**Final Project**

A group of planets in space

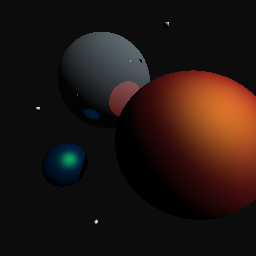
Description automatically generated

**Final Output**

Lock One. Lock Two. Lock Three. Loch Lomond!

For my project I attempted to go as far as I could with my raytracing algorithm. My first task was to fix the resolution issues. I realized I had inadvertently hardcoded 256 resolution into my code and fixed that pretty quickly. Secondly, I implemented recursive reflection tracing. This step was both simple and tricky to implement. I had to refactor the phong calculation into a separate callable function that would do all necessary calculation within itself and return the values. The recursive part after that was relatively straightforward. I am not certain that it works exactly as it should however. It definitely improved the reflections but I’ve never seen it actually reach the max depth recursion, however I have confirmed that it does recurse at least a few times. My final step was to attempt to implement distributed jittering sampling. I however was not able to accomplish this task. For some reason my attempts at distributed sampling always resulted in a completely purple picture. I had hoped to turn that sphere in the back into the death star and have it shoot a laser at the red sphere. I believe, avoiding extensive tinkering with triangles and planes in json, this would have required texture mapping which I did not get to. It also would have required spot lighting that would have raised lots of interesting challenges in my implementation. I wanted to stick pretty heavily to my original scratch ray tracer implementation done in lab 4 so I did not take advantage of any external libraries. My sources used for this were essentially the slides and a ton of google searches I forgot to save. I found gathering new information somewhat difficult on the topic as all online walkthroughs or demos rely pretty heavily on libraries that I wanted to avoid.

Original output for lab 4



I’m happy with the reflection improvement. I wish I could’ve gotten distributed sampling to work better as most of my time on this project was devoted to implementing that to no avail.