Procedural map generation using OpenGL

James Hopkins

University of South Wales - BSc Computer Games Development

Introduction

This project looks into several procedural map generation algorithms to create a tool that generates scenery for use in other projects.

Procedurally generated content is widely used in video games to create a sense of realism. This concept can be applied to a map for developers to gain procedurally generated maps that appear natural and unique.

For example, Mojang uses a modified version of the Perlin noise algorithm to create an open world experience that generates different consistently maps.

Research Procedural Algorithms

The Perlin Noise algorithm was created by Ken Perlin, who created the algorithm whilst working on the movie Tron. The algorithm creates a grid. Each point in the grid has a random gradient vector applied to it. The neighbouring grid point then takes the dot product, applies the difference between the pixel and the neighbouring point and contributes it to the pixel.

The Diamond Square algorithm takes the four corners of the terrain to fill the terrain with midpoint displacement values. Consisting of two steps, the algorithm continues until all values are filled. The diamond step takes the four corners and averages them (with the addition of a random value) to create a diamond shape. The square step averages the values from the diamond step, the centre value becomes part of the next diamond step.

Results

The resulting project is a tool that flips between a terrain made using Perlin noise, and Diamond Square. Upon pressing the 'F' key the seed value will change and the terrain will be reproduced. Keys '1' and '2' change the procedural generation algorithm used.

The Perlin noise plane works as expected, the terrain appears smooth and realistic. Perfect for addition of models and textures.

The Diamond Square algorithm did not function as expected. The terrain does not obtain the corners of the array properly so the average fluctuates.

