Report for Ray Tracer

I compiled on Ubuntu 14.04

You will need libjpeg to compile successfully

It comes default on most Linux’s but I am not sure about Gentoo

To use my ray tracer you type the following

./raytracer options scenefiles

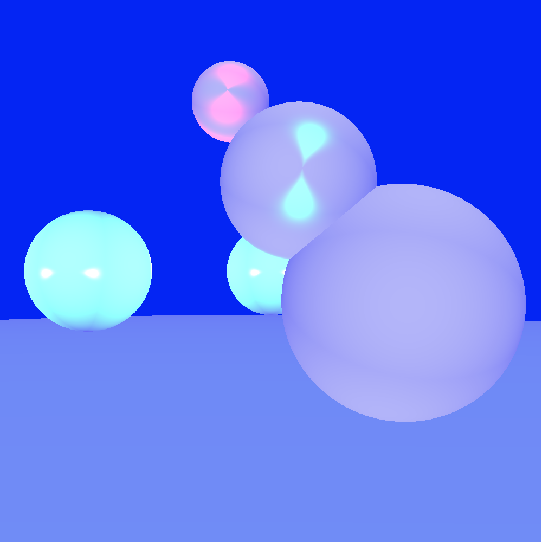
The options section can contain the following

n – for normal render

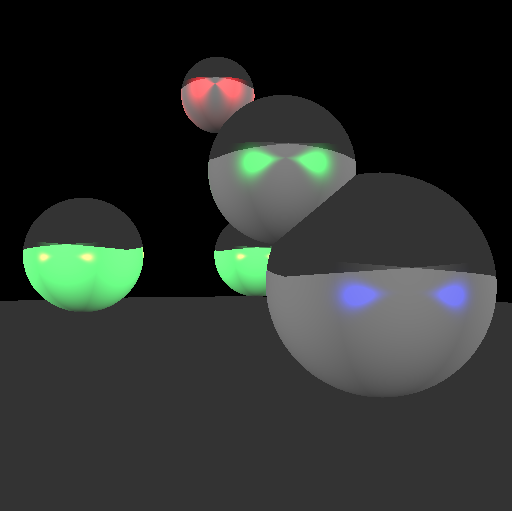
ss – for super sampling

dof – for depth of field

First lets start off by saying this was an adventure of craziness. It took me about 40 or so hours to do all the things I did including debug time. I ran into all sorts of problems first I couldn’t figure out if my normal were right or not that took me about 2 days then I finally started lighting. I did not have to much trouble with this as we had gone over it a bunch. Mainly my issue was that I kept getting this image where the spheres looked as if they had eyes. You will see what I mean in the image below although this image also highlights the fact that I had trouble realizing that glm uses radians by default because in Visual Studio it was using degrees.



After my lighting adventure I moved to shadows where I had cancer problems as well as this interesting error which I don’t even know how it happened.



After this I moved on to reflection that I didn’t really have any problems with. At this point in the ray tracer is where I started to understand how to fix my own problems easier. Next came refraction which took me a couple of days because I was getting a solid sphere at first. Finally after many hours refraction started to work but it was wrong. I fixed this by having you (Dr.King) render me a refracted image then I debugged till I got close. My refraction might still not be right but it is close. I’m not sure what the problem is now. At this point is where I used Open MP to speed up my ray tracer which helped a lot. I ended up rendering the sphere flake level 5 with a recursion depth of 5 in 35 min on a laptop. You can find the result in the cool images folder.

After this I finished my ray tracer by doing many other things. A list of everything I did is below.

List of everything I did:

Primary rays

Intersect with spheres

Lighting

Transform spheres with lighting

Multiple lights

Shadows

Reflections

Refraction

Antialiasing – super sampling

Depth of field

Ability to load more than one scene file and save them all separately

Textures using images - libjpg

Acceleration Techniques – Open MP

Also try the renderAll.sh script in the main directory for some cool scenes

Check the cool images folder for some awesome errors