COSC342 Assignment 2 Report: Ray Tracing

William Wallace
1216661
18/05/2020

Approach

For this assignment, I started with the basics such as implementing the ray-plane interactions, and then progressed onto more complicated shapes such as cubes and cylinders. At first, I relied heavily on using my lab work to guide how I set up my scenes and how to calculate the hit points, though I was able to successful implement the required features in my test scene, as illustrated in Figure 1.

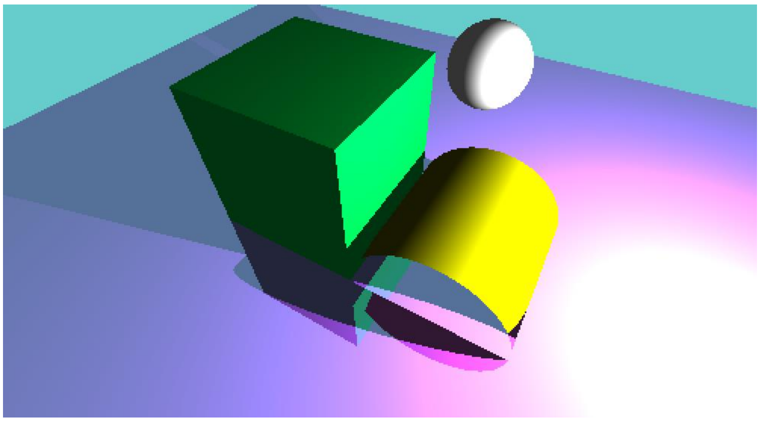

Figure 1. Sample scene, consisting of different objects and lighting features.

Implementing objects

Each of my objects' intersections relied on applying the inverse transform to the ray, rather than transforming the object itself. This is beneficial because the ray tracing algorithm that is used in this assignment is greatly simplified when it assumes the shape that we are intersecting with is at the origin, unrotated and with unit size. This is reflected in the line below, which is present at the beginning of each shape's implementation:

```
Ray inverseRay = transform.applyInverse(ray);
```

Next I computed the intersections for each object. For the requirements, the formula to do this is equal to 0 on the surface, $f(x, y, z) = 0$. I then got the intersections by solving $f(x_0 + td_x, y_0 + td_y, z_0 + td_z) = 0$ for t, where (x_0, y_0, z_0) is the starting point of the ray and (d_x, d_y, d_z) is its direction. The algorithm would then find whether the ray intersected with, glanced, or missed the sphere based on the number and value of the solution(s). Finally, I applied the forward transform to the hit points.

Using what I knew about how to implement a plane, I made a cube by joining 6 planes together as one primitive, so that I could move it around as a whole. As the lab suggested, I began by making the faces of the cube parallel to the *X*-*Y* plane, and I added faces after verifying that I had implemented each pair successfully in the display window.

I found the ray-cylinder interaction to be the most time-consuming and frustrating object to implement in Assignment 2. The curved edge was possible, however the ends of the cylinder – what I had intended to be planes – ended up not appearing. This was because it had the normals of a sphere so I needed to remove the *Z* component in my normal calculations to change them to cylinder normals. Despite my best efforts, I could not fix this in time for submission.

Implementing lights

Diffuse and specular lighting were made possible by following the steps in the lab book. As a result, I found it more difficult to implement the light sources rather than the lighting models. I found implementation of the directional light source to be quite straightforward, since it is only defined by its direction only, since its distance is infinite by definition and the inverse squared law does not apply. Consequently, I computed the object's colour to be flat, and I returned the colour. The spotlight's implementation was similar to a point light source, but a key insight was that the direction of the light is such that all of the light outside of the spotlight's cone was at an intensity of zero, and that dot products can be used to calculate angles between vectors.

