**Programming Project 6 Report**

Student Name: Robert Johnson

Student UAID: 010796992

**Problem Statement:**

The goal of this programming assignment was to work with raytracing. Specifically, to add some features to some base raytracing code that just had the minimal features so that we can understand raytracing some more. There were no real inputs for this program. Only the image outputs. Four features were available to implement, however we were only required to pick two of them.

**Design:**

I decided to implement Feature 1 and Feature 3. For feature 1, I decided to go with some arrays to work it out. They were arrays of spheres, colors, vectors, and points/positions. These data types come from the imported ray\_classes Class that we were given. Feature 4 involved an array for keeping lights.

I would not say my design choices had pros or cons. I think adding the arrays was what we were supposed to do based on the ray\_classes and other classes we were given.

**Implementation:**

I started with the ray\_trace2.cpp code we were given, as was told to start with by the instructional document. This required also using the ray\_classes.h and ray\_classes.cpp code we were given.

For feature 1, I grabbed and adapted some code from balls.cpp, mainly the timer callback. For this feature I extended the code by taking the code of initializing the spheres out of the ray\_trace() function and into an init\_balls() function like in balls.cpp. If this wasn’t done, everytime ray\_trace() function was called in the timer callback, the spheres would respawn in random locations and not just spawn once and then move and bounce around like was asked.

I tried to work with feature 2 for many long hours but I basically lost all hope for working out this feature. I also had not realized we had only had to do two of the four features and not all of them. This gave me some hope back.

Tried working on feature 3 for a few hours as well, got stuck on that. Eventually tried feature 4. I think I set it up somewhat correctly, but I am pretty sure I did not fully get it working as intended.

All of this was over the course of about 3 days, about 6-12 hours each day.

**Testing:**

So, for feature 1, it took me awhile to get it working. After a few hours of implementation, I finally started testing my code. I felt confident that it was supposed to be working, however, when ran, the image was frozen. After talking with Professor Gauch, I did a little debugging, put some code in to print out my values of the positions to make sure they were being changed in the timer callback. Then, I put a line of code in ray\_trace() function to make sure it was being called. Both were showing everything was supposed to be working as expected. Eventually I realized that I was changing the positional values correctly in the timer callback, however I was never setting the spheres center values to the new positional values, so the spheres were not actually being moved. After adding one line of code, my hours of headache had been solved and Feature 1 was complete.

Below is an image taken mid-bouncing!

Icon

Description automatically generated

As stated in the implementation portion, I worked on features 2,3,4. I ended up scrapping what I had for features 2 and 3 because I could not figure them out. Feature 4 I got some correct code for it; however, I am pretty sure I did not get it working correctly. As of me finishing, I think it is just using the last light source in the array and not adding up all the object colors from each light source. However, not enough time nor anyone to help me fix it with the time I had left before it was due. So, no picture to show for this. I did notice the program started running terribly slower after I added the code for this as well.

**Conclusions:**

Overall, tough little project, however, to be expected from the last project of the semester. Raytracing is a topic I still have not really wrapped my head around. Computer graphics in general really, I still have a lot to learn and understand. I understand bits and pieces just not how all of it works and comes together, I guess. Well, I might understand it more than I am giving myself credit, just not how to code it. As usual, should have probably started this project a little earlier so I had more time to get help. Had a fun semester, learned a lot. Did not really know what to expect from this class whenever I signed up, but I feel like I enjoyed it and learned more than expected. Raytracing the hardest subject we went up though.