|  |  |
| --- | --- |
| UFCFS4-30-3 Creative Technologies Project Proposal Document | |
| Student Name: | Alex Feetham |
| Student Number: | 17016942 |
| Project Title: | CPU Path Tracing |

# Description

This project will entail the creation of a path tracing system which will be able to render high quality images and will be run using the central processing unit, rather than the graphics processing unit. The project is to be comprised of a program, written in C++, which can import given three-dimensional objects and subsequently render an image of the scene in a pop-up window.

Path tracing is a computer graphics Monte Carlo method (Metropolis & Ulam, 2012) of rendering high-quality, realistic, three-dimensional images. Path tracing uses the Monte Carlo method to randomly sample a scene in order to produce a high-quality final rendered image.

**Deliverables:**

* Create an efficient software implementation of a path tracer using C++
* Implement global illumination
* Implement virtual cameras/scenes
* Allow the loading of 3D models into a scene
* Implement a variety of materials
* Implement sampling for ray generation
* Display the resulting high dynamic range images

# Research and Background

I got the idea for the project after building a ray tracing program for another module. While working on that project I found the subject area very interesting and as a result have decided on a similar area for this project.

### Ray vs Path Tracing

In ray tracing, a ray is fired out from the camera into the scene and is traced until it intersects with a solid object in the scene. From here, a ray is cast to each of the light sources in the scene to calculate the illumination and the surface shading is calculated for the intersection point of the 2 rays. Then if the object is transparent, the ray is sent out at an angle to simulate refraction, or the ray is sent back in the opposite direction is the object is reflective (Dunsterwald, 2016).

A path tracer builds upon this concept. In a path tracer, rays are distributed randomly within each pixel in camera space and at each intersection with an object a new reflection ray, pointing in a random direction, is generated. Once a ray has finished bouncing around the scene, sample value is calculated based on the objects the ray bounced against and the sample value is added to the average for the source pixel (Öqvist, 2015). The more samples per pixel, the higher the quality of image that will be produced. If the number of samples per pixel (SPP) was say 200 SPP, then it would merely produce an image that resembles a bunch of dots on screen. However, if the number of samples per pixel was more in the thousands, a real image would start to be produced. Hundreds of thousands of samples per pixel will likely be required for this project.

### The Rendering Equation

The rendering equation is integral in computer graphics. In the equation, the equilibrium radiance leaving a point is given as the sum of emitted plus reflected radiance under a geometric optics approximation. The rendering equation as of (Kajiya, 1986) is:

Where:

* is the related to the intensity of light passing from point to point
* is a “geometry” term
* is related to the intensity of emitted light from to
* is related to the intensity of the light scattered from to by a patch of surface at

# Objectives

### Project Objectives

* Investigate how to build a path tracer in C++
* Design an efficient software implementation of a path tracer
  + Including global illumination
  + And loading 3D models into a scene

### Research Objectives

For this project I will need to conduct research on how to:

* implement virtual cameras/scenes
* create a variety of materials
* implement sampling for ray generation
* display the resulting high dynamic range images

### Learning Objectives

From this project, I want to develop a broad understanding of how:

* path tracing works
* existing rendering systems implement path tracing
* to effectively implement a path tracing system

# Methods, Techniques, Tools and Processes

Simple and Fast Multimedia Library (SFML) provides a simple interface to the various components of your computer (Gomila, 2014). I will use SFML software in conjunction with Microsoft Visual Studio 2019. SFML will manage the window which the system will use to display its output. This means that I will not have to create a window from scratch using DirectX and can focus more on the path tracing system.

Almost all graphics programs have at least one class for storing geometric vectors and colours. In many systems, these vectors are 4D (3D plus a homogenous coordinate for geometry and RGB plus an alpha transparency channel for colours) (Shirley, 2020). This system is no different and shall also require a similar class system to be implemented to hold information for basic data structures, such as vectors, and simple geometry, such as spheres or cubes.

Loading of 3D models will be done using the open source library Open Asset Import Library (aka Assimp). Assimp allows the import of various popular 3D model formats like .obj, .blend and .3ds to name a few.

The project will need to make use of spatial data structures such as bounding volume hierarchies (BVH). A BVH is a volume which fully encloses (bounds) all the objects in a scene. By implementing a BVH the project will become vastly more efficient as it will only be checking for rays which hit the BVH.

