CS770/870 Assignment 7

- Out: Wednesday, October 28th.
- Due: Wednesday, November 4th.

Overview

In this assignment, you will write a ray caster. This is a ray tracer that uses only primary rays (no reflected, refracted or shadow rays). You should be able to render both spheres and triangles.

Pseudocode

You are free to implement the ray tracer any way you see fit, but I suggest you follow this pseudocode:

```
for each pixel in image {
    make_a_ray(s=pixel, v=pixel-eye)
    pixel = ray_color (ray, scene)
}

ray_color (ray, scene) {
    hit = first_hit (ray, scene)
    obj = hit.object
    return obj.color
}

first_hit (ray, scene) {
    list = (empty)
    for each object in scene {
        if (ray intersects object at location) {
             hit = new hitpoint(location, object)
             list.append(hit)
        }
    }
    return list.get_minimum()
}
```

The Scene

Your picture should render a scene that contains several spheres, and several triangles.

C++ Classes

You will probably need to implement the following classes to represent aspects of the problem:

- `Object` : a geometric shape in the scene. The most elegant way to proceed is to make this an abstract class, with subclasses `Triangle` and `Sphere`. This class should define an abstract `intersect(ray)` method, which sets a `Hit` object if the ray intersects the object. If you can't get abstract classes working, your scene could simply consist of two arrays: a list of `Triangle`s, and a list of `Sphere`s.

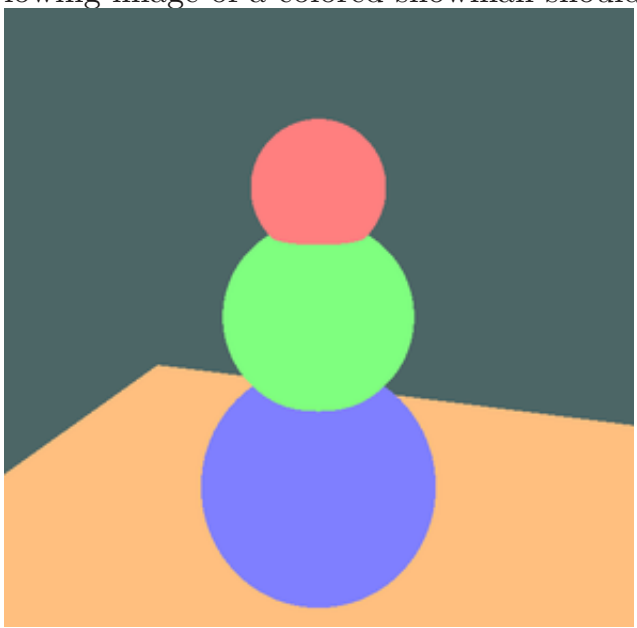
- `Triangle` : (a subclass of `Object`) a polygon, contains:
  - three vertices
  - surface normal
  - color

- `Sphere` : (a subclass of `Object`) a sphere, contains:
  - center location
  - radius
  - color

- `Hit` : describes the intersection of a ray and an object, and contains the following data members:
  - location of intersection
  - color of object that was hit

More Information

The starting code defines a scene with three spheres and two triangles. It gives the camera an initial position, which you can change. From the initial position, the following image of a colored snowman should be obtained:



.

Turning in your Work

When you are done, go to `mycourses.unh.edu`, find the course, Assignments, and find the assignment. Then upload these files:

- `rt.cpp`
- `triangle.cpp`
- `sphere.cpp`
- `hit.cpp`

If you modified any other files, upload them too.