

School of Computing

FACULTY OF ENGINEERING AND PHYSICAL
SCIENCE



UNIVERSITY OF LEEDS

Final Report

Fractal Terrain and Erosion for Game Engines

Samuel Dean Thornton

**Submitted in accordance with the requirements for the degree of
Bsc Computer Science**

2019/2020

40 credits

The candidate confirms that the following have been submitted:

Items	Format	Recipient(s) and Date
<i>Final Report</i>	<i>Report</i>	<i>FILL IN</i>
<i>Source Code</i>	<i>FILL IN</i>	<i>FILL IN</i>
<i>Instructions</i>	<i>FILL IN</i>	<i>FILL IN</i>

Type of Project: Exploratory Software

The candidate confirms that the work submitted is their own and the appropriate credit has been given where reference has been made to the work of others.

I understand that failure to attribute material which is obtained from another source may be considered as plagiarism.

(Signature of student)

Summary

Generating realistic 3d terrain for graphical purposes is a complex problem making use of mathematical algorithms and simulated geographical phenomena. Landscapes are fractal by nature, as described in the report, and thus fractal algorithms can be used to model them, and operations can then be performed in order to make the landscapes more realistic before rendering. This project aims to cover methods of generating said landscapes, with a specific focus on use in video game and graphics engines.

NOTE: formatting doesn't seem to be working on my laptop so atm only the structure and part of the content is there.

<

Reminder about basic requirements of layout and format:

The report must be in typescript, sequentially page numbered, on A4, single or double-sided, with 1in margins. Point size 11 and one-and-a-half line spacing should be used.

Page Numbering: The pages preceding the body of the text, i.e. from "Summary" to "Contents" inclusive, should be sequentially numbered in Roman numerals. All the remaining pages should be numbered in a single sequence of Arabic numerals.

Length: This report should be no more than 40 pages excluding appendices, references and front material (title page, summary, table of contents etc.)

>

Acknowledgements

NONE YET

Table of Contents (example of how to format)

Summary.....	i
Acknowledgements.....	ii
Table of Contents.....	iii
Chapter 1: Introduction.....	1
1.1 Introduction.....	x
1.2 Aim	x
1.3 Objectives.....	x
Chapter 2: Background Research.....	x
2.1 Terrain Generation.....	x
this is inaccurate cause its hard to format on Linux	
2.2 Fractal Terrain.....	x
subheadings woo	
2.3 Erosion.....	x
2.4 rendering terrain	

Chapter 3: Design and Planning.....	X
3.1 System Requirements	
3.2 Diamond-Square Algorithm	
3.3 Fast Fourier Transform	
3.4 Rendering the Landscape	
3.5 UI design and implementation	
3.6 Plan	
3.6.1 Project methodology	
3.6.2 Plan and milestones	
3.6.3 Risk Mitigation	
 Chapter 4: Implementation.....	 X
TODO	
Chapter 5: Evaluation.....	X
5.1 Requirements Met	
5.2 Performance and Tests	
5.3 Issues and Bugs	
 Chapter 6: Conclusion.....	 X
6.1 Aims and Objectives	
6.2 Further Work	
6.3 Legal, Social, Ethical and Professional Issues	
6.3.1 Legal	
6.3.2 Social	
6.3.3 Ethical	
6.3.4 Professional	
6.4 Personal Reflection	
 List of References.....	 X
 Appendices.....	 X

Chapter 1

Introduction

INSERT PICTURE OF TERRAIN HERE

1.1 Introduction

Terrain is an integral part of artificial world generation, as seen in movies and video games. The accurate simulation of terrain and erosion is also useful for study and predictions in a geographical discipline. Perhaps the most widely known example of computer generated terrain is the videogame 'Minecraft', which creates an entire randomly generated world.

The method used to compute artificial terrain often provide artificial and unrealistic looking landscapes, and while this can be useful for certain implementations, such as science fiction, geologists who wish to simulate accurate terrain cannot simply use fractal algorithms. The goal of this paper is to generate realistic looking terrain. This can be done via colour, lighting or adding plantlife and such, however this paper focuses on the abstract shape of the landform itself, via using fractal algorithms and simulated erosion.

Landscapes are inherently fractal, in that as one goes into deeper levels of detail, they become self-similar. This allows to represent landscapes as fractal on a computer, and define them using fractal algorithms on a 2d array to generate a heightmap. This heightmap can then be displayed as a 3d model, using the position of a value in the array and the x and z coordinates, and the value in that position as the y coordinate. This project in particular aims to observe and analyse modern methods for generating terrain, and to produce software that allows for ease of generating custom landscapes which can then be exported and used for any number of other projects.

