

Literature review

Lighting and rendering in THE unity GAME ENGINE

Awais Tasleem | Final Year Project | 11/26/2022

Table of Contents

[Introduction 5](#_Toc121063373)

[Lighting 5](#_Toc121063374)

[Rendering Pipeline in Unity 6](#_Toc121063375)

[Application 6](#_Toc121063376)

[Geometry 6](#_Toc121063377)

[Rasterization 6](#_Toc121063378)

[Rendering process 7](#_Toc121063379)

[HDRP vs URP 7](#_Toc121063380)

[DRM and Security Issues 8](#_Toc121063381)

[Methodology 9](#_Toc121063382)

[What is a methodolgy 9](#_Toc121063383)

[Agile 9](#_Toc121063384)

[Scrum 9](#_Toc121063385)

[Programming Tools 10](#_Toc121063386)

[Unity Game Engine 10](#_Toc121063387)

[C# 10](#_Toc121063388)

[Visual Studio IDE 10](#_Toc121063389)

[Visual Scripting 11](#_Toc121063390)

[Summary 11](#_Toc121063391)

[References 12](#_Toc121063392)

[Appendix a: HDRP vs. URP Feature Comparison Overview 0](#_Toc121063393)

[Appendix b: Unity HDRP Lighting tutorial project 1](#_Toc121063394)

[Appendix c: Process of rendering 3](#_Toc121063395)

[Appendix d: Visual Studio Languages 3](#_Toc121063396)

# Introduction

This literature review discusses how lighting is used in the 3d game engine, and how important lighting is in any game and what lighting can show the gamer either physically in the game world, or emotionally in the gamer’s mind. It will also discuss how lighting in game development has changed and improved over the years and how it has changed the industry, for example ray tracing has made a huge impact on games in the past few years where even the new generation of consoles (PlayStation 5 and Xbox series consoles X | S) and phones with the snapdragon 8 gen 2 SOC (System on a chip) now focuses on ray tracing and lighting after Nvidia released the RTX series of graphics cards back in 2018. I will also talk about the Unity game engine and what role this engine has played in the game development industry.

# Lighting

In Unity, Lighting is approximated to how light works in real life. Unity allows the use of models that have high detail to get a more realistic look, or a more simplified model to get a more stylized look. (Unity Technologies, 2022)

Unity uses several different methods for lighting and one of these methods is Direct lighting. Direct lighting is when the light has bounced off a surface once and is reflected to a sensor such as the game camera. Indirect light is all the other types of light that is reflected to the camera such as skylights, multiple bounce lights. Most realistic games use a combination of them both to make a better-looking game.

Another lighting method that Unity uses is baked lighting. Baked lighting is technique of performing lighting calculation before runtime, and then the results are stored as lighting data, then when the game is run, the engine will get the lighting data and apply it at runtime. This saves in lots of performance as the engine only needs to do a lookup operation instead of a full calculation.

Another lighting method that Unity uses is real-time lighting. This is the opposite of baked lighting where in baked lighting the lighting data is calculated beforehand, but for real-time lighting, the data is calculated at runtime, this could be every frame in the game by using the update method in Unity. This way of doing lighting is very taxing on the graphics card as the calculation needs to be run on every frame. Nvidia has a solution to do this by using RT cores in their RTX line of graphics card, where all the lighting calculations are done on those cores, which increases performance.

Another method that Unity uses for lighting is called Global Illumination. Global illumination is a group of techniques from the direct and indirect techniques to make a more realistic looking game. Unity has 2 built-in global illumination systems, which combine direct and indirect lighting. (Unity Technologies, 2022)

One of the global illumination systems is called the baked global illumination system and it uses light mappers which calculate the brightness of the surface before runtime, light probes which provide a way to see data of light moving through an empty space in the game scene and reflection probes which ‘allow the visual environment to be sampled at strategic points in the in the scene’. (Unity Technologies, 2022)

Unity has a build in project tutorial that shows how lighting works in a HDRP project, and I have followed this tutorial and worked on a sample scene. Appendix b:

# Rendering Pipeline in Unity

Rendering is the process of drawing a game scene which could be full of models and lighting on a computer screen. It involves a combination of many different methods, such as geometry calculations, textures, and lighting. (Schrute, 2019) The rendering pipeline transforms all the game data such as models, lighting, and textures into a virtual environment on the player’s screen to show the player the game world. A rendering pipeline consists of 3 layers, Application layer, Geometry Layer and Rasterization layer.

