

# CS324 - Report

January - 2022

Word count: 985

## Introduction

We were tasked to write a graphics program using WebGL and GLSL to create a video game. Our game is a recreation of a children's game shown in Squid Game [1]. This game is known as Red Light, Green Light, where you must reach to the end of a field without being detected.

## How to run

In order to run the game we require you to run a local server, this can be done via HTTP servers. For example, in Python:

```
//Python 2.x  
python -m SimpleHTTPServer
```

```
//Python 3.x (recommended)  
python -m http.server
```

This will server files from the current directory, at localhost under port 8000. Therefore, just type *http://localhost:8000/* in the address bar and click on **main.html** to run the game.

## Interactivity

For the player to be able to interact with the game using a mouse and keyboard, we utilised a Three.js function known as PointerLockControls [2] which is based on the Pointer Lock API, it provides input methods based on the movement of the mouse over time and allows us to capture mouse events.

For keyboard controls, we used event listeners, so we listened to when the keys 'WASD' were used and set the corresponding movement for them, furthermore, having an event listener for the key 'C' to switch camera.

## Camera

For our camera we used a perspective camera [3] as it is designed to imitate how a human eye sees. To switch between first and third person cameras, we have our player model invisible copying all our movement while in first person, then after the player switches, the camera is moved up and the player model is moved forward and becomes visible, this is to give the illusion of a third person camera, so as the user moves, the player model will move ahead mimicking the user's move.

## Menu

To create our menu, we used HTML to structure the page and then CSS to style it. The menu consists of a logo, instructions on how to play the game and three buttons, the first two buttons will take you straight to the game, level 1 and 2 respectively, where as the final button will take you to the acknowledgements. For styling, we took inspiration from Squid Game, the logo is based from the actual series' logo and the

font is also Squid Game style. The colour palette for Squid Game is black, pink and green, so the colours of the background and text represent that palette.

## Levels

There are two levels which the player can pick from at the menu, however, if they choose the first level and successfully complete it they will be moved to the next level. Both levels are very similar in terms of gameplay but differ in the scenery and the mechanics, for example, in level 2 the player is much slower. In order to achieve having two levels, we created two scene objects and have one main scene always running, depending on what the player chose, the main scene will either be level 1 or level 2.

## Models

For our model, we used a software called Blender [4] to create our 3D Model of a lamp post.

Firstly, we created a cylinder and flattened it to create a base.

