Real-time Rendering of 3D “Fractal-like” Environments

Deliverable 1: Final Year Dissertation

Solomon Baarda

Solomon Baarda  [Company address]

# Abstract

The aim of this project is to develop a prototype rendering engine, capable of displaying complex 3D scenes containing fractal-like objects in real-time. The performance of the engine will be benchmarked across various systems to determine whether the “real-time” aspect of the project has been achieved.

I, Solomon Baarda confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:

# Table of Contents

[Abstract 1](#_Toc85450323)

[Table of Contents 3](#_Toc85450324)

[1 Introduction 4](#_Toc85450325)

[1.1 Aims & Objectives 4](#_Toc85450326)

[1.2 Project Description 4](#_Toc85450327)

[1.3 Scope 4](#_Toc85450328)

[2 Literature Review 4](#_Toc85450329)

[2.1 Fractals 4](#_Toc85450330)

[2.2 Ray Tracing 5](#_Toc85450331)

[2.3 Ray Marching 5](#_Toc85450332)

[2.3.1 Signed Distance Fields 6](#_Toc85450333)

[2.3.2 Primitives 6](#_Toc85450334)

[2.3.3 Alterations & Combinations 6](#_Toc85450335)

[2.3.4 Collision detection 6](#_Toc85450336)

[3 Requirements Analysis 6](#_Toc85450337)

[3.1 Use Cases 6](#_Toc85450338)

[3.2 Requirements Specification 7](#_Toc85450339)

[4 Software Design 7](#_Toc85450340)

[4.1 Structure 7](#_Toc85450341)

[4.2 Technologies 7](#_Toc85450342)

[5 Evaluation Strategy 7](#_Toc85450343)

[6 Project Plan 7](#_Toc85450344)

[6.1 Design Methodology 8](#_Toc85450345)

[6.2 Legal, Ethical & Social Issues 8](#_Toc85450346)

[6.3 Risk Analysis 8](#_Toc85450347)

[6.4 Timetable 8](#_Toc85450348)

[7 References 8](#_Toc85450349)

[8 Appendices 8](#_Toc85450350)

# Introduction

## Aims & Objectives

Motivation

## Project Description

Is the topic meaningful, complex and challenging?

## Scope

# Literature Review

How relevant is the literature that is covered?

• Is there missing material?

• Is it well structured?

• Are good quality sources used and properly cited?

• How strong are the comparative and critical aspects?

• Is the literature review of an appropriate length?

## Fractals

In mathematics, a fractal is a complicated pattern built from simple repeated shapes, which are reduced in size every time they are repeated. <https://dictionary.cambridge.org/dictionary/english/fractal> These shapes are self-similar, though not often symmetrical.

The idea of fractal geometry appeared in the late 1970s, inspired through the work of Benua Mandelbrot and his book “Fractals: form, chance and dimension”, released 1977. This book introduced the concept of a “fractal dimension”, a measure of the complexity of how the detail in a pattern will change in respect to the scale at which it is measured.

How they are calculated – running sum etc

3D fractals

## Ray Tracing

Ray tracing is a technique used to render 3D environments, where one or more “rays” are sent out

Ray tracing is a method of rendering 3D environments, often with photorealistic detail. In ray tracing, a ray (simply a line in 3D space) is extended until it collides with the surface of an object. From there, the ray can be absorbed or reflected by the surface, taking into consideration light absorption, reflection, refraction, and fluorescence. Ray tracing is a computationally expensive

of each pixel of the camera.

Background of ray tracing

Well suited for polygon meshes, or simple primitives

Ideal for very realistic graphics

not real-time

hybrid approaches can be used for real-time

Unsuitable for self-repeating shapes, infinite precision, non-Euclidean geometry where an object can be seen multiple times from the same view

## Ray Marching

Ray marching is a variation of ray tracing, which differs in the method of detecting collisions between rays and objects. Ray tracing will calculate the exact point of contact between a ray and an object’s surface using a collision function. Ray marching is instead when we march or move along a ray in small increments, until we collide with an object. This may sound more computationally expensive since

benefits – simplicity, solution to problems above

infinite precision as no polygons

allows more complex 3d shapes

Use ray marching optimisation techniques

Ray marching – lots of effects (ambient occlusion, hard or soft shadows, glow, fog) are free

To determine how feasible a real-time rendering of 3D environments using ray marching.

### Signed Distance Fields

Distance to the closest object

-, +, 0

### Primitives

Sphere, box, plane, etc

Deriving an SDF

### Alterations & Combinations

Elongation, rounding, “onioning”

Translation (position), rotation and scale, Symmetry

Infinite or finite repetition

Union, subtraction, intersection

Surface displacement (using an equation, say noise function or sin wave)

### Collision detection

Collision detection is possible

Marble marcher

# Requirements Analysis

## Use Cases

## Requirements Specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Requirement ID | Description | Priority | Justification |  |
|  |  | MUST |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Software Design

## Structure

## Technologies

# Evaluation Strategy

# Project Plan

## Design Methodology

## Legal, Ethical & Social Issues

A well-researched consideration of any Professional, Legal, Ethical, and Social Issues pertinent to the project. (e.g. codes of conduct (BCS), codes of practice, standards, computer law, ethical decision making, intellectual property, social aspects, copyright, data protection, and so on)

## Risk Analysis

## Timetable

# References

# Appendices