1.2 Aim

The main aim of this project is to provide a computer program which can generate fractally generated terrain. Several different algorithms and methods will be researched and implemented. The terrain itself will then be rendered in realtime, and there will be an option to export the terrain for use elsewhere.

1.3 Objectives

- Research Fractal Terrain Algorithms
- Research Simulated Erosion Algorithms
- Implement the two in conjunction
- Render the generated terrain using OpenGL
- Provide an accessible user interface to generate terrain using differing parameters

Chapter 2

Background Research

2.1 Terrain generation

Synthesised terrain is required in many disciplines, from simulation to video games to movies. There are several methods for creating such terrain; Spectral synthesis, Midpoint displacement algorithms, Voronoi diagrams and Fast Fourier Transforms to name a few. While we will cover these in brief, in this paper we will focus on the Fast Fourier Transform and Midpoint Displacement methods, in particular the Diamond-Square algorithm. These two methods are fundamentally different, in that the Fourier Transform Method uses discrete Fourier Transforms on random noise to define a landscape, essentially converting frequencies into a cohesive landscape. The Diamond-Square algorithm however takes a set of predefined values on a heightmap and applies the same operation on them, then repeats this at progressively smaller scales, using random numbers to provide variation.

2.2 Fractal Algorithms

In 1967, Benoit Mandelbrot published a paper titled “How Long Is the Coast of Britain? Statistical Self Similarity and Fractional Dimension” [1]. **This paper discussed the coastline paradox; the given length of a coast would depend on what scale was used to measure it, and thus the length of said coast is not well defined.** This is due to the fact that coastlines have fractional dimensions; they are not one-dimensional. This means that a coastline is fractal. The same property applies to landscapes, and it is this property that allows us to represent them as fractals.

2.2.1 Diamond Square

The Diamond-Algorithm is 3D adaptation of the 2d Midpoint-Displacement method. The midpoint displacement method is as follows:

1. **Find the midpoint of a line on a 2d plane**
2. **Move this point up or down by a random value**
3. **Repeat on the newly created lines**

INSERT DRAWN MIDPOINT DISPLACEMENT FIGURES

The Diamond-Square algorithm adapts this method to work in 3 dimensions, by first assigning random values to the corners of an array. These 4 values are then averaged and changed

by a randomly assigned value. This value is then given to the midpoint of the four corners – The Square Step. This step creates 4 'Diamonds' as shown in figure 2.2.1. The corners of these diamonds are then averaged and randomly shifted, and assigned to the midpoints of that diamond. This is the Diamond Step, and creates a series of squares on the grid with which the algorithm can be run again on a smaller scale, using a smaller random perturbation. This is repeated until the desired level of detail is acquired.

Figure 2.2.1 CREATE EXAMPLE ON PC

The Diamond-Square algorithm creates fractal landscapes and is relatively simple to implement, however the landscapes generated are sometimes not very realistic. .

2.2.2 Fast Fourier Transformation

IN PROGRESS

2.2.3 Multiplication

2.3 Erosion

headings for different types god this formatting thing is annoying

Design and Plan

Implementation

Evaluation

Conclusion

List of References

<It is expected that the list would reflect the breadth and depth of scholarly research undertaken by the student during the course of the project.>

[1] Mandelbrot, B, https://en.wikipedia.org/wiki/How_Long_Is_the_Coast_of_Britain%3F_Statistical_Self-Similarity_and_Fractional_Dimension, 1967

Appendix A

External Materials

<Level 1 Heading with 'heading 1' Style Applied by Pressing Ctrl Shift 1> Text under appendix heading. Text under appendix heading. Text under appendix heading. Text under appendix heading. Text under appendix heading. Text under appendix heading.

A.1 Level 2 Heading with 'heading 2' Style Applied by Pressing Ctrl Shift 2

Text under level 2 heading. Text under level 2 heading. Text under level 2 heading. Text under level 2 heading.

A.1.1 Level 3 Heading with 'heading 3' Style Applied by Pressing Ctrl Shift 3

Text under level 3 heading. Text under level 3 heading. Text under level 3 heading. Text under level 3 heading.

A.1.1.1 Level 4 Heading with 'heading 4' Style Applied by Pressing Ctrl Shift 4

Text under level 4 heading. Text under level 4 heading. Text under level 4 heading. Text under level 4 heading.