CS488 Project Proposal 2

Fall 2011

Anthony Cameron
20231689
a5camero

For Stephen Mann

CS488 Project Proposal 2

Topics/Purpose

I mean to completely ignore the classical OpenGL fixed pipeline by implementing my own pipeline and use GPU shaders. By creating a small game world, I aim to explore what can be done with shaders to produce compelling effects. I would like to explore the following concepts: reflection, shadows, textures, bump mapping, global illumination inspired techniques, animation techniques.

The core of this project is the abstraction of the terrain generation to be procedural: land, architecture, and trees will be dynamically generated and placed. It will be possible to change the camera angles and position.

Houses/buildings will be explored with context-free grammars, and if time permits, context sensistive (in terms of their arrangements / city structure). Trees will be explored by L-systems and others (e.g., random upward walks). If time permits, clouds may be generated by 3D or 4D Perlin noise. Textures like marble floors and wood can also be textured procedurally with Perlin noise. The terrain itself will be generated procedurally by means of some volume map modification, using noise generating functions and blending functions. The goal is to insure the noise function is continuous over the world chunk, so that adjacent chunks can be calculated (and also achieve the same continuity). Water will be explored via various shader techniques.

There will be some bare minimum amount of UI work (in menus, and so-on) for the user to properly use the program, but this is not the focus.

Statement

My goal is to create a small strategy RPG engine. By engine, I mean just one instantiation of a possible battle. It will be possible to re-generate a world to fight in. Thus, this project serves practical use as a potential engine for a full game. It may turn out I don’t have enough time to do a full battle system, and as such, it will just be a world to explore. Additionally, I would like to explore the effective mixture of 2D sprite art (stolen from the Internet, ripped from other games) in a 3D world (my creation). The concept is heavily inspired by the isometric games Final Fantasy Tactics, Ogre Battle, and Disgaea.

Technical Outline

Lighting and texture maps will proceed very straight-forwardly. It is a relatively simple matter to look up a texture and modulate shading accordingly. These are things I’ve done in the past, so I hope you’ll excuse my brevity on the subject.

Procedural[i] textures: I’ve had some experience in the past writing procedural textures. I’ve explored 2D/3D Perlin noise for texturing 2D and 3D objects respectively, but never 4D noise. I think that the nature of the world I’m constructing, 4D noise will come in handy to model the continuous changing noisy appearance we see of clouds. Perhaps not. If not, it might still be possible to use 4D noise to model some neat feature artifact of the world.

A height-map texture file or procedurally generated height map can be used for bump-mapping objects. And similarly, it can also be used for terrain generation, though for this purpose, it might be necessary to specify more than 1 map. Each object that can accept a height-map will then have its own procedure to determine exactly how to apply it to itself. For example, spheres will require a different interpretation of a square height-map than might rectangular prisms due to their topological differences.

Objects will have an index of translucency from [0,1]. 0 represents complete opacity whereas 1 represents complete transparency. This might be useful in modeling things like glass.

Objective List:

1. Bump[ii] mapping
2. Texturing
3. Screen-space Ambient Occlusion[iii]
4. Particle effects, animated
5. Procedural terrain modeling via combinations of noise and blending functions[iv] [v] [vi] [vii] [viii] [ix]
6. Procedural architecture via CFG
7. Camera animation on splines
8. AI pathing
9. Reflections on water surfaces
10. Trees

Declaration:

I have read the statements regarding cheating in the CS488/688 course handouts. I affirm with my signature that I have worked out my own solution to this assignment, and the code I am handing in is my own.

[i] Procedural Content Generation Wiki,
http://pcg.wikidot.com/

[ii] Blinn, James F. "Simulation of Wrinkled Surfaces", Computer Graphics, Vol. 12 (3), pp. 286-292 SIGGRAPH-ACM (August 1978)

[iii] Screen Space Ambient Occlusion Shader
http://www.coniserver.net/wiki/index.php/Screen_Space_Ambient_Occlusion_Shader

[iv] Implementing Improved Perlin Noise (GPU Gems, by S Green),
http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter26.html

[v] Implementing Improved Perlin Noise (GPU Gems, by K Perlin),
http://http.developer.nvidia.com/GPUGems/gpugems_ch05.html

[vi] My own work in procedural textures,
http://anthonycameron.com/lab#texture

[vii] Terrain, Part 1,
http://www.shamusyoung.com/twentysidedtale/?p=141

[viii] Minecraft Like Rendering Experiments in OpenGL 4,
http://codeflow.org/entries/2010/dec/09/minecraft-like-rendering-experiments-in-opengl-4/

[ix] More on Minecraft-type world gen,
http://www.gamedev.net/blog/33/entry-2227887-more-on-minecraft-type-world-gen/