

Fall 2006
22C:151 Introduction to Computer Graphics
Assignment 9

Due: Wednesday November 8th at 11:59 pm

Goal: Write a basic ray tracer.
--

Problem 1 (5 points): Think about your imaginary camera that is going to take a ray traced picture of your synthetic scene. You know the eye position, where you are looking at, and the up direction. You also know the field of view and aspect ratio. Using this information, you need to compute a ray leaving the eye that passes through each pixel in your image. Write the code to compute a ray for each pixel.

Hint: This involves writing the `Camera::Camera()` constructor and `Camera::GenerateRay()` methods in the ray tracing framework provided for you on the webpage.

Problem 2 (8 points): Now we need to determine if our rays see any objects. This means we need to intersect a ray with some objects. For this assignment, you'll need to implement two different objects: a sphere and an infinite plane. Write code that determines if a ray intersects a sphere and write code that determines if a ray intersects an infinite plane.

Hint: This involves writing the `Sphere::Intersect()` and `InfinitePlane::Intersect()` methods in the ray tracing framework.

Problem 3 (5 points): Now that we know if our eye sees any objects, we need to light them. To light an object, we need to have a surface normal. Write code that returns the surface normal at the hitpoint along a ray.

Hint: This involves writing the `Sphere::ComputeNormal()` and `InfinitePlane::ComputeNormal()` methods in the ray tracing framework.

Problem 4 (8 points): Now we're going to add simple lighting. Given an intersection position, a normal, and a light position, we can compute simple diffuse (or Lambertian) lighting. Remember, if \hat{N} is the unit-length surface normal at a point, \hat{L} is the unit-length direction to the light, and C_L and C_M are the colors of (respectively) the light and the object's material, then the color at the point is: $\max(0, (\hat{N} \cdot \hat{L}) * C_L * C_M)$.

Hint: Adding lighting involves writing the `LambertianMaterial::Shade()` method in the ray tracing framework.

Problem 5 (4 points): In addition to rendering the required scene posted on the webpage (and included in the framework), create two other scenes to test your ray tracer. Post images of these scenes on your web page.

Extra Credit (5 points): Load and apply textures to the spheres and infinite planes in your scenes.

NOTE: A "README" file is required in order to get full credit! Please tell us how to compile and run your program to recreate ALL your scenes. If there are any special command-line parameters, make sure you specify them!