All of these objectives are completely new topics to me. I've been playing videogames for much of my life, and am familiar with many of these effects in terms of the end result (how they look), but I haven't given much thought as to how they are implemented. A lot of ground will be covered - procedural generation with L-systems, mapping of height and texture, and interesting lighting effects. As I noted above, this will be the most advanced real-time program I have attempted, so I expect to learn a lot in that area.

Technical Outline :

   The terrain will be modelled from a height map, with a grayscale gradient representing height. The height map will be generated using Perlin noise.

   I will implement a skybox in the scene using the technique of cubemapping.

   The landscape will have water up to a fixed height. Using OpenGL's stencil buffer, I plan on adding reflections to the water.

   The environment will be populated with foliage - grass and trees.

   Since it's unrealistic to render each individual blade of grass, I will use billboards to render 'clumps' of grass. Additionally, I plan on using a 'screen-door' effect (also known as a dissolve effect) to simulate alpha transparency for grass - since calculating the attentuation of alpha transparency for the grass as distances increases is expensive.

   I plan on making use of Lindenmayer systems to procedurally generate tree models to make the world seem more "life-like".

   The ground and models (grass and trees) will have textures applied to them using texture mapping.

   Finally, I plan on making the scene look nicer (for some definition of nicer) by implementing high dynamic range (HDR) rendering and bloom effects, which both make use of framebuffers.

Bibliography :

   https://developer.nvidia.com/gpugems/GPUGems3/gpugems3_ch01.html

   Ryan Geiss' chapter in GPU Gems 3 is all about procedural generation.

```
https://developer.nvidia.com/gpugems/GPUGems2/gpugems2_chapter01.html
```

David Whatley's chapter in GPU Gems 2 covers a lot of ground on how to represent foliage in a scene effectively.

```
http://algorithmicbotany.org/papers/abop/abop-ch1.pdf
http://algorithmicbotany.org/papers/abop/abop-ch2.pdf
```

The first two chapters of The Algorithmic Beauty of Plants provide a detailed introduction to L-systems, then go on to speak more about using them to generate trees.

```
http://www.allenpike.com/modeling-plants-with-l-systems/
```

Another paper on L-systems, with particular attention to generating trees.

Objectives:

Full UserID:_____   Student ID:_____

___ 1: Objective one: UI: Implement a first-person camera with associated controls to allow navigation of the scene, including movement in 3 axes, speed adjustment, and camera rotation.

___ 2: Objective two. Modelling: Add a skybox to the scene using cube mapping.

___ 3: Objective three. Implement reflections for water using OpenGL's stencil buffer.

___ 4: Objective four. Generate a pseudo-random terrain heightmap with Perlin noise.

___ 5: Objective five. Add grass to the scene using billboards to create the illusion of many blades of grass.

___ 6: Objective six. Add texture to the ground and foliage using texture mapping.

___ 7: Objective seven. Use L-systems to procedurally generate trees.

___ 8: Objective eight. Implement shadows using a depth map stored in an OpenGL frame buffer.

___ 9: Objective nine. Implement bloom using framebuffers and Guassian blur.

___ 10: Objective ten. Use a 'screen-door' effect to simulate alpha transparency for grass.