CS 535 Computer Graphics
Programming Assignment 3 (40 points)
Due: 12/07/2024

[Ray Tracing]

Modify your 2nd program so that you would be able to use ray tracing technique covered in class to render the objects in your scene (see the first two figures below) or a new scene (such as the second two figures below). The goal is to (significantly) reduce the time required to generate a high quality animated image with the support of openGL pipeline's hardware computing power.

You need to define two point light sources far enough so that most of the objects in the room are on the same side of the light sources. At least two objects in your scene must be transparent, such as a piece of thick glass and a crystal ball. At least one of the primitives use in your scene is a cylinder, one is a cone and one is an ellipsoid.

You need to ensure that the ray tracing process does not get into an infinite loop for any pixel of the screen. This can be avoided by setting up an upper bound on the depth of the recursive tracing process for both the specular reflection ray and the refraction ray for each pixel of the screen, as we have discussed in class. For each primitive used in your scene such as a cylinder, there should be only one ray-cylinder intersection function used in your compute shader program. Every instance of that cylinder must use that ray-cylinder intersection function to do ray-cylinder intersection test for each ray, that means the ray-cylinder intersection function must be performed in the local coordinate system of the cylinder, not in the world coordinate system of the scene.

At least one of the objects in your scene is a three-blade or four-blade floor fan (like the one used in your second programming assignment). The fan should be built with cylinders and ellipsoids only and it must be spinning. To help you with this assignment, a sample program is placed on the class website for your reference. You can find this sample program in the subpage "Ray Tracing". But the program you turn in **must** be a program of your own.

Mail me a copy of your program and a screen shot of your program output, like the examples shown below, on or before 12/07/2024. Then come in during my office hours between 12/07/2024 and 12/14/2024 for a demo.