

COLORADO SCHOOL OF MINES

CSCI 444

ADVANCED COMPUTER GRAPHICS

Project Proposal

Author: Robinson Merillat

Supervisor: Dr. Jeffery Paone

May 8, 2018



Contents

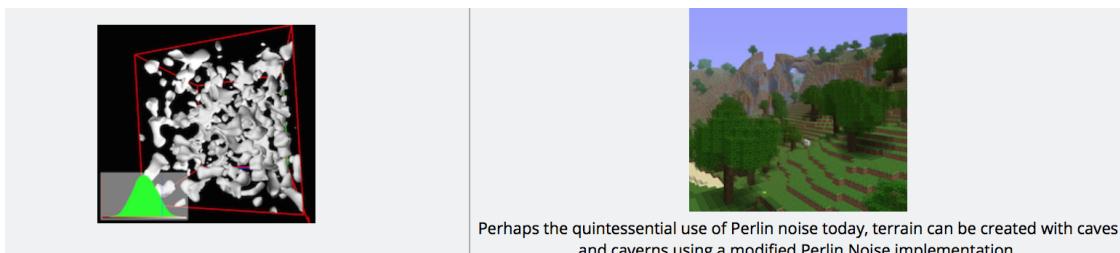
1	Summary	2
2	Research into the Topic	2
2.1	Height Mapping	3
2.2	Goals/ Stretch Goals	3
3	Technical Challenges	4
3.1	Multiple noise/generation algorithms simultaneously	4
3.2	Minimising Shaders	4
3.3	Graphical Techniques	4
4	Examples	5

1 Summary

The world builder is a random terrain generator base upon a random selection of an ecosystem. Several different ecosystems could be generated such as mountains, desert, caverns, and islands. This terrain will be populated with various flora and if time permits, fauna as well (possibly animated, but non-moving).

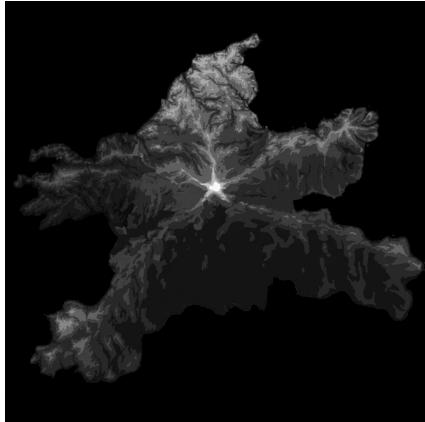
2 Research into the Topic

One of the largest users of procedurally generated terrain is the Video Game Industry. A multitude of games implement procedural terrain techniques such as Minecraft, Spore, Civilization, Diablo, and Left 4 Dead just to name a few. These techniques are even used in film such as in the *MASSIVE* engine used in Peter Jackson's The Lord of the Rings to generate significantly larger combatting armies than cast that was at his disposal. Each variation of procedural generation utilizes a different algorithm in order to simulate the real world. Several of these algorithms implement various uses of generated noise such as in the simplex noise algorithm while others use various clumping or boring algorithms such as in the case of generating forests and caves.



2.1 Height Mapping

One of the main basis for procedurally generated terrain is the use of heightmapping. As opposed to procedurally generating geometry, heightmapping procedurally generates a black and white texture. When a plane is sent through a vertex shader, the y value (vertical) of each vertex is then set to be some scaled value corresponding to the shade (0-255) of the texture at that location, thus generating a unique terrain that ultimately saves an immense amount of computation time. On top of this, normal mapping techniques are often used along with several other texturing techniques to make a realistic looking environment.



2.2 Goals/ Stretch Goals

Wrapping this back around to this project, the overall goal of this project is to be able to utilize all of these graphical techniques to generate procedural interesting and semi-realistic looking terrain of various regions of our, and even fantastical worlds. Some stretch goals I would like to look into should I complete the basis of this project would be to add models such as trees, rocks, and grasses using grouping algorithms.

3 Technical Challenges

3.1 Multiple noise/generation algorithms simultaneously

The algorithms that are used to generate heightmaps for a mountain or island are not the same as what would generate a tundra or desert. similarly, ice holds a different texture than cracked dry earth. That being said, there are many similarities. Exploring those similarities and how I may be able to utilize aspects of each of them to compute the textures is a technical challenge that I will surely face.

3.2 Minimising Shaders

It's easy to get carried away with shaders and start to implement a new one for every new thing you create, however, there is a limit to the number of shaders that can be used. In addition, why recreate a similar shader for a new object when you may be able to accomodate it in the one that currently exists. One of the challenges I will face is determining when and where it is appropriate to create a new shader.

3.3 Graphical Techniques

Beyond the sheer monstrosity that this project is sure to become, it will be challenging to implement so many generation techniques simultaneously. I will be using a full range of techniques that I learned throughout the course of the semester and it will challenge me to make something spectacular.

Of the techniques learned in this course, I plan on using:

- UBO's
- procedural noise
- deferred shading
- tessellation shaders to minimize geometry based on proximity
- heightmaps

(note: this is subject to change if I learn something new that I would like to use after beginning this project)

4 Examples

1. WebGL Procedural Terrain
2. Anime Concept of Procedurally Generated Floors/Dungeons/Quests
3. Minecraft world generation