**Ray-Tracing**

Okay, so first off the program that I used during this assignment was POVRAY. I find this program much easier to conduct raytracing, as I find the coding to be much easier and be able to produce animation at a much faster rate. What I was able to achieve in this assignment was the implementation of some of raytracing key elements; just as changing the location of the light source and blighting and diming the light. From my observations, the ways light is handled in raytracing is a big part of using raytracing over any others type of graphic programs that are very limited on implementing lighting in a graphic scene. I will go more in depth on how visual techniques are used in raytracing, but right now I’m just explaining my observation. Always looking at my GIF file, once I construct the two objects that I made, I added some balls into the scene. I then proceeded to move the light source to the left of my environment and notice the shadows of each of the balls were responding to the movement of the light. I then dim the light, and the shadows from the balls got more darker, and still interact whenever I continue moving the light source more to the left. I then change the color of the balls, and for each color change, the reflection of the balls to the two shiny objects also change as well, some darker than others. Anyways I started to move the camera as well, and for each change of view for the camera, you noticed the reflection of the balls change as well. At the end, I decided to add a big giant red ball that go over the two shiny objects and noticed the reflection and the shadows that the big ball cast over my environment scene.

So my understanding of how raytracing works is from this website <https://www.scratchapixel.com/lessons/3d-basic-rendering/introduction-to-ray-tracing/how-does-it-work>

Okay based on this website, it said that we first need to produce a 2d image, and then we use a graphic concept called image plane. This concept allows for 3d objects to be represented by 2d surface, due to the use of a canvas, or more specifically the camera. This also give the illustration of perspective projection, which is a technique that real life artists use to portray their painting as 3d, since the 15th century. Anyways the next section talks about coloring and lighting that are being used in raytracing. Lighting is reflective to real-life scientific laws about the behaviors of a photo, and how it interacts with its environment. Photos can either reflect, absorb, or transmit whenever they hit an object, however the sum of all three behaviors once the photos interact with an object must be the same to the total number of photos that enter the scene of where the object is at. For example, if a white light hit a red object, the green and blue lights get absorb, while the red light is the only one that get reflected.

Let talks about forward raytracing. This technique tried to mirror real-world behaviors into a computer-generated world, where rays come from the light source to a object and then to the receptors. Basically, whenever a photo reflect back to the user (more specifically the image panel), then the pixels of the image panel will grow in brightness (greater than 0), which will repeat till all pixels are adjust for that scene. However, much like the real world, photos do not directly hit a human eye in one direction. In fact about a zillion of photos are produce that go in every directions once the photos hit an object, only a small fraction of it hit the observer eyes. This cause a long delay for a program to render a scene, as the program must wait till every photo is accounted for, until it produces an image to the user. Backward raytracing is the reverse of forward raytracing, where the rays come from the receptors, to the object and then to the light source. This allow for minimal photos to be produce, compare to forward raytracing, where whenever the primary rays hit an object, it will produce another ray called the shadow ray to produce the needed photos. If the shadow ray hit another object, then our primary ray will return to the users a shadowy view, thus the name shadow ray.