Text

Description automatically generated

Figure 1: Pseudo code for a BVH

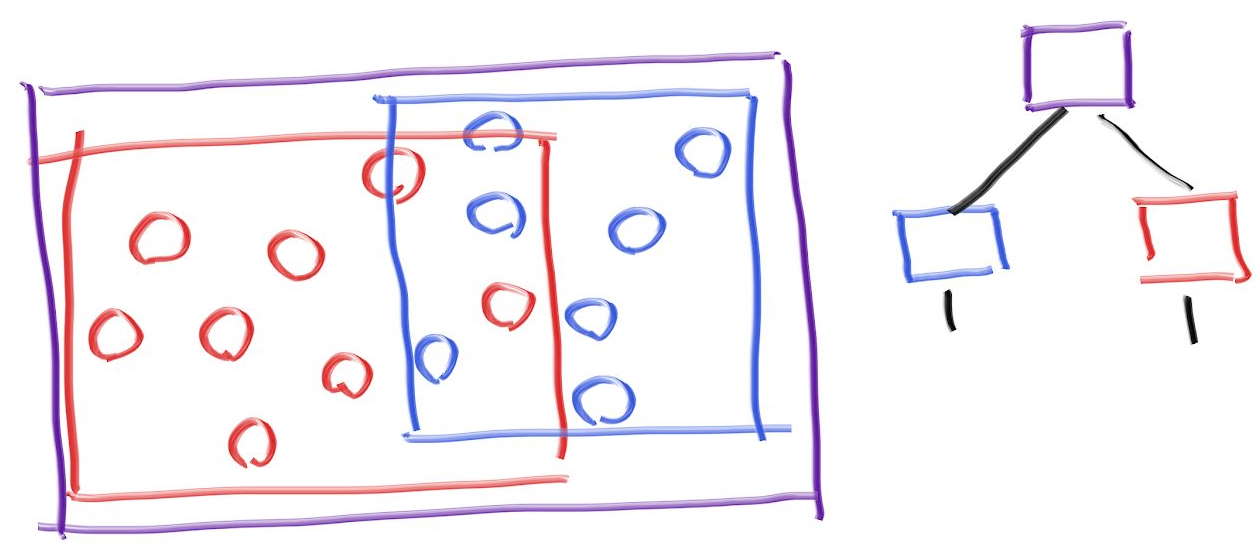


Figure 2: Bounding Volume Hierarchy

Lighting the scene will be a big task and will be done in small steps. Firstly, I intend to implement direct lighting using a single point light source. From there, I would seek to add direct light from area light sources, using the Monte Carlo method. Finally, moving towards indirect lighting (global illumination) by sampling reflected directions again using the Monte Carlo method.

The objects in the scene will utilise physically based rendering (PBR) shaders. These specialised types of shaders provide a more accurate representation of how light interacts with surfaces, which will be key given the lighting element of the project (Pharr & Humphreys, 2010). The materials that I will implement are:

* lambertian (matte)
* metal (reflective)
* glass (refractive)

To aid in management of the project, a gantt chart shall be made using Microsoft Excel. A gantt chart will allow for quick and easy visualisation of how the project is developing. It is also fluid and easy to adapt if the project is ahead/behind schedule.

# Risks and Issues

|  |  |  |
| --- | --- | --- |
| Risk | Mitigation | Contingency |
| Computer breaks and I lose my work | Save all work to OneDrive and GitHub | Restore from cloud saves |
| CPU overheating | Increase the cooling | Replace the CPU |
| Global Illumination not implemented | Start early and research basic principles and other implementations | Implement direct lighting instead |
| Loading 3D models not implemented | Use existing libraires to import the models rather than building from scratch | Create basic primitives in code |
| Project output window does not render correctly or takes excessive time | Set the window to be small so it has less pixels to render | Output rendered image as a .png or.jpeg |

# Specialist Resources and Support Required?

For this project I shall be using my personal desktop computer which is currently using an AMD Ryzen 7 2700x CPU with its included heatsink. AMD state that the processor has a maximum operational temperature of 85°C. If this is exceeded, I will require a better cooling solution such as a water-cooling system. Failing this I would need a new processor with more cores and higher clock speed to distribute the high workload.

# Sources and References

Dunsterwald, S., 2016. *Path Tracing vs Ray Tracing.* [Online]   
Available at: https://www.dusterwald.com/2016/07/path-tracing-vs-ray-tracing/  
[Accessed 11 November 2020].

Gomila, L., 2014. *Simple and Fast Multimedia Library.* [Online]   
Available at: https://www.sfml-dev.org/index.php  
[Accessed 18 October 2020].

Immel, D. S., Cohen, M. F. & Greenberg, D. P., 1986. A Radiosity Method for Non-Diffuse Environments. *ACM SIGGRAPH Computer Graphics,* 20(4), pp. 133-142

[Accessed 4 November 2020].

Kajiya, J. T., 1986. The Rendering Equation. *ACM SIGGRAPH Computer Graphics,* 20(4), pp. 143-150 [Accessed 4 November 2020].

Metropolis, N. & Ulam, S., 2012. The Monte Carlo Method. *Journal of the American Statistical Association,* 44(247), pp. 335-341

[Accessed 4 November 2020].

Öqvist, J., 2015. *Path Tracing.* [Online]   
Available at: https://chunky.llbit.se/path\_tracing.html  
[Accessed 11 November 2020].

Pharr, M. & Humphreys, G., 2010. *Physically Based Rendering: From Theory to Implementation.* Second Edition ed. Burlington, Massachusetts: Morgan Kaufmann Publishers

[Accessed 4 November 2020].

