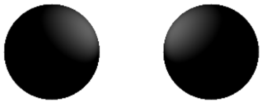
Assignment_1 – pyrokuna (Ryan Gutierrez)

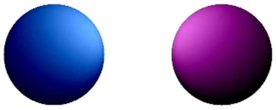
When producing the source code for Assignment 1, I was able to properly implement parts 1.1, 1.2, and 1.3. Note that I made no attempts to implement either of the two extra credit parts. For part 1.4, I was unable to figure out how I was to load the two meshes and so it will not have a dedicated section explaining my work. I extensively attempted to implement part 1.5 and unfortunately failed at an early stage of the process. For part 1.6, I was surprisingly able to implement some form of reflection, although not the correct one. ***I have no special compilation instructions.

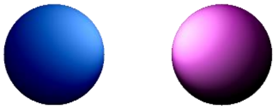
Part 1.1



Part 1.1 was implemented properly—as far as I can tell—through a simple construction of a sphere class to hold my data and a replication of the color calculations and writing (basically the big *if* statement).

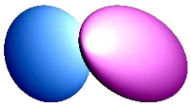
Part 1.2





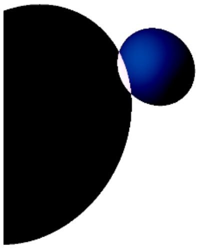
When implementing part 1.2, I first changed the colors by creating three new matrices (shown in the first picture on the left) labelled R, G, and B—respective to their color channels—and modified the diffuse values to change the colors. Then I added specular shading as seen in a comparison between the first and second photo (viewed from left to right). Following this addition, I added a second light source by creating a new light source origin and doing all the same calculations for that light source and adding it as was done for the first (seen in the final picture).

Part 1.3



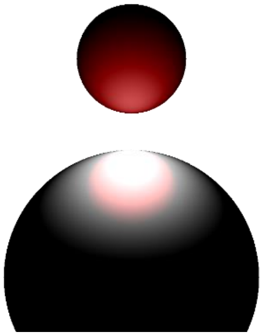
In part 1.3 it seems as though my implementation of perspective projection was a bit off. To achieve this projection, I simply changed the origin and direction of the projected rays. I believe this *warping* effect was due in part to the distance from the "camera".

Part 1.5



In my attempts to achieve a shadow I unfortunately could not progress very far. In the end, my implementation was halted by an inability to properly draw spheres in z space based on their distance (the sphere that was drawn second was the one that would show up in front instead of the one that was closer to the camera).

Part 1.6



When implementing part 1.6, I am nearly certain that my calculation of the reflection ray (r in the course slides) is correct. But, when trying to add it to the objects color (the larger sphere on the bottom in this case), I did not know how to go about using the function raycolor() as I could not find a description in the slides nor the textbook. I instead mimicked the addition of specular lighting and added the reflection in the same way. You'll notice in the source code that this was only done for the R channel (or Matrix) as I simply got lucky that I initially made the upper sphere completely red. When doing the same for the other two channels, the reflection still appeared only it was completely white and not the color of the upper sphere. I still think this implementation was somewhat of a success as even though the color is technically incorrect, the reflection still appeared.