CS4499
Project 5 - ray tracing
Due Dec 4, 5:00 pm

Overview

In this assignment, you will implement a ray tracer. Skeleton JavaScript code is provided for you, but you may implement your ray tracer in a different language if you wish, although it is discouraged for this assignment. We will not be writing any shader code.

Skeleton code

Open **ray.html** in a browser. You should see a blank canvas. Add `?test=1` to the URL and refresh. You should see a test image. Open **raytracer.js** in a text editor. The method being called is `getColorTestPattern()`. When you remove the test flag from the URL, the method `getColor()` is called instead, which is the method you'll be implementing. See the comments to `getColor()` for instructions.

Ray tracing is expensive. To help speed it up you can ray trace your scene at varying resolutions and in various areas. For example, add the following to the URL and refresh:

```
?test=1&resln=32&window=[0,0,0.75,0.5]
```

This will render a small window of the test image at 1/32 the resolution. This way you can speed up your testing by rendering targeted areas at lower resolution. Be sure not to put any whitespace into the URL.

Suggested approach

Consider developing your code as follows:

1. Compute the ray from the eye to the pixel. You will need to decide where the eye is and at what z coordinate the screen is at.
2. Add a sphere by defining the position and size of the sphere and adding ray intersection code. If the ray intersects the sphere, color the pixel red.
3. Shade the sphere using Phong or Blinn-Phong shading.
4. Add a plane.
5. Add more spheres.
6. Implement the `trace()` function as we discussed in class and as is described in section 12.3.1 of the textbook.

7. Go crazy. Try adding other types of objects, varying normals to make shading effects, etc. Place objects off-screen so you only see their reflections. Add multiple lights with varying hues and intensities. Texture map surfaces with functional effects (checkerboards, stripes, trigonometric functions, fractals, etc).

Scoring

For full credit your ray tracer must support the following:

- spheres
- planes
- light sources
- ambient, diffuse and specular lighting
- reflection

Extra features include:

- additional object types, including triangles
- textures (either from images or functional)
- varying normals
- other cool stuff

The scoring breakdown is:

1. 20% - Render sphere
2. 20% - Render plane
3. 20% - Shading
4. 20% - Reflections
5. 10% - Extra features
6. 10% - Coding quality and style

Notes

1. Submit your code, a **README.txt**, *and* a screenshot of your ray-traced image(s).
2. You may not use any third-party libraries. You may consult sources outside the textbook and class discussion only after implementing the basic features.