## Application

The application layer is where the developers command the calculations for the geometry and the rasterization. This is the top of the stack. The application usually talks to a graphics API such as DirectX 12 which then talks to the physical hardware on the device.

## Geometry

This layer consists of all the shapes and vertices that make the models in a game. This layer also calculates the position of the camera in the game world. The engine then calculates all the data and then passes them onto the rasterization process.

## Rasterization

This is the process of converting 3d modeled objects and the environment data such as lighting and textures to a 2d plane. This is since our devices screens are flat, and the data needs to be converted to fit this view or the game would not look correct to the player. The conversion sends the 3d world data through filters that then in turn show us a rasterized 3d image on a 2d screen.

## Rendering process

The rendering processes consists of Geometry, Illumination, View Perspective, Clipping, Screen Space Projection, Adding Post Processing and Display, see Appendix c:

# HDRP vs URP

Before starting up a new unity project I had to decide which render pipeline I had to use. A render pipeline preforms a set of operations that take the contents of a scene and display them on the screen. (Unity Technologies, 2022) Unity has allowed developers to create their own render pipeline, but they have also provided 2 render pipelines for us to use. One is the universal render pipeline (URP) and the other is the high-definition pipeline (HDRP). A HDRP project is not compatible with a URP project. (Unity Technologies, 2022)

URP is a prebuilt render pipeline made by unity to allow developers an easier and user-friendly way to build their games that will work on a variety of devices such as mobile phones to high end desktop PC’s. (Unity Technologies, 2022)

HDRP is a high-fidelity scriptable render pipeline that was made by Unity to target higher end hardware such the new Nvidia RTX 4090 graphics cards. (Unity Technologies, 2022) HDRP utilizes physically based lighting techniques, linear lighting, HDR lighting and a configurable hybrid tile/cluster deferred/forward lighting architecture. (Unity Technologies, 2022)

For this project I have chosen to use Unity’s high definition render pipeline, as it allows me to have more realistic lighting, it also allows for real time global illumination which is a group of techniques that model both direct and indirect lighting to provide a realistic lighting effect on the world. (Unity Technologies, 2022)

Another feature of this pipeline that I may use as it is one of my optional objectives, is that it allows me to use raytracing in the game to give a more real world look and feel for the lighting. Raytracing is the process of generating an image by tracing out rays from the camera through each pixel and recording the color contribution at the hit point. (Unity Technologies, 2022). The main issue with ray tracing is that it requires a lot of graphical horsepower to run, and my current system may not handle it well, but I may be upgrading in the near future, and this is something I may be able to focus on which is the reason why it is an optional objective.

# DRM and Security Issues

DRM stands for digital rights management, and it is used in game development to reduce piracy of games. If you are buying a game on a digital platform like Steam on PC or PlayStation store on PS5/PS4, you aren’t buying the game, you are buying a license to run the game on the specific hardware that the license allows. If you buy a game on the PS5, you won’t be able to play that game on an Xbox Series X console.

The DRM software first checks for a license file when the game first launches. If the DRM software finds a match, then the game will run normally, and the DRM software will stop running. If the DRM software doesn’t detect the correct license file, then the game will not run, and an error message will be shown to the player.

Some new DRM software requires a constant internet connection all the time and the DRM software runs all the time when the game is launched. This was used on many games such as sim city 2013 and Diablo 3. (Roach, 2020) This can cause performance issues when a game is run, but developers are reducing the use of this method.

