Holden Profit
CS500
Spring 2019
Instructor: Dr. Gary Herron
Project 1: Basic Ray Tracing

For the first project, the basic intersections of AABoxes, Spheres, Cylinders, and Triangles was implemented as well as the use of a KDTree to greatly speed up render time. AACube, Sphere, Cylinder, and Triangle all extend the Shape class, which holds a Bbox and Material pointer as well as an interface method for the Eigen KDTree and provided a definition for the Hit method on the parent Shape class to determine when and where a ray would hit the object and the normal and material data at the point of intersection if one exists.

To validate that each was working correctly, the project was done in two parts. Part one was creating the shapes first and testing the rendering of albedo and normals without the KDTree to verify that intersections and normal generation was properly working. Part two was getting the existing system to work with the Eigen KDTree. Once the KDTree was working properly, I was able to far more quickly generate the previous images as well as the depth image of the scene.

Why Each Is Important

Normals Image – This image displays whether or not the normal vectors returned from the ray's intersection with an object are correct. While not important just yet, the normal of a surface will become hugely useful when calculating light, reflections, and refractions in future iterations of the project.

Depth Image – This image displays the t value for the ray divided by 10. This is a way to correctly determine that intersections are happening as well as showing that the t value does, in fact, vary depending on the surface struck and it's distance from the camera. Distance might be useful for subtle lighting effects in the future or color attenuation when travelling through a glass object or water.

Albedo Image – Simply shows that the material data associated with a point struck matches the expected object. This will be of great importance for all lighting techniques implemented in the future.