

CS 372 Assignment 3

Due date: 2008-10-23 7pm

This assignment is worth 12.5% of your final mark.

The 3D animation studio piCSar has hired you to build a new ray tracer. The analysis team has broken down the development of the ray tracer into 4 simple steps, which are given below. After completing each step, you should make a copy of your project folder, so that you have a folder for every question at the end. Submit all the four project folders as a single ZIP archive to the assignment drop box. Each project folder should only contain the `.h` and `.cpp` files.

Question 1: Ray Casting (40%, 8 points)

The first step is to build a ray caster, i.e. a ray tracer that traces only primary rays. Use the source code from the lectures as a template (on the website). Set up the camera so that the eye position is at $(0,0,4)$ and the u , v and n axis is aligned with the x , y and z axis respectively. The viewplane parameters should be $N = 1$, $W = 0.5$, $H = 0.5$. At this stage, just use the diffuse color of the intersected object as the pixel color.

a) The First Image

Render an image with a single unit sphere at the center. The sphere should have your favorite color. The background should be light blue.

b) Adding a Plane

Implement a class `Plane`, which is a subclass of `SceneObject`. `Plane` should define a normal `Vector n` and a distance from the origin `a`. Now add a green plane to your scene with a normal of $(0, 1, 0)$ (i.e. pointing “upward”) and a distance from the origin of -1 (i.e. one under the x - z plane).

Question 2: Lights and Shadows (30%, 6 points)

Now it is time to add proper light sources to the scene. Set up an array of `Light` objects containing a single white light at position $(-2, 2, 2)$. Make sure that the sphere and the plane have their ambient, diffuse and specular colors and their shininess set.

a) Phong Illumination

Modify the `shade()` function so that for each shaded intersection point, the ambient, diffuse and specular reflections are summed up over all the light sources and returned as the color. Use the Phong equation with the halfway-vector approximation. Fine-tune the lighting parameters until you get a nice highlight on the sphere.

b) Shadow Feelers

Modify the `shade()` function so that a “shadow feeler” ray is constructed and used to check if there is anything between the illuminated point and each of the light sources. Add the diffuse and specular reflection only if there is no object between a light source and the point. Then, add another light source with colored light.

Question 3: Reflection (15%, 3 points)

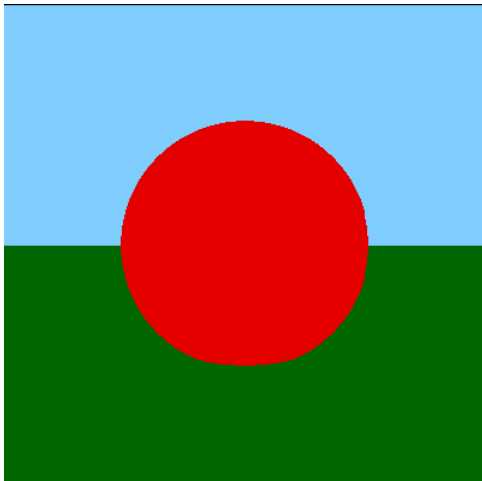
Modify the `shade()` function so that the color is influenced by the reflections from other surfaces along the ray. This means you have to construct a reflection ray, calculate the next intersection of that ray, and call the `shade()` function recursively. Dampen the reflected color with the reflectivity of the surface. Limit the reflections to a maximum number of 10 reflections per pixel and only those objects that have a reflectivity of at least 0.1.

Question 4: Transformations (15%, 3 points)

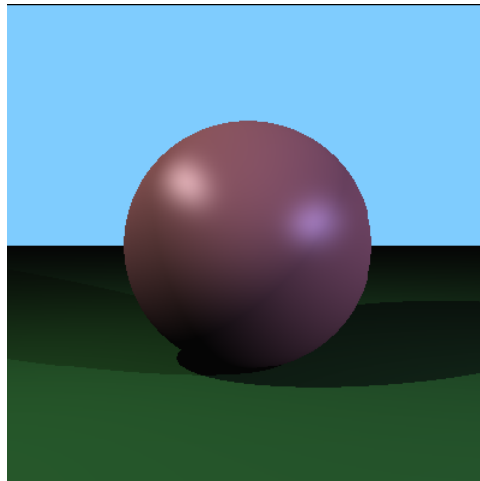
At this stage, it should be possible to render transformed objects. The `SceneObject` class already contains the fields `scaling` and `translation`, which should be available to first scale and then translate an object. Modify the `intersect()` function so that each ray is inversely transformed with the modeling transformation given by the `scaling` and `translation` fields. Then, modify the `Normal()` methods of the `Sphere` and `Plane` classes so that the normals are adjusted accordingly. Add two more spheres to the scene, using different translations and scalings (e.g. a “flat” sphere).

Sample images:

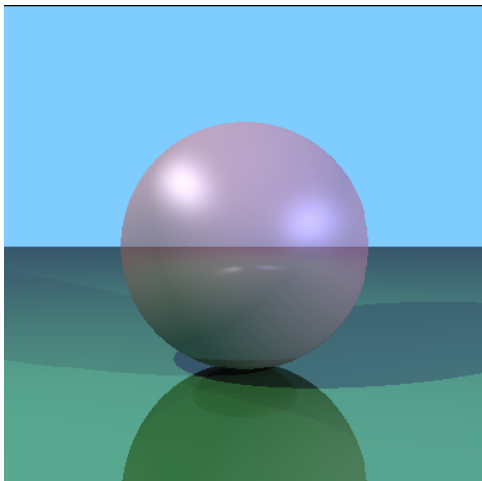
Question 1



Question 2



Question 3



Question 4

