Project 3: part b, Ray-Shader: Out of 50 possible points. Due 11:59PM Thursday Nov 15

Purpose: is to understand how shading works for ray-tracing. Additionally 5450 to understand how to incorporate shadows.

Use code from Proj 3, part a, see image

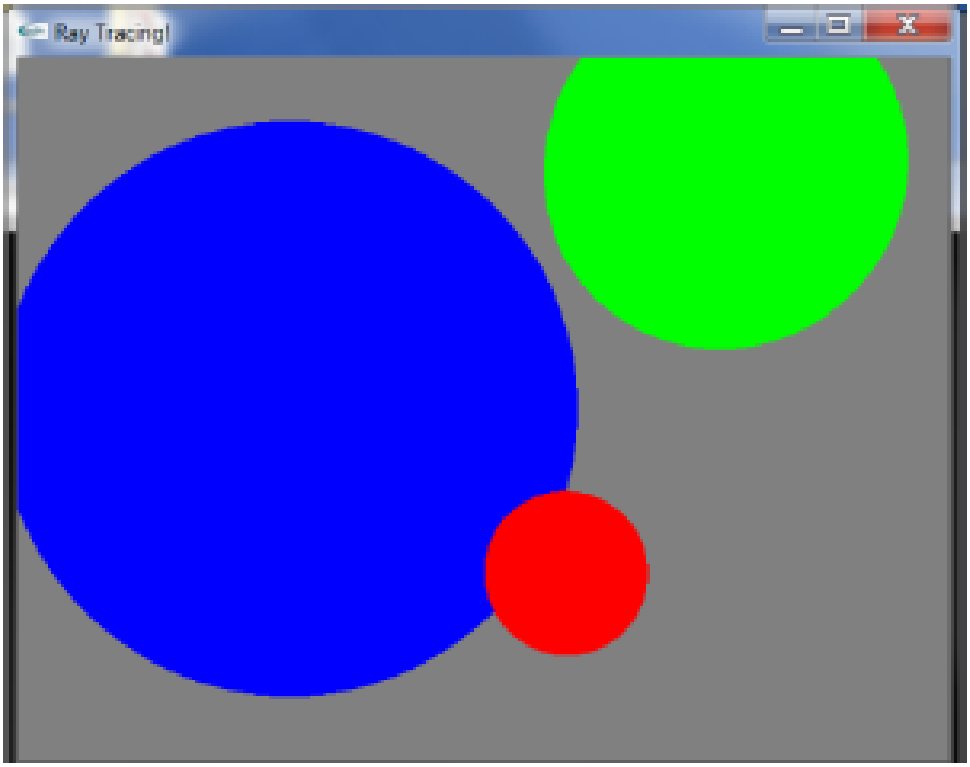

Figure 1: result from Project 3, part a

Your project should have the following elements within a 3D world coordinate system:

1) **Set up Lights in your environment.** Set up a global ambient light with 25% red, 5% green, 5% blue, and fully opaque (alpha =1). Set up one point light at position (200,100, 50) that has ambient with 65% color and fully opaque, giving off zero amount of diffuse and specular light but fully opaque. Set up one bluish spot light, located at (-1,0,1) and directional vector of <0,0,-1> with an angle of

30% and exponent of 0.5. The spot light should give off zero amount of ambient and diffuse light but fully opaque, and specular 40% red, 40% green, and 70% blue. Add one yellow directional light in the direction of <-400,692,0>, give off zero amount of ambient and specular light but fully opaque, with diffuse light at 80% red, 80% green, and 60% blue.

2) **Calculate the 3d hit point.** Using the hit time found, calculate the 3D hit point of that object. Additionally use this hit point to calculate the vectors L (to the light source) (warning: be careful what you do depending if it is a directional light or a point light), V (view vector, to the eye), H (halfway vector), and N (normal- be sure it is facing the right direction).

3) **Modified phong shading** (what we covered in class). Implement in the shade function which calculates the total illumination that follows the formulas for the modified shading model. This should include the emissive, ambient, diffuse, and specular to find the total color values to shade each pixel. Keep in mind depending on the type of light source you will do things slightly differently. Your program should look like Figure 2 before moving on to Project 3, part c.

Additional Tasks for 5450 (and extra credit for 4450):

4) Change your initial ray implementation to Orthographic projection when computing the rays.

5) Add a keyboard callback to switch between Perspective (original ray code) Key P or p, and Orthographic in item 4, key O or o

REMEMBER: You should not be using OpenGL at all for transforming objects, defining objects, or displaying objects (other than pixmaps).