Another security measure that games us called the Anit Cheat system. Cheating has become a big threat to digital online games as a cheat could render a game unplayable by players and then the game is abandoned, which would make the development company lose lots of game revenue.

One way a cheater would exploit a game to get a competitive advantage was to modify game drivers to allow them to see through walls in game or have an aim bot that would give the player perfect aim. These would make other players angry, and they would rage to quit the game. Some of these wouldn’t return to the game or they would only play the game with their close friends. Some would completely leave the game and never join again; this would make the games development companies lose lots of revenue. (Ellis, 2020)

A controversial way that companies are trying to stop cheaters in games is releasing anti cheats that access the kernel at ring 0 level. One game that does this is Valorant that uses ring level 0 for its cheating software. This is a huge on pc games as this level is the most central part of the OS, if anything fails here, it will probably cause an entire system shutdown. (Raspberry Pi, n.d.)

# Methodology

## What is a methodolgy

A methodology is a structured process that is used when working on a project. The main reason for using a methodology is to provide a systematic approach to the software development process.

A definition of a software methodology is ‘a framework that is used to structure, plan, and control the process of developing an information system.’ (Saylor.org, 2021)

## Agile

I have chosen to use the agile methodology during the development of this project. The reason I have chosen the agile methodology is that it allows me to create a feature and then quickly respond to the change. In game development, requirements are always changing, and deadlines are also changing, so that is why agile is used a lot in the game development industry. Agile is for quick iterative software changes on shorter sprints.

All my objectives that are stated in my project proposal are compatible with the agile framework, as each objective can be broken down into smaller parts and then all the smaller parts can be joined together to make the entire project complete. One example of this is that objective 6 is where I need to complete a minimal viable product, and objectives 7 and 8 are improving on objective 6. This shows that adding small quick features makes it easier to work on developing the game instead of working on the project as a whole and then testing at the end.

One major advantage of Agile development is that it allows more collaboration between a team and allows them to work faster together. It allows the teams to work on smaller projects instead of working on the entire project at once.

## Scrum

Scrum is an agile methodology, and it is designed for quick short sprints, usually 1 week, where you work on a feature and evaluate it then get feedback on it withing a week. This is one of the methodologies that I used when I was on placement last year at Knorr-Bremse.

Scrum allows the developer to quickly add features, and then also test and fix errors quickly. In game development, this is very useful as one feature could break many other features, and quick testing and fixing allows the game to be more smoothly developed and reduced the number of bugs that are in the game.

# Programming Tools

## Unity Game Engine

I will be using the Unity game engine to develop my game. Unity is a professional game engine that has been used to make many AAA quality games such as Fall Guys: Ultimate knockout, Among Us (2018), Cuphead and Pokémon Go. Unity is also used to make many mobile games. About 50% of all mobile games are made using Unity. (Unity Technologies, 2022)

Unity Game Engine has been around since June 2005, and it was a paid software. Then in 2009, the Unity game engine became free to use by anyone, this was announced at the Unite 2009 Conference. (Helgason, 2009)

## C#

Unity game engine allows scripting in the C# language which was developed by a Microsoft employee called Anders Hejlsberg. (Pluralsight, 2022) C# is an object-oriented language and makes developing games easier as you can create C# objects that are linked to game objects. For example, you could have a player model and the class linked to it would be a player class, and the class would have sub classes such as player health, player armor and player inventory. These classes would be linked to the actual 3d model in the game.

Unity also allows other .NET languages if they can compile a compatible DLL file, this means C++ could be used for faster and efficient running code in the game as C++ is a lower level, more down to metal language and can run code much faster than C#. (Unity Technologies, 2021)

## Visual Studio IDE

I will be using Microsoft IDE called Visual Studio 2022 to code in C#, and these scripts automatically get added to Unity as Unity has built-in support for opening external scripts.