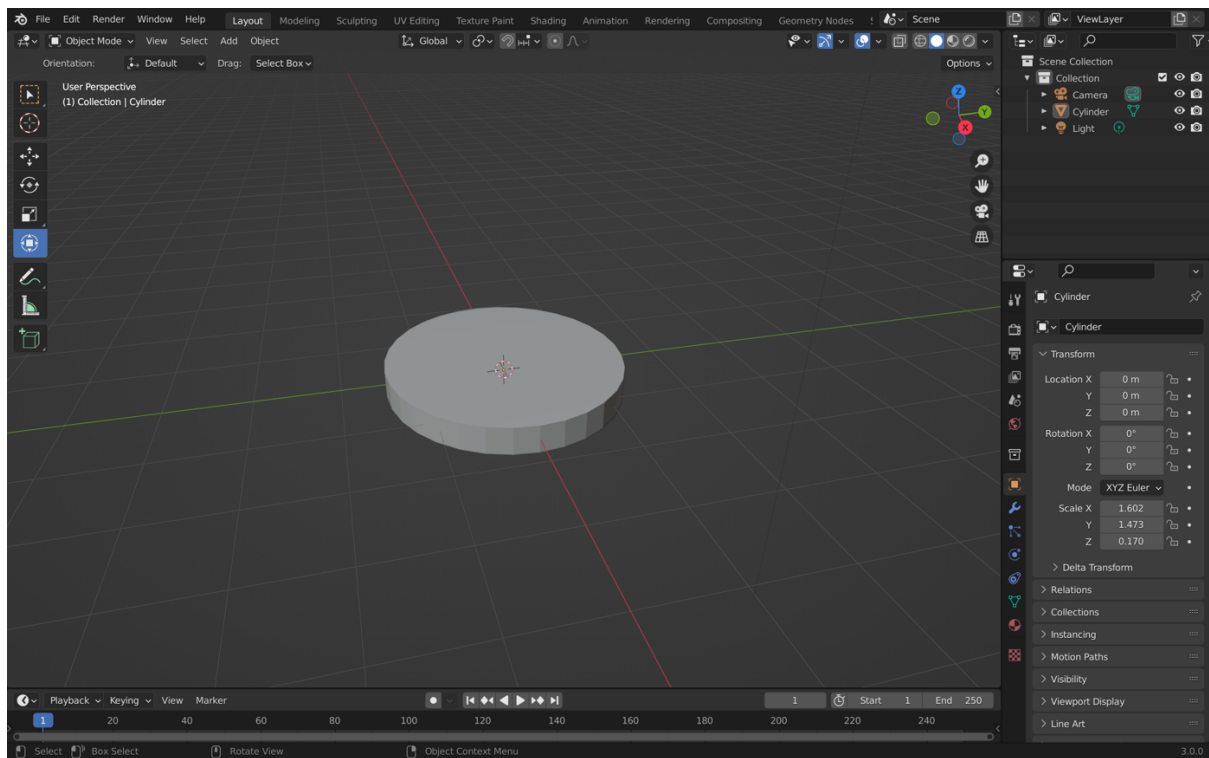


Figure 1: Base

After, we added two more cylinders, one with a slightly smaller radius than the previous cylinder and stretched it higher, the other cylinder had an even smaller cylinder and went above the previous one, we utilised the bevel tool to create chamfered edges between these two cylinders.

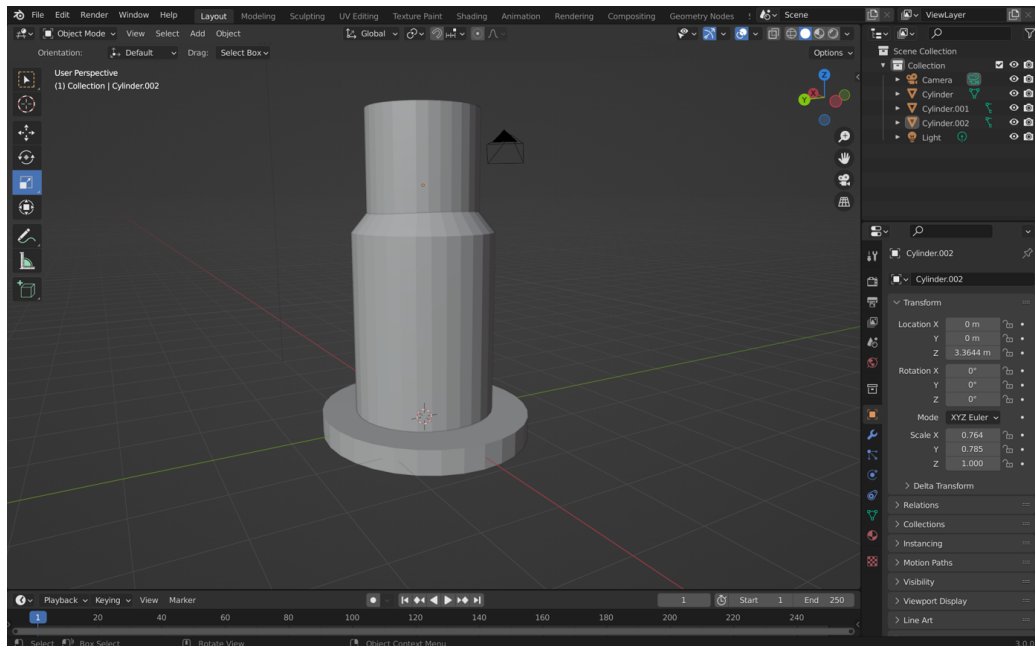


Figure 2: Base 2

We continued further and created another cylinder which will be the top half of our lamp where it will bend around, this cylinder also has a smaller radius and will be much taller, we used the bevel tool once more to create the chamfered edges. Also at this point we created edges and vertices on each face of the cylinder so it will be made out of only quads and triangles.