You will be graded on elegance of code and the completion of implementation details for elements specified. Elegant code should include programming concepts and formation of classes and functions that you have learned in Data Structures and Algorithms course.

Point values are distributed among the categories as follows:

4450:
Element 1 – 10 points
Element 2 – 15 points
Element 3- 25 points
5450:
Element 1 – 5 points

Element 2 – 20 points
Element 3- 20 points
Element 4- 10 points
Start Early! Ask Questions! Don't forget to comment your code!

. Visual Solution:

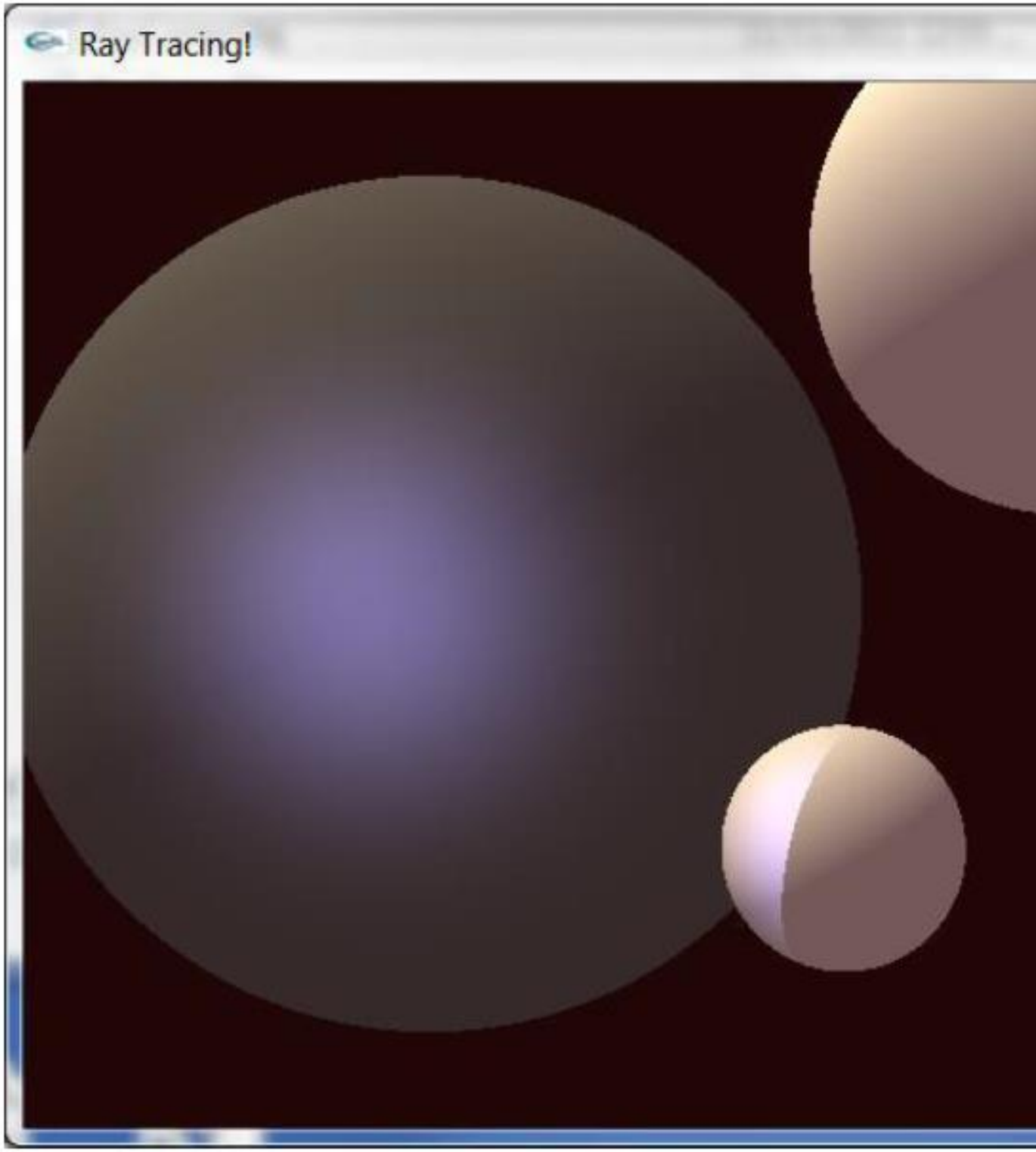

Figure 2: Depth =0 or 1;