Visual Studio was released in 1997 by Microsoft, and it is an IDE used industry wide by many professionals and companies. When I was on placement last year, I used Visual Studio to do my work. Visual studio is a multi-language IDE where it has support for more than one programming language. You can code in C#, C++/C, JavaScript, python, F# and visual basic. Appendix d:

## Visual Scripting

Unity game engine also allows visual scripting like the game engine called Scratch which allows the user to drag and drop blocks of instructions to code the game object. Visual scripting allows easier collaboration between programmers, artists and designers and it allows for faster prototyping and iteration. (Unity Technologies, 2021)

Unity’s visual scripting contains logic-nodes attached to each other to do an action, this could be a walking animation where a key press makes the player go forward and an animation if this is shown.

# Summary

This project will be developed using the Unity Game Engine. I will be learning to code using Unity’s documentation and online tutorials. I will also be using the Visual Studio IDE to code in. The 3D models will be made using the Blender modeling software, and the textures will be created using photoshop. The Project will be managed by using Trello. I will be targeting my Unity builds to be run on windows.

The main skill that I need to focus on is my time management skill as this project is worth 40 credits and I need to spend an appropriate amount of time on it. It will also allow me to stick to my methodology.

# References

Brackeys. (2020, March 1). HDRP vs. URP - Which Unity Template should you choose? *HDRP vs. URP - Which Unity Template should you choose?* Denmark: Youtube.

Ellis, N. (2020). *Security and Online Gaming: Understanding Anti-Cheat*. Retrieved from schellman: https://hub.schellman.com/blog/security-and-online-gaming-understanding-anti-cheat

Helgason, D. (2009, October 28). *A free Unity?* Retrieved from Unity: https://blog.unity.com/technology/a-free-unity

Pluralsight. (2022, November 14). *What is C# prgramming? A beginers guide*. Retrieved from Pluralsight: https://www.pluralsight.com/blog/software-development/everything-you-need-to-know-about-c-#:~:text=When%20compared%20to%20long%2Dstanding,a%20history%20for%20popular%20creations.

Raspberry Pi. (n.d.). *FutureLearn*. Retrieved from Layers of an Operating System: https://www.futurelearn.com/info/courses/computer-systems/0/steps/53514#:~:text=Ring%200%20(most%20privileged)%20and,be%20running%20in%20kernel%20mode.

Roach, J. (2020, October 13). *What is DRM in video games and how does it work?* Retrieved from Digital Trends: https://www.digitaltrends.com/gaming/what-is-drm-in-video-games/

Saylor.org. (2021, June 22). *Software Engineering*. Retrieved from Saylor.org Academy: https://learn.saylor.org/course/view.php?id=73

Schrute, A. (2019, November 7). *Rendering Pipeline In Unity*. Retrieved from Medium: https://medium.com/shader-coding-in-unity-from-a-to-z/rendering-pipe-line-f0471aa0904b

Unity Technologies. (2021, February 24). *Managed Plug-ins*. Retrieved from Unity docs: https://docs.unity3d.com/2020.1/Documentation/Manual/UsingDLL.html

Unity Technologies. (2021). *Unity Visual Scripting*. Retrieved from Unity: https://unity.com/features/unity-visual-scripting#ready-get-started--2

Unity Technologies. (2022, November 25). *Glossary*. Retrieved from Unity Docs: https://docs.unity3d.com/Manual/Glossary.html#globalillumination

Unity Technologies. (2022, November 18). *High Definition Render Pipeline overview*. Retrieved from Unity Docs: https://docs.unity3d.com/Packages/com.unity.render-pipelines.high-definition@14.0/manual/index.html

Unity Technologies. (2022, November 25). *Introduction to Lighting*. Retrieved from Unity Docs: https://docs.unity3d.com/Manual/LightingInUnity.html

Unity Technologies. (2022). *Make a 2D game in Unity*. Retrieved from Unity: https://unity.com/how-to/beginner-2D-game-resources

Unity Technologies. (2022, November 25). *Reflection Probes*. Retrieved from Unity Docs: https://docs.unity3d.com/Manual/ReflectionProbes.html