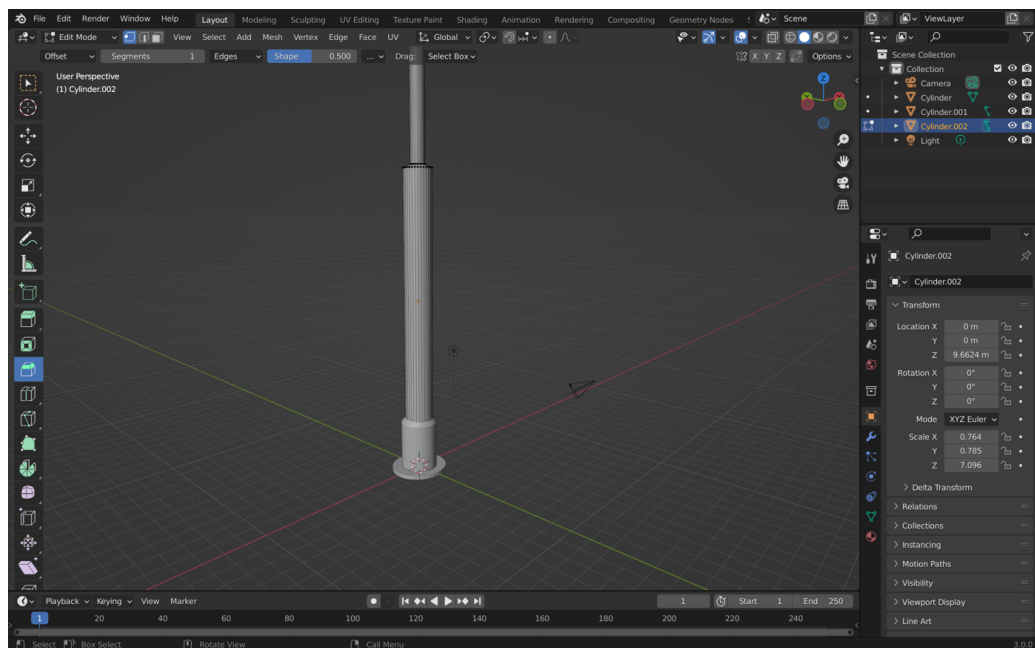


Figure 3: Base 3

To create a bending cylinder, we used the spin tool, we selected the vertices on top of the cylinder and extrude them in a circle around the origin with the spin tool, we also moved the origin to the direction where we want to bend.

