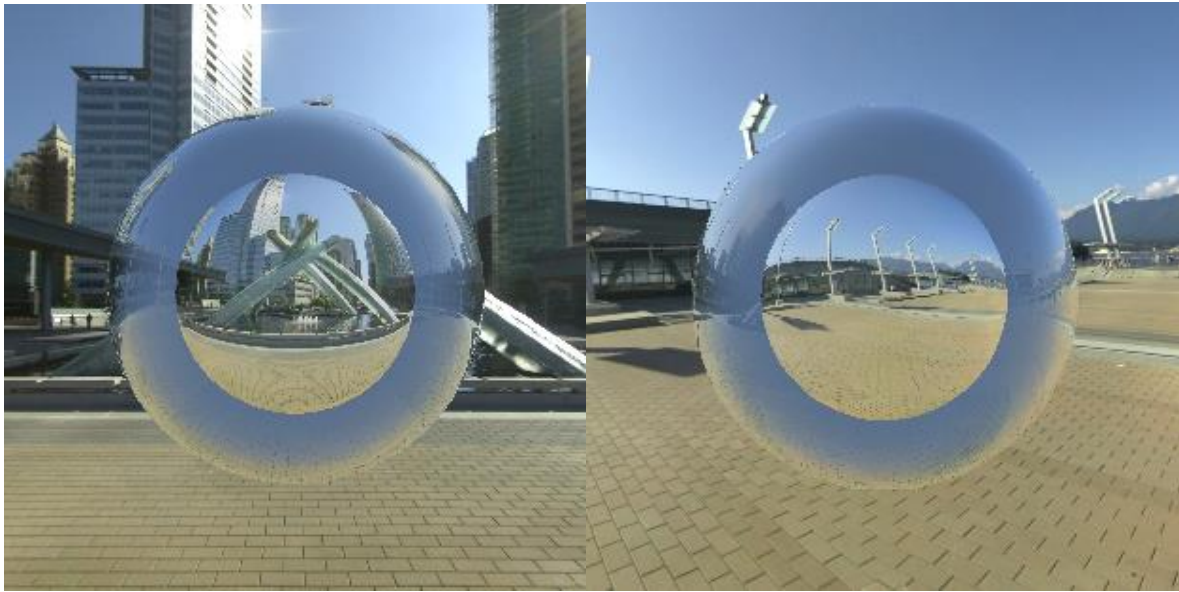


CSCI 4110U – Assignment 2

Rhys Agombar – 100515147

Reflection and Refraction

To construct the sphere and the background independently, (and thus able to have lighting effects on one, but not the other), I compiled two separate shaders, and passed the same cubemap to both of them. The background shader simply displayed the background cube, while the sphere shader computed positions to sample from the cubemap texture, using the reflect and refract functions, before combining the two and colouring the reflecting and refracting sphere. The refraction index used was 1.52, since it is similar to glass.



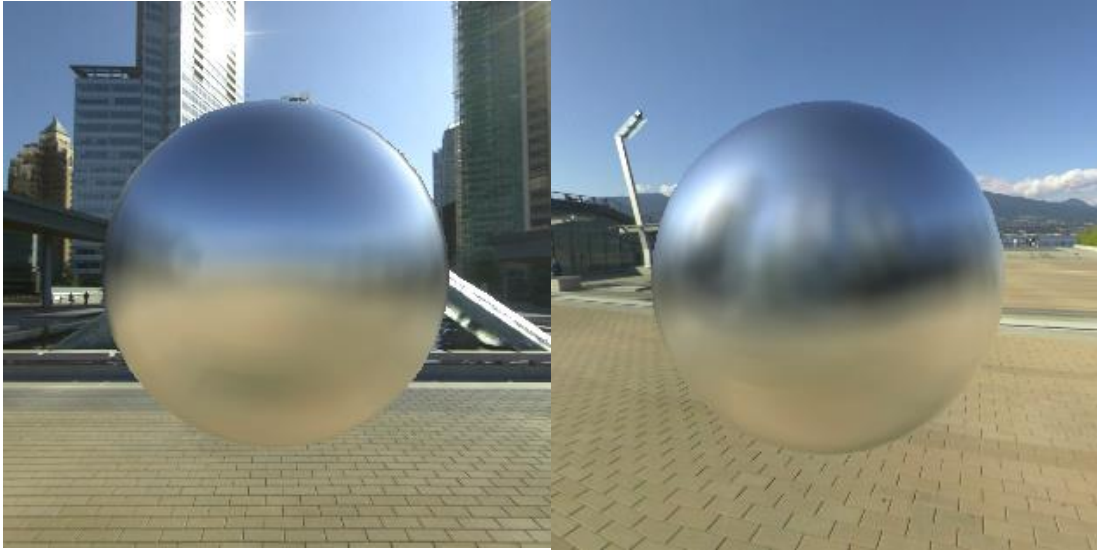
Diffuse Reflection – Part 1

This part of the assignment was a simple extension of the previous part. I expanded the shader implementation to allow for a second texture, so that the sphere could be textured with a blurred cube map while the background retained the detailed version. This produced the following results.



Diffuse Reflection – Part 2

In order to solve the issue of the lines showing up in the diffuse reflection texture, I used the uniform sampling strategy (as the random sampling strategy did not look as nice). Using the formulas given, I set the sampling radius (L) to 0.5, and the angle between the samples to be 10° . This meant we would be taking 36 samples per point on the sphere, and seems to give a decently accurate and smooth blurring look. With greater radii, the reflection started to become uniform and indistinguishable, while with a smaller radius, the reflection was not blurred enough to make a difference. The sample size was chosen because a large number of samples negatively impacted the performance, and a smaller number of samples lead to the sub-images being visible (albeit faded and offset from the actual one).



Errors:

For the last part of the assignment, I encountered issues with passing the values of θ and L to the shaders. Not because I was unable to retrieve their values, that part worked fine, but because when I set the uniforms the background texture would switch to the blurred texture. The sphere's blurring would be modified correctly, but the texture for the background would change when I tried to pass even 1 more uniform than expected. This did not happen with values hard coded into the shader (which is how this is currently set), so I have commented out the lines for passing the uniform to demonstrate that the blurring algorithms work. I do not know why the texture suddenly switches when I try to pass a float to the shader, but it does and I don't have time to fix it.