

KV6003: Individual Project

Project Terms Of Reference

Cuthbert Mutumbwa
Design and Implement a Rendering Engine

November 23, 2018

1 Background

Creating A rendering engine is a task that requires one to understand multiple different topics and subjects, for example not only will I need to know how lighting works, but how it interacts with other objects, and how to represent it on screen. I will also need to learn how textures work, how to wrap them around an object and how the light will interact with the texture wrapped around the object. I will also learn how a modern GPU, by writing code for it, as well as understand what role the CPU will have when drawing the 3D images.

In college, I used two pieces of software that are called Unreal Engine and Blender. Unreal Engine was a game engine, and Blender was a 3D Creation suite that can be used as a rendering engine. My first question when using this software where "How does one Create 3D graphics software", then I realized I also was unsure of how someone designs this software.

I want to create a rendering engine that uses core rendering techniques and features, and Software engineering practices to design the engine. for the design aspect, I have opted to use different types of literature such as Designing a Modern Rendering engine [?] to help me. I will also use literature to choose which type of technique I will use to implement my features, this is because there are multiple different ways to implement features such as shadows, lighting, and even the way the engine will render a scene.

Rendering engines are still or more specifically rendering is a very interesting field of study. New rendering features are being made to improve quality in 3D animated films, and recently Nvidia released a new Graphics Card to allow Real-Time raytracing in Video Games¹. Such an Achievement is one that is truly impressive as raytracing is known for being computationally expensive.

Creating a rendering engine is not only about knowing how to write good and efficient code, and knowing how lights, shadows, and texturing works are not enough. I will need to understand the underlining mathematical principles of computer graphics such as vectors, matrices and more. I will need to be able to read mathematical notation, as well as choose which formulas I will use to achieve the results I want. I want to improve my mathematical skills, as well as know, learn what types of maths is needed to achieve this I have picked out literature and books such as Game Physics Cookbook [?]. This book teaches the reader the maths needed to make the rendering engine.

¹<https://www.pcgamer.com/uk/what-is-ray-tracing/>

2 Proposed Work

Writing a literature review will be something I shall do, I will then do some more analysis by doing a Requirements analysis and specification. For the design phase I will use Design Documents to help me conceptualize the rendering engine, and use them as a reference when creating the product. The analysis phase should allow me to further justify my choices when creating the renderer. I will create a 3D scene environment that will showcase the features I plan on implementing. The user will be allowed to freely move around the 3D scene environment. The user will see different showcases on features within this scene, for example, there may be a location where the user is able to see an object reflection from another object, and a location that demonstrates shadows by casting a light above the object. Since the user will have very limited interaction with the software I will not need to create any interface or HCI work.

Since there are many types of ways to implement one feature, in my design phase I have to consider what the best way to implement a specific feature will be. I need to make sure that it fits my criteria, and that I am able to implement the feature, while also making sure it is in scope. The analysis will be an essential part of not only the design phase but the implementation phase as well. To Aid me I will use books, papers, and online tutorials to help me learn. I will also create a small prototype of the project while in the design phase, this will again be simply used to help me learn, and will not be a part of the project.

To create the engine I will identify the programming language that I believe will be the most beneficial to the project. There are many programming languages that support many different APIs (Application Programming Interface) that will help in the creation of the engine. An example of programming languages I can choose are C

C++, and the OpenGL API. There are many more, and I should have chosen one when doing the analysis part of my project. I will also create a test plan, to test the rendering engine, and to identify anything that may need fixing.

3 Aims and Objectives

Aims

- To design and build a rendering Engine using modern design principles

Objectives

1. Literature Review
2. Requirements analysis and specification
3. produce design documents
4. Design a rendering engine
5. Build a rendering engine
6. Test the rendering engine.
7. Evaluate the rendering engine
8. Write Report

The 3 UML Diagrams I will be creating will be Class Diagram, Sequence Diagrams, and State Machine Diagram. The 8 Features I will implement will be showcased within the scene environment. For testing and debugging I will generate a test plan.

4 Skills

1. Programming in C/C++ (from module KF4006)
2. Software Engineering (from module KF5012)

3. Algorithms (from module KF5008)
4. OpenGL API (Learnt enough to build a 3D scene)
5. Computer Graphics Theory (Currently Learning in Module KF6018)

I am confident I have a good foundation for each of these skills, however, I have many resources such as books that will not only aid me but further improve my knowledge on the subject.

5 Resources

Hardware

- Computer

Software

- Visual Studio
- Nvidia Nsight

Visual Studio will be the IDE that will allow me to write the rendering engine code. Nvidia Nsight is a graphics debugger and will not only help me debugging the code but to also help me with the test plan.

6 Structure and Contents of Project Report

Planned Chapters

1. Literature Review(Analysis)-
2. Requirement analysis and specification(Analysis)-
3. Designing the Renderer (Synthesis)-This Chapter will explain what techniques and tools did I use to design the renderer.
4. Implementing the Rendering Engine (Synthesis)-This Chapter will explain how I actually created the renderer. It will allow me to justify my choices, as to why I chose to implement certain features the way I did. It will also allow me to show my understanding of the topic.
5. Evaluating the Rendering Engine (Evaluation)-I will furthermore justify my choices as well as evaluate on my work. I will talk about what I believe I did right, and what I would have done differently.
6. Conclusion on the Rendering Engine(Evaluation and Conclusion)

7 Marking Scheme

Project Type Software Engineering projects

Project Report Chapters 40%

Analysis

- Requirement Analysis and specification
- Analysing Rendering Techniques

Synthesis

- Designing the Renderer
- Implementing the Rendering Engine

- Creating the Scene environment
- Testing and

Evaluation

- Evaluation and Conclusion of the Rendering Engine

Product Deliverables list

1. Requirement analysis and specification
2. Source Code
3. Design Documents
4. Test Plans and Results

The design document includes anything related to design such as UML diagrams, requirement analysis etc.

Fitness of purpose and Build quality Lists

Fitness of Purpose 40%

1. Quality of Functionality
2. Meeting of Requirements as identified during project

Build Quality 60%

1. Requirement analysis and specification
2. Code Quality
3. Software is responsive
4. Test plan and results
5. Design documents were created correctly

The code quality must have multiple classes, and each class corresponding with the class diagram. Software must not stutter when the user attempts to move the camera view around the scene. Design documents must not use wrong notation. Test plans must be thorough and show which problems managed to be fixed

Precentage Breakdown

Fitness for Purpose: 40%

Build Quality 60%

8 List of Appendices

1. Terms Of Reference
2. Finalised UML Diagrams
3. Test Plan
4. screenshots of the Rendering Engine

Having screenshots of the rendering engine will allow me to show and further explain certain aspects of my engine. Showing my finalized UML diagrams will allow me to show how helpful they were in the creation of the software.

9 Project Type

This will be a Software engineering project

10 Project Plan

Signature Apporval Please Sign Below

Supervisor (Alun Moon): _____

2nd Marker (Lynne Conniss): _____