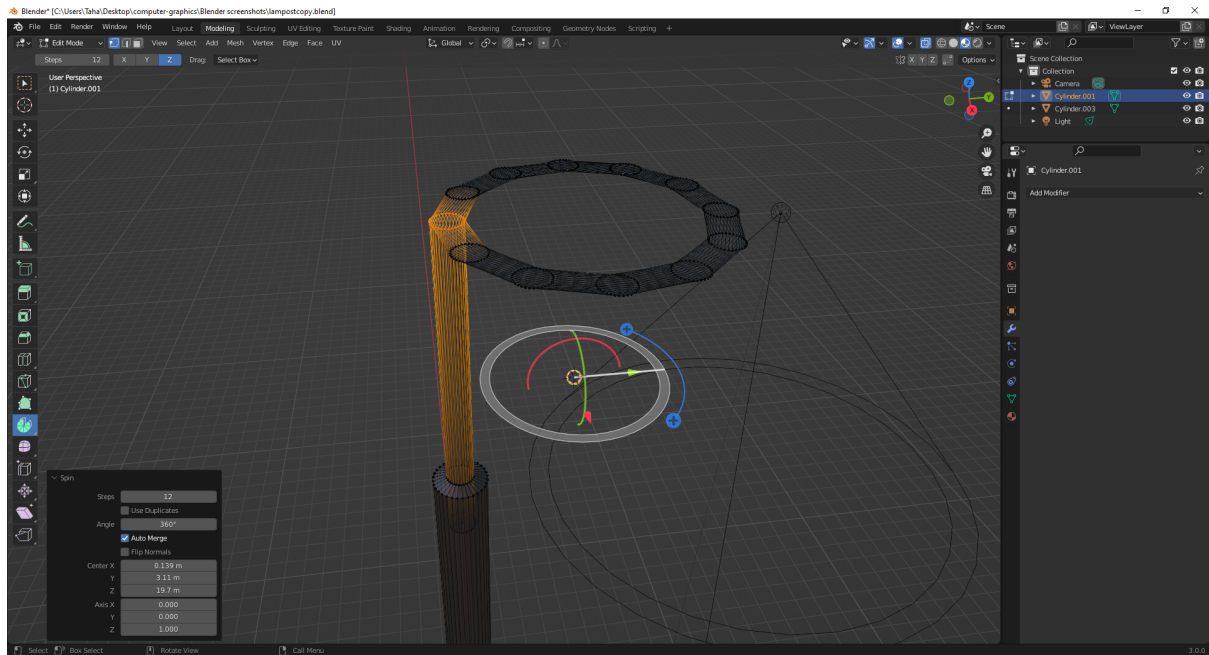


Figure 4: Bend 1

Now we set properties using the spin tool, we set the x-axis value to 1.000 and the z-axis to 0.000. We then set the angle to 180 degrees so we can model a lamp post. Finally, Using the center of spin we can define the amount and radius.

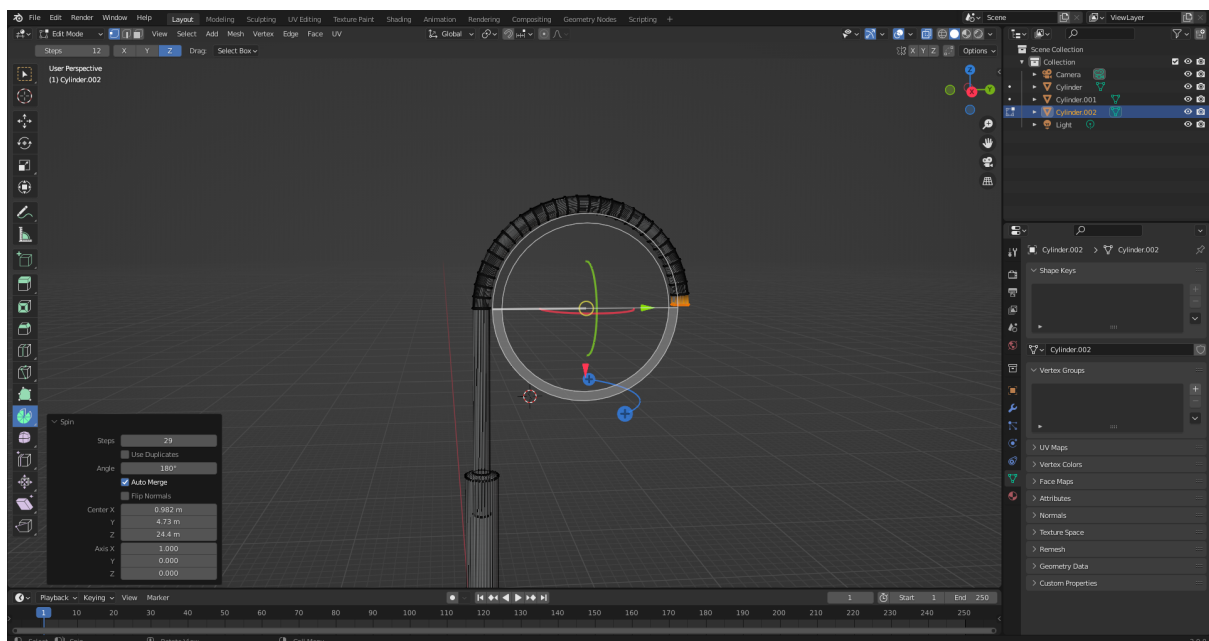


Figure 5: Bend 2

Finally, to create the lamp shade, we selected all the vertices at the end of the bend, extruded and expanded the radius to create a cone-like shape. We then used a solidify modifier to give it thickness.

