Supervisor: Dr. Julien Cordry

Clariana Justes, Oriol Marc

V8086261

Procedural Terrain Generation in UE4

BSc. Computer Games Programming

Contents

Abstract 2

Acknowledgements 2

Introduction 2

Methodology 2

Research 2

Perlin Noise 2

Level of Detail 3

Tile / Chunk 4

Design 4

Implementation 5

Optimization 5

Testing 5

Conclusion 5

Bibliography 5

# Abstract

Procedural Generation is a feature becoming more popular each year in the videogame industry and for the infinity possibilities that it grants. In this project, using Unreal Engine 4 and the Procedural Algorithm Generation it has been created a Tool with a variety of options to customize the final result used in conjunction with a series of techniques to create high quality procedural terrain in real-time and pre-bake.

To showcase this, an easy-to-understand and well-structured interface was built and all necessary elements to visualize the product were created. This report explains the classes that were built in order to visualize the procedural terrain, how it is generated and the parameters to customize the terrain.

# Acknowledgements

I would like to thank to my project supervisor Julien Cordry who provided support and advice on the product all throughout the development of this project. I would also like to thank my peers and friends on the course who provided support and feedback during the development process.

# Introduction

# Methodology

# Research

## Perlin Noise

Perlin noise is a type of gradient noise developed by Ken Perlin in 1983. The most common used ones are one, two and three dimensions. The frequency of the noise controls the distance at which samples are taken and it is used to control the scale. The amplitude controls the maximum and minimum values that the noise can take. In graphics programming Perlin noise can be used to generate two dimensional textures which later can be used as heightmaps for terrain generation or as a regular texture.

Heightmaps can be used to describe terrains by interpreting the values of the noise texture as heights. Perlin can be effectively used to generate a basic random terrain. From here, different approaches can be taken to modify the Perlin noise equation so as to make a terrain feel more realistic.

|  |  |
| --- | --- |
| Imagen relacionada | Resultado de imagen de perlin noise |
| Figure 1 | Figure 2 |

The Figure 1 show a regular sample of the 2D texture of Perlin Noise.

It is possible to make the result of Perlin Noise more realistic / different by iterating several times over its values using different types of algorithms such as Fractals or Hydraulic erosion.

## Level of Detail

The Level of Detail (LOD) involves decreasing the complexity of a 3D model representation as it moves away from the viewer or according to other metrics such as object importance, viewpoint-relative speed or position.

This technic is used to optimize the 3D environments to reducing the complexity of the models at different distances changing the number of triangles rendered an object as its distance from the camera increases. It allows reduce the consuming a huge number of resources on models far of the camera vision.

|  |
| --- |
| Resultado de imagen de level of detail camera distance |
| Figure 3 |

As seen in Figure 3, the LOD change the density of polygons is changed but also trying to keep the initial shape of the object. This technique can be used in terrain rendering due to the fact in a scene that is viewing a terrain, both close and far terrain can be viewed at the same time and do not require to have the same polygon quality. The further away the objects are the easier it is to describe with lesser polygons than the initial shape. And this simple concept allows to save resources at the rendering moment.

## Tile / Chunk

The Tile/Chunk involves split the terrain generation in small pieces to reduce the calculation time and the number of polygons rendered.

Combining with Perlin Noise and the Level of Detail (LOD) makes possible generate an infinity number of pieces of the terrain using the Perlin Noise value and make it works faster reducing the LOD on the far chunks.

Also allows the possibility to use a Pooling System to reduce the number of chunks on the world modifying the chunks not rendered. In this way allowing to avoid creating more memory for the generation of the land using the memory already created but not used.

# Design

# Implementation

# Optimization

# Testing

# Conclusion

# Bibliography