**Programming Project Report**

Student Blake Williams

Student 010974718

**Problem Statement:**

The goal of this programming assignment was to simulate 2 rotating spheres, and apply another light source onto the rendering screen. The inputs were to allow the user to change the camera angle.

**Design:**

The first design was to add 2 spheres and make one stationary and the other rotate. The design for this would be to implement a timer() function that calculates and updates the position of the rotating sphere based off the cos() and sin() of its last position. Then the second design choice was to implement another light onto the render screen. To do this a modification to the raytrace() function was required. By making there not be a single light source, but two rendered.

**Implementation:**

To implement the first design (2 spheres) the init function was modified. Inside it declares a stationary sphere[1] that is set to 0,0,0 and is scaled down. Then a second sphere is set to be outside the inner sphere by a factor of 1.5. This prevents, during rotation, the two spheres intersecting with each other. One the two spheres are rendering on the screen, next was implementing the movement of the second sphere around the first. To do this a timer() function was created which modifies the position of the second sphere. By taking the last position of the spheres x,z coordinates, it calculates its next position using sin() and cos() and the old coordinates. Then the timer calls for a redisplay of the windows after running the ray\_trace function again. With this the spheres rotate around each other. Next was adding another light source, to do this the ray\_trace() function was to be modified. Specifically in the provided phong shading mode. Here is where the initial light is declared. To make it so multiple lights are being rendered a for loop was created to run through the number of lights (2). Outside of the for loop the initial light\_color and light\_dir variables are redeclared to light\_color[lightcount] and light\_dir[lightcount] this enables the use of the for loop to assign individual values to the two light sources as light\_color[0] and light\_color[1] represent two different light sources. Then inside the for loop the in\_shadow()call was changed the variable light\_dir to light\_dir[i]. Next the line shader.SetLight(light\_color,light\_dir) was changed to shader.SetLight(light\_color[i],light\_dir[i]) to set the positions of each individual lights which are declared earlier in the ray\_trace() function with there own individual values. Finally the colors rendered and based of a variable called total, which allows for both lights to be displayed. If it was pixel.R,pixel.G,pixel.B then the original light would get overwritten by the second light. This is a basic implementation on how the code was created.

**Testing:**

Testing this code followed a specific pattern. First, add a new feature (another ball). Second, test to see if the new feature interacts with the other features already coded. If they do then the feature is approved, else rewrite the feature to work with the others. Example of this was rendering the rotating spheres. First the Red sphere was rendered to see if the ball was properly ray traced. Because it was properly ray traced then the second ball could be rendered in the same fashion. Then when the ball began to rotate, see if there was a shadow being cast from the sphere onto the other sphere. If there was continue. Everything worked as expected in the end, and made sense.

A red and blue ovals on a black background

Description automatically generated

Above: Shows the Blue Ball casting shadow on red. Red ball has 2 light sources

A red and blue ovals

Description automatically generated

Above: Another angle of previous image

A red and blue spheres on a black background

Description automatically generated

Above: Shows Red ball casting shadow on Blue Ball

**Conclusions:**

The result of this project was a success. As one sphere is properly rotating around the other while two lights are rendered onto them. While rotating there is a shadow cast by one sphere onto another, meaning the shading and raytracing is working. If I were to continue this project I would add more light sources and spheres to the render. This project took 4hrs to complete, with 30minutes for the report