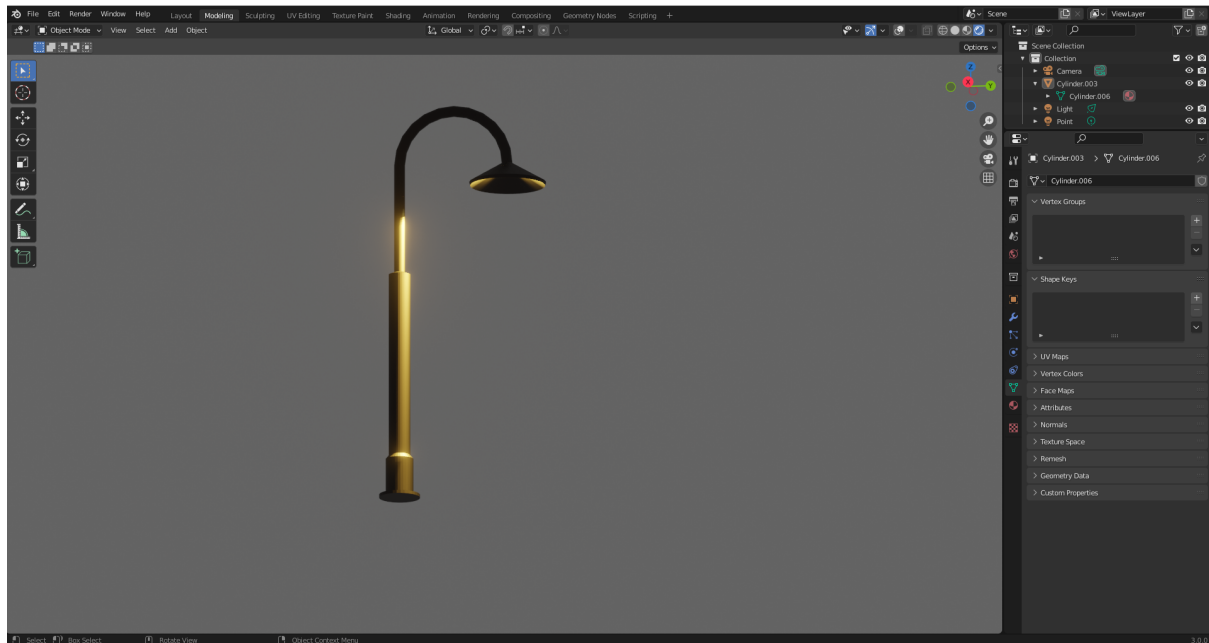


Figure 6: Finished lamp

## Lighting

We introduced two light sources, an ambient light that globally illuminates all objects in the scene equally. The second light source were two point lights, which is a light that gets emitted from a single point in all directions, we used this to represent the light coming from our lamp posts. For each lamp post, their light covers just over half the arena, so it can create double shadows in some cases, we used two different colours for level 1 and 2, one that represents dawn and the other that represents dusk.

## Textures

We introduced a couple of textures for our game, one was for the wall which represents the arena from the actual Squid Game show, another was a beach texture for the floor.

## Desirables

We introduced some sound effects in our game, one was a remix of the Squid Game theme which we used as background music throughout the game. Another sound effect was a death effect from a game known as 'Roblox' which is used when the player is caught by the doll.

Lastly, we introduced some animations for the doll, we utilised the GSAP library [5] which allows us to build some animations. Our animations were only used for the doll turning around and back between phases.

## References

- [1] Inc. © 1990-2022 by IMDb.com. *Squid Game*. URL: <https://www.imdb.com/title/tt10919420/>.
- [2] mrdoob. *PointerLockControls*. URL: <https://threejs.org/docs/#examples/en/controls/PointerLockControls>.
- [3] mrdoob. *PerspectiveCamera*. URL: <https://threejs.org/docs/#api/en/cameras/PerspectiveCamera>.
- [4] Blender. *Blender*. URL: <https://www.blender.org>.
- [5] Inc. © 2020 GreenSock. *Professional-grade JavaScript animation for the modern web*. URL: <https://greensock.com/gsap/>.