

Karina Rigby
A1, CSC 305
February 3, 2018

Assignment 1: Ray Tracer

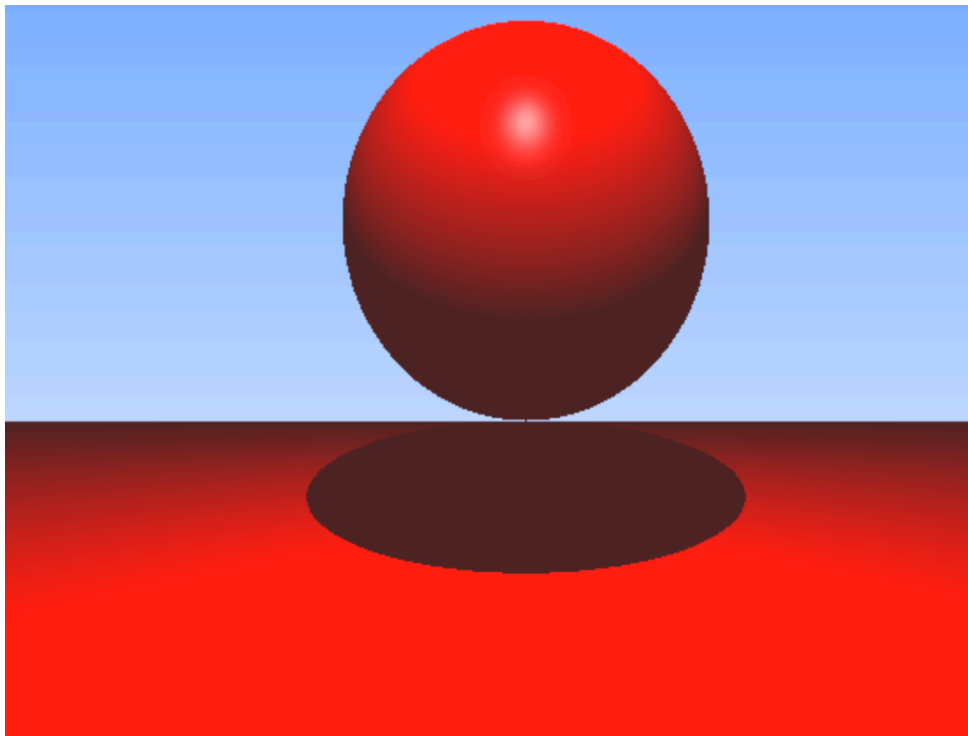
The implementation was largely assisted by the textbook by Peter Shirley, as well as “Ray Tracing in one weekend” by Peter Shirley.

I mainly based my design from an object-oriented perspective.

I created the class “hitable”, that specified which objects could be intersected with the ray. This included “hitable_list”, “sphere” and “plane”. Hitable_list is a class that stores objects, and has a virtual member function to iterate through each object in the list to find the intersection point.

Using this approach, I was able to define a world, and when I called “hit” and passed in the parameters, the program iterated through calculating the objects’ intersection points, and kept track of the closest point to color later.

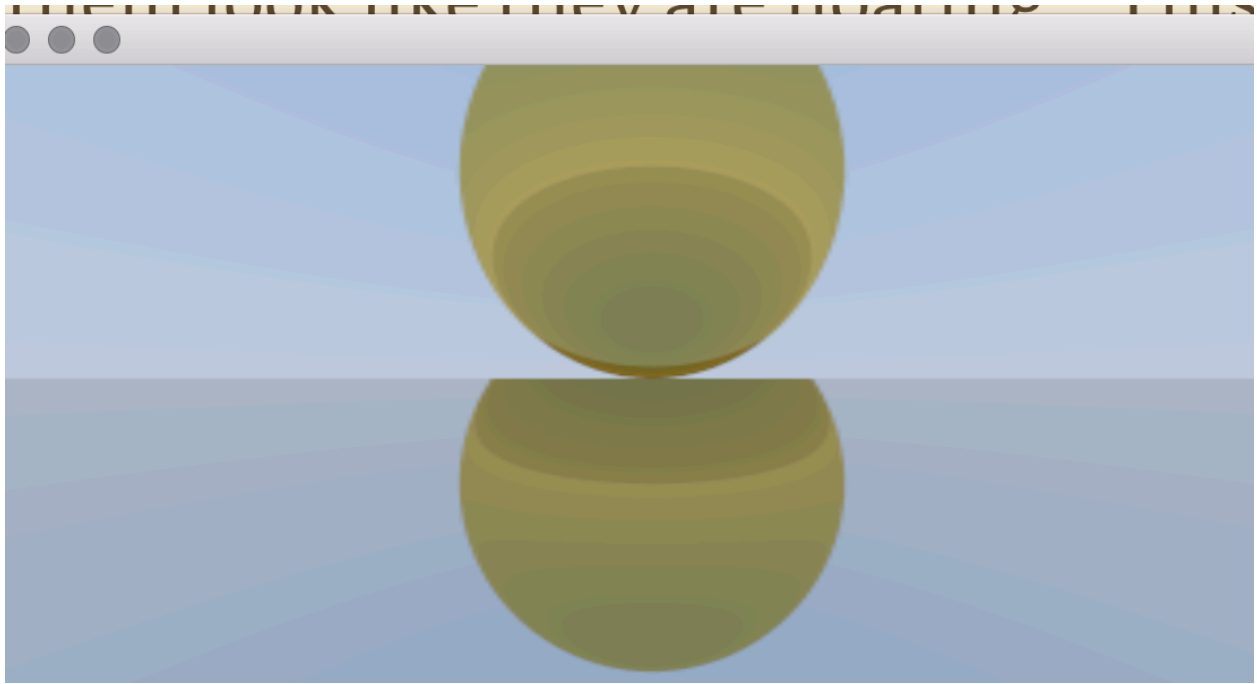
I implemented the Phong shading model, using the addition of ambient, diffuse, and specular lighting.



To calculate the shadows, once the point of intersection was found, I shot a ray back to the origin to see if there was an object in the way, and if there was, then I colored it the dark color.

To cover the objects in material, I used the abstract class “material”, to allow the hitable objects to be assigned a material, like metal or diffused lighting. I implemented the metal class, to calculate the reflection given the normal and the incoming ray. **

I was able to successfully create the reflective surface using `get_color`, a recursive function.



In this implementation, I also anti-aliased the image, by sampling multiple randomly generated points within each u and v section of the pixel, and then averaging the result.

My ray tracer is able to put in multiple objects into the scene, but I did not include a picture because it takes too long to render the image.

Overall, my tracer implemented having a light source, shadows, ray and sphere intersections, phong lighting model, anti-aliasing, and reflection.

Reading the textbook and the Ray Tracing in One Weekend book provided assistance with understanding how to implement the requirements.