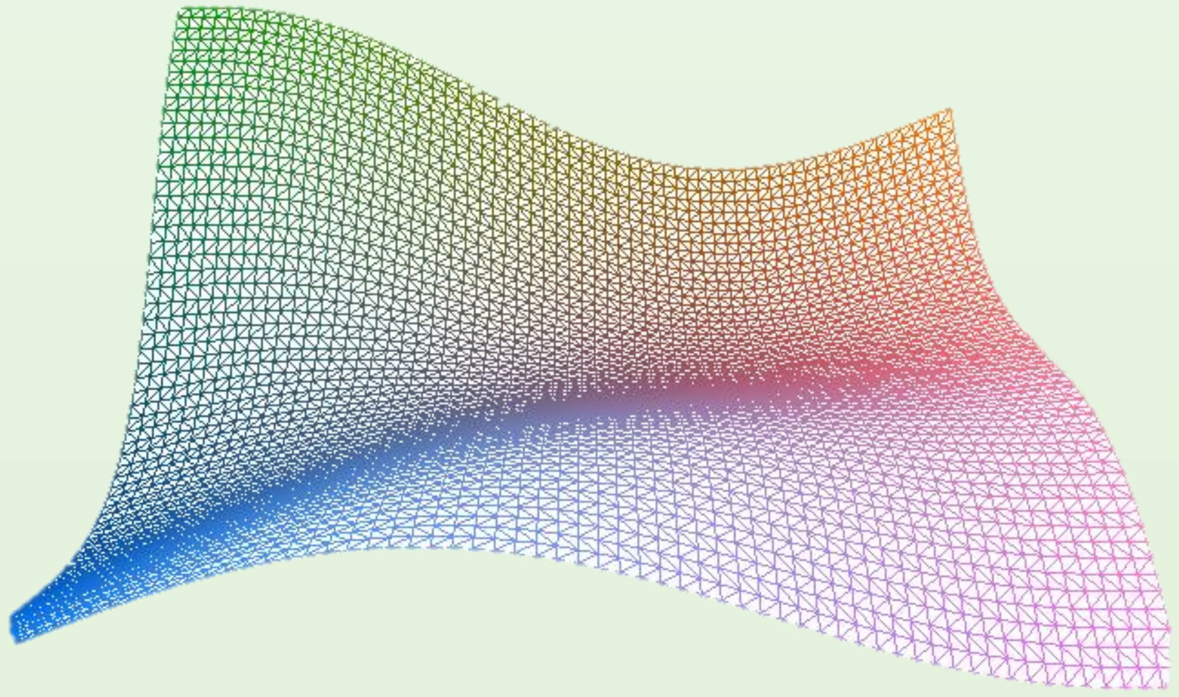

Figure 1 – A generated plane using Perlin Noise, Seed value 80

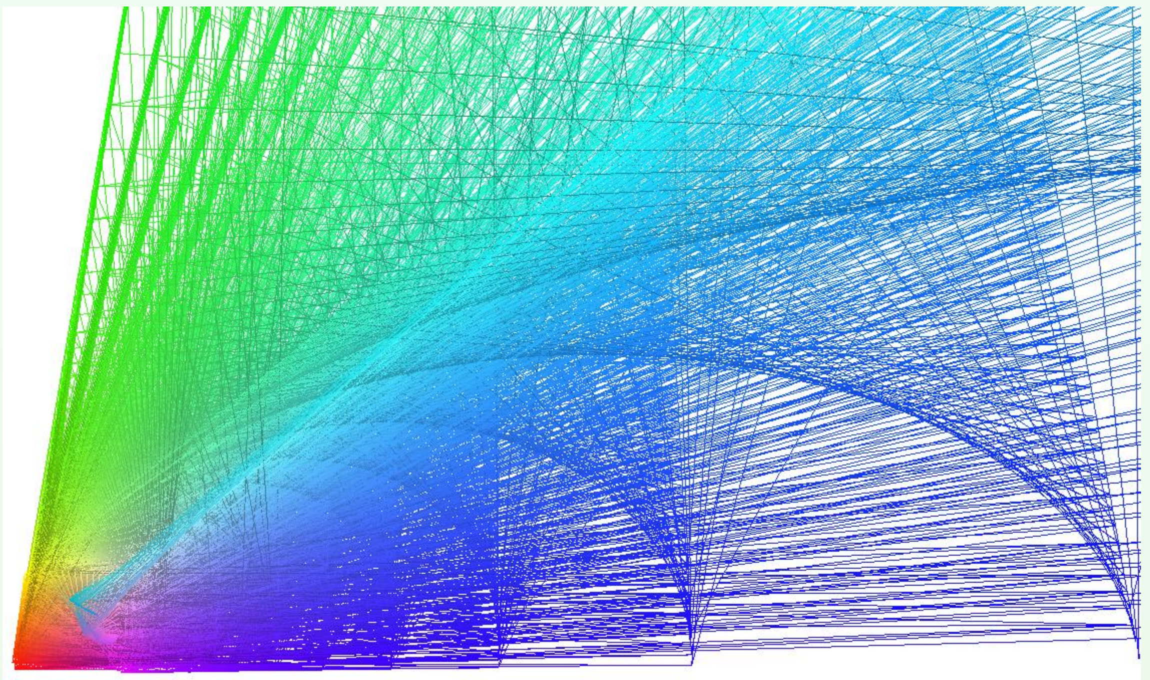

Figure 2 – A generated plane using Diamond Step, Seed value 239

Conclusions

The aim of the project was to create a tool that creates a terrain plain with procedural values. The objectives set for the project were mostly completed. However, the ability to export the map was not implemented, also example models were not added to the terrains.

The testers of the program found that the terrain was displayed well and very informative, though they noted that there wasn't much information to speculate.

References

Understanding Perlin Noise. (2014, August 09).

Retrieved from Adrian's Soapbox: https://flafla2.github.io/2014/08/09/perlinnoise.html

Gustavson, S. (2005). *Simplex noise demystified.* Linkoping.

Perlin, K. (2002). Improving Noise. *https://mrl.nyu.edu/~perlin/paper445.pdf*, 2.

Segal, M., & Akeley, K. (2010). *The OpenGL Graphics System: The Specification.* The Khronos Group .

Archer, T. (2011). Procedurally Generating Terrain. 2.