Shirley, P., 2020. *Ray Tracing In One Weekend.* v3.2.2 ed. s.l.:s.n

Figure 1 & Figure 2

Available at <https://raytracing.github.io/>

[Accessed 7 November 2020].

# Monthly Project Plan

|  |  |  |
| --- | --- | --- |
| October | Final proposal to be submitted by (12/11/2020)  Conduct research on different subject areas and narrow down project choices  Make choice of project and begin research | 35 days  14 days  2 days |
| November | Project decided and meeting with supervisor, begin writing proposal first draft  Finalise proposal ready for submission  \*lost a week due to contraction of Covid-19, proposal submission extended until 19/11/2020  Submit Final proposal (19/11/2020) | 7 days  4 days |
| December | Work on gantt chart  Work in progress documentation  Implement SFML into C++ project  Construct Vector Class  Construct Spatial Data Structures for Geometry | Throughout Project  Throughout Project  3 days  3 days  4 days |
| January | Implement lambert PBR  Implement simple point light  Research Report & Prototype Demo (14/01/2021)  Implement metal PBR  Implement glass PBR | 2 days  2 days  7 days  2 days  3 days |
| February | Project Demo (05/02/2021)  Implement BVH  Implement Assimp Library | 1 day  21 days  4 days |
| March | Implement indirect lighting (global illumination)  Test everything and prepare for final hand-in | 21 days  21 days |
| April | Hand-in (15/04/2021) | 1 day |
| May | Final Video (18/05/2021)  Viva (25/05/2021) | 3 days  1 day |



**Ethical Review Checklist for Undergraduate and Postgraduate Modules**

*Please provide project details and complete the checklist below.*

**Project Details:**

|  |  |
| --- | --- |
| **Module Name** | **Creative Technologies Project** |
| **Module Code** | **UFCFS4-30-3** |
| **Module Leader** | Michaela Palmer |
| **Project Supervisor** | Thomas Bashford-Rodgers |
| **Proposed Project Title** | **CPU Path Tracing** |

**Applicant Details:**

|  |  |
| --- | --- |
| **Name of Student** | Alex St John Feetham |
| **Student Number** | 17016942 |
| **Student’s email address** | [alex2.feetham@live.uwe.ac.uk](mailto:alex2.feetham@live.uwe.ac.uk)/[alexfeetham@outlook.com](mailto:alexfeetham@outlook.com) |

| **CHECKLIST QUESTIONS** | | **Yes/No** | **Explanation** |
| --- | --- | --- | --- |
|  | Does the proposed project involve **human tissue,** **human participants, animals, environmental damage, or the NHS.** | No | *If the answer to this is ‘No’ then no further checks in the list need to be considered.* |
|  | Will participants be clearly asked to give consent to take part in the research and informed about how data collected in the research will be used? |  |  |
|  | If they choose, can a participant withdraw at any time (prior to a point of “no return” in the use of their data)? Are they told this? |  |  |
|  | Are measures in place to provide confidentiality for participants and ensure secure management and disposal of data collected from them? |  |  |
|  | Does the study involve people who are particularly vulnerable or unable to give informed consent (eg, children or people with learning difficulties)? |  |  |
|  | Could your research cause stress, physical or psychological harm to humans or animals, or environmental damage? |  |  |
|  | Could any aspects of the research lead to unethical behaviour by participants or researchers (eg, invasion of privacy, deceit, coercion, fraud, abuse)? |  |  |
|  | Does the research involve the NHS or collection or storage of human tissue (includes anything containing human cells, such as saliva and urine)? |  |  |

Your explanations should indicate briefly for Qs 2-4 how these requirements will be met, and for Qs 5-8 what the pertinent concerns are.

* **Minimal Risk:** If **Q 1 is answered ‘No’**, then no ethics approval is needed.
* **Low Risk:** If **Qs 2-4 are answered ‘Yes’ and** **Qs 5-8 are answered ‘No’**, then no approval is needed from the *Faculty Research Ethics Committee* (FREC). However, your supervisor must approve (a) your information and consent forms (Qs 2 & 3) and (b) your measures for participant confidentiality and secure data management (Q4).
* **High Risk:** If **any of Qs 5-8 are answered ‘Yes’**, then you must submit an application for full ethics approval *before* the project can start.This can take up to 6 weeks. Consult your supervisor about how to apply for full ethics approval.

**Risk Assessment:** Separate guidance on risk assessment can be found on UWE’s Health and Safety forms webpage at <https://go.uwe.ac.uk/RiskAssessment>. If needed, you must complete a Risk Assessment form. This must also be attached to your application for full ethics approval if your project is **High Risk**.

|  |
| --- |
| **Your supervisor must check your responses above *before* you submit this form.** |
| **Submit this completed form via the *Assignments* area in Blackboard (or elsewhere if so directed by the module leader or your supervisor)***.* |
| After you have uploaded this form, your supervisor will confirm it has been correctly completed by “marking” it as *Passed*/100% via the *My Grades* link on the Blackboard*.* |

Further research ethics guidance is available at <http://www1.uwe.ac.uk/research/researchethics>