Homework 6 Report

Zack Fravel
010646947

Problem Statement:

The goal of this programming assignment was to take what we learned from the semester and combine it all into a program that can perform ray tracing on spheres. Since we were given full implementations of all other classes, the only work required for us was to call the functions correctly to perform the shading calculations. My program has no inputs, all the inputs are hard coded. The output of the program displays a window with a shaded sphere. Not much error handling was required since most of the functions we're using have full implementations already.

Design:

I decided to try and make as parsimonious of design decisions as I could. I initialize a vector of rays, aptly called tracers, and in my main method I initialize the scene using the standard input parameters for the camera and light we were provided. I then initialize my spheres and one Phong object. I then loop through all the points on the image plane and create a ray from the camera to each point and push_back on the vector of rays. Now I have a vector of all the rays I'll need for color calculations.

Next, I loop through all the rays and check whether or not any of the rays intersect with my spheres. If so, we set the color appropriately to the returned color value. If not, we set the color to black. I decided to add onto the Ray3D class definition by allowing a color to be stored and return to allow for easy access of the colors. The pros of this is it made development very quick and the number of rays is always exactly the number of pixels so they iterate in accordance with one another. The cons are that color and rays don't always need to be together so it might be redundant design.

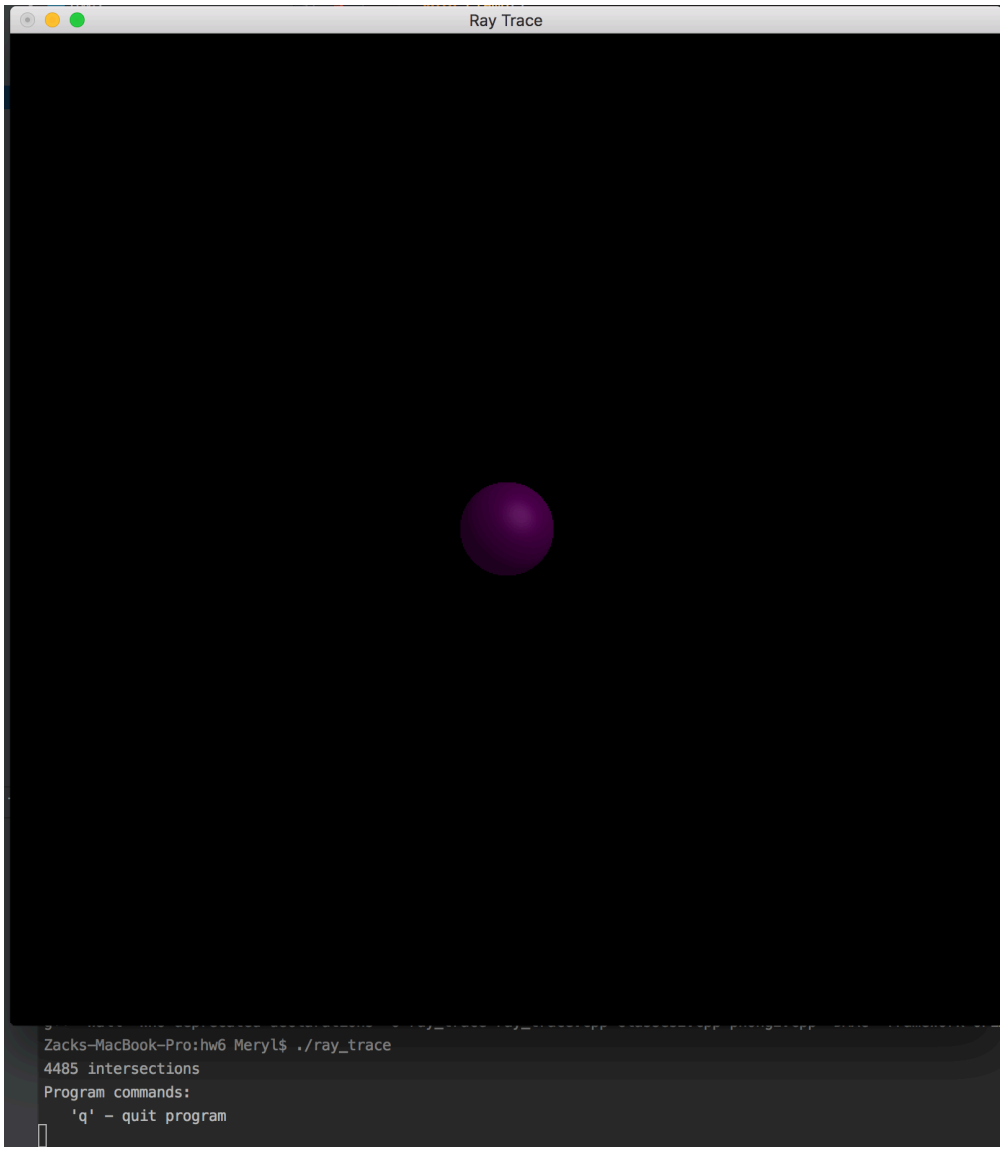
Implementation:

The implementation process was very simple, I began with Phong.cpp and Classes2.cpp with full implementations of Color, Point, Sphere, Ray, and Vector classes. As stated before, I extended the Ray class to store colors for each ray to make displaying easy. Other than that it was just about calling everything in the main() method and then iterating through the vector of rays in the display() method and assigning colors to the image[y][x][3] data structure.

Testing:

I tested the program incrementally as I was designing it. The main hurdle was making sure the sphere intersections were happening correctly so I created a counter

variable that prints the number of intersections every time the program runs to make sure I am getting the correct data. A sample output is included below.



Conclusions:

Overall the programming project was a huge success! I don't think I would make any different design decisions. Overall the project took about 3 hours to complete. As stated before, the goal of the programming assignment was to implement ray tracing.