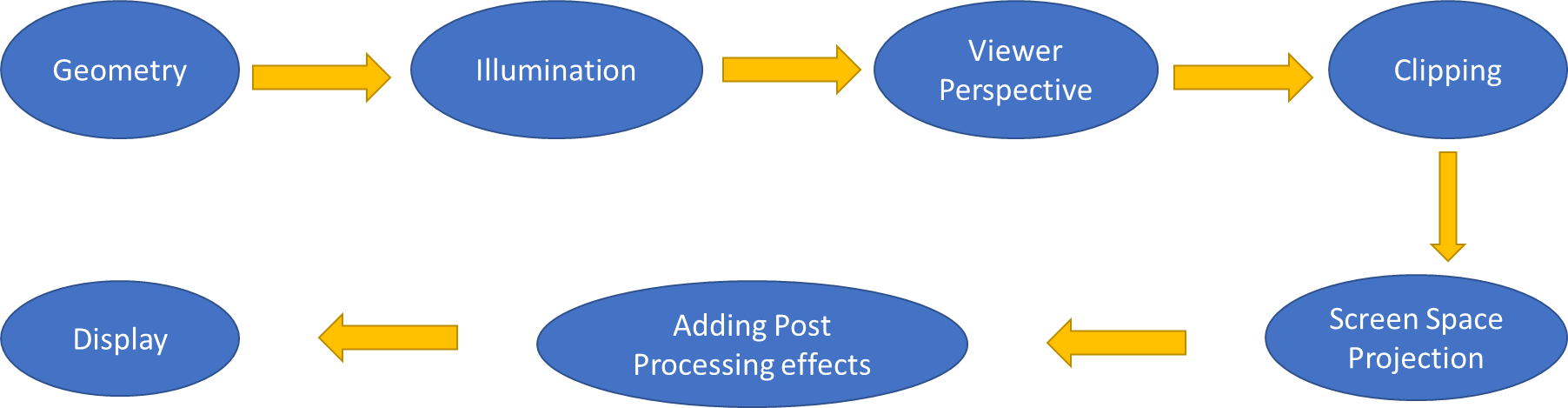
Unity Technologies. (2022, November 18). *Render Pipelines*. Retrieved from Unity Docs: https://docs.unity3d.com/Manual/render-pipelines.html

Unity Technologies. (2022, November 18). *Universal render pipeline overview*. Retrieved from Unity Docs: https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@12.1/manual/index.html

|  |  |  |
| --- | --- | --- |
| Feature | HDRP | URP |
| Shader Graph | Yes | Yes |
| Visual Effect Graph | Yes | Yes |
| VR | “Yes” | Yes |
| Decals | Yes | In research |
| Deferred Rendering | Yes | In research |
| **Lighting** |  |  |
| Realtime Global Illumination | Yes | In Research |
| Raytracing | Preview | LOL |
| Volumetric Lighting | Yes | No |
| Line and Area lights | Yes | No |
| **Shaders** |  |  |
| Subsurface Scattering | Yes | No |
| Stack Lit Shader | Yes | No |
| Detail Maps | Yes | No |
| Heightmap | Yes | No |
| Parallax Mapping | Yes | No |
| Light Cookies | Yes | In Research |
| **Post Processing** |  |  |
| Ambient Occlusion | Yes | In Research |
| Auto Exposure | Yes | No |
| Screen Space reflections | Yes | No |
| **Camera** |  |  |
| Camera Stacking | Yes | Yes |
| Physical Camera | Yes | Yes |
| **Visual Effects** |  |  |
| Halo | Yes | No |
| Lens Flare | Yes | No |

1. HDRP vs. URP Feature Comparison Overview  
   Brackeys a YouTube creator that makes videos about the Unity game engine has made a video comparing the features of HDRP and URP (Brackeys, 2020). He’s also released a table that shows all features available from Unity and which pipeline can use which feature.
2. Unity HDRP Lighting tutorial project Graphical user interface, text

   Description automatically generated Graphical user interface

   Description automatically generated 
3. Process of rendering
4. Visual Studio Languages