Testing objects

My primary method of testing my objects was by adding the object and a point light, modifying the transforms in my scene file, `sampleScene.txt`, and viewing it. Modifications included rotating my objects around on all their axes by changing the `camera` and `rotate` variables, and applying different light sources to them to see how the objects would react. In doing so, I would check in the viewport window to see if my objects had the expected shape, colours, shadows, reflections and specular highlights. Despite this, sometimes the cube would appear more as a square or a plane - either because the axes were aligned to the camera, the light and colour choice did not show enough depth, or the camera was too close to the cube. Trial and error gave me the position `X 3 Y 2.5 Z 0.5` for the cube, which put the cube at a good angle for my testing purposes. I kept the ambient light at a moderate level while these tests were being conducted (for example RGB = `0.4 0.4 0.4`), however I found that adjusting the ambient light up and down would sometimes help me to see the effects of my changes more clearly. Often, I found that my object's normals were difficult to test with just my ambient lighting, I made sure to test them with diffuse and specular lighting.

Testing the lighting

Overall, I found the difficulty of testing the lighting to be mixed, where the light sources were difficult to test, and the light models to be easier.

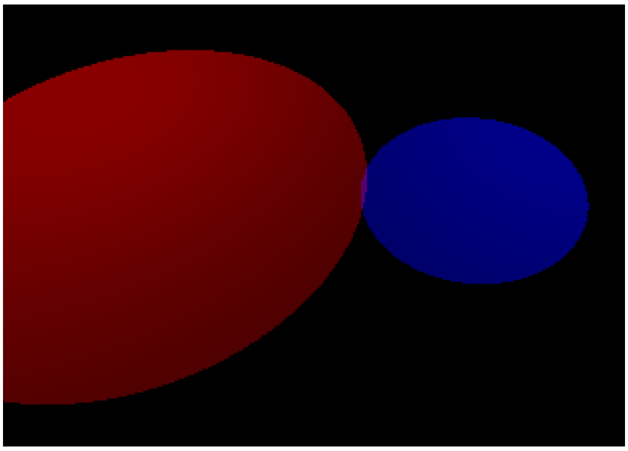

Figure 2. Spotlight test scene, in a Venn diagram arrangement.

Testing the diffuse and specular lighting was easiest when I had imported a sphere from one of the previous labs to be somewhere in the camera's view so that I could see the effects of diffuse and specular lighting. I was able to view these lighting effects more clearly by comparing my viewport to the viewport in lecture slides. Moreover, spheres uniquely apt as a reference because they allowed me to have a standardised idea of the shading effects, since it appears as a circle from any angle.

I was able to see if the directional lighting was working by giving it a colour and direction in the sample scene and seeing if the shadows of and object aligned with the outline of its shape and if the colour of the object was uniform. I had trouble testing directional lights because I made an error in my light calculation code as mentioned in the shadows section. After this was fixed, it worked as expected.

I tested my spotlight by defining its location, colour, angle and direction in the sample scene, and observing those effects on a sphere centred on a plane. This worked in my test scenes where I used a Venn diagram type arrangement (*Fig. 2*) to show off their own colours, however, after much testing I could not find where my spotlight was shining in my final scene, so it doesn't appear in it.

Sample Scene

My sample scene shows a plane, cube, sphere, and cylinder; each with their own properties of shading, reflection and shadows. The cut-in-half cylinder shows adds more interest to the scene with its reflections and shadows, while the directional lighting adds another element to the scene, too.

Evaluation

Overall, I am happy that I was able to implement most of the tasks required for this ray tracing assignment, however I have two reservations about my final submission.

Firstly, my testing process involved trying to find objects in my scene by adjusting the variables' values and checking the viewport to see what had changed, which seemed slow. I am unsure if there was a better way to do this, and perhaps I would have worked faster if I hadn't spent so much time guessing and checking values to see the effect on my scenes.

Secondly, my octahedral was not implemented. My approach started by joining four triangular planes together at a square base and then mirroring that on the other side of the square, however I was unfortunately not able to construct it. If I had a clearer idea of how to go about this, perhaps by asking a lab tutor, I may be able to implement the octahedral shape in the future.