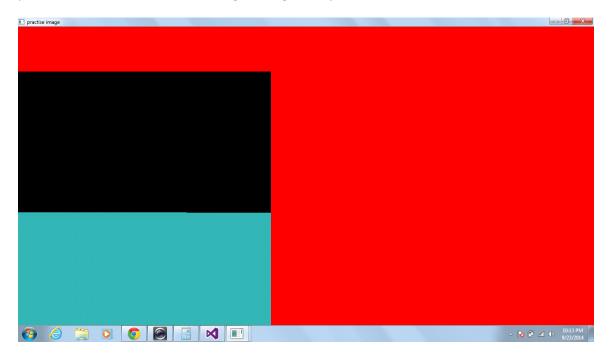
Report

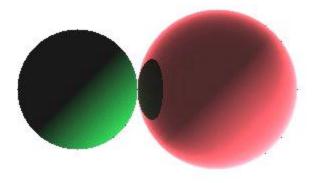
Name: Garima Singh

UFID: 51975877

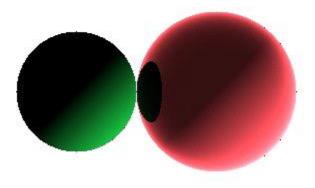
1) This is the image generated as a part of question1. Just colouring a few pixels on the screen with a different colour using glDrawPixels. This had great learning value as to how the glDrawPixels accepts its pixel values. It seems trivial but can go wrong terribly.

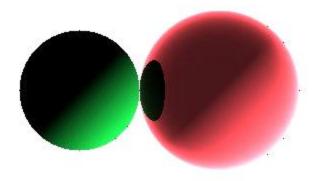


2) This is the image generated in the second question, by the ray tracer using parallel ray tracing(look at vector at (0,0,-1) and light direction of (1,1,0)). Here, what we see is an inverted image though(discrepancy in the way that ofstream and gldrawpixels accept their input).



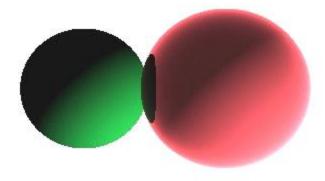
3) As per the third question, I've added ambient and diffused shading to both the spheres. I've also added the blinnphong shading but isn't visible. (lines 191 and 203). First image is without ambient lighting, next is with ambient lighting.





4) The red sphere has a glazed reflective surface and hence the hazy appearance as an effect of the reflection index, hence the green reflection of the green sphere. Reflection was implemented as a recursive call to the getPixelColour function (secondary ray tracing).

5)For the animation, I changed the direction of the look -at vector rotating each time with -2 degrees with respect to positive x axis. For a rotating effect, it was necessary to change the view window each time along with the look at vector. I did this by centering it at (0,0) in the x-z plane. y co-ordinate doesn't change because we are rotating the window about the y axis. The spheres start looking a little stretched up as the angle increases because I've used the same z- coordinate value for all pixels whereas actually they would change in a gradual manner from 0 to width*cos(angle made with positive x-axis). But this involves mapping pixel values from 0- width to this new projected length, which was a little complicated to do, so the animation works for small angles.



One of the images generated by rotating the look -At vector and the frame of view.

I was unable to complete implementation of tetrahedron . I have just written a function to calculate barycentric coordinates for the point of intersection.