Jake Wagoner
u6048387
CS6610: Interactive Computer Graphics

Final Project: Progress Report

What You Implemented

So far, I have implemented a refraction model which uses Snell's law to compute a basic single-surface refraction model. This takes a texture as input for the background. The program takes two arguments, a mesh (.obj) and texture (.png or .jpg). By clicking and dragging, you can rotate the objects. You can also change the index of refraction by using the up and down arrow keys.

What Will You Implement

Next, I am hoping to implement Wyman's method of multiple surface refraction. Currently, the refraction is only based on the front face, this will make the refraction much more realistic. Then, because I set up the refraction to use a 2D texture as input for the sample map, I think it would be possible to perform multiple passes to render textures and pass these into the refractive objects. This would allow me to have multiple objects in a scene which could refract through each other. The performance may be impacted pretty heavily, depending on the complexity of the scene.

Finally, I'd like to try and make an object which is actually a realistic lens to see how the refraction performs when compared to real images (if I can find something good to compare to).

How to Use Your Implementation

To use the implementation, build and run the project using the visual studio solution or CMAKE. If you want to change the obj file, you may need to update the scale of the model, as the dragon.obj file is WAY larger than the teapot or sphere.

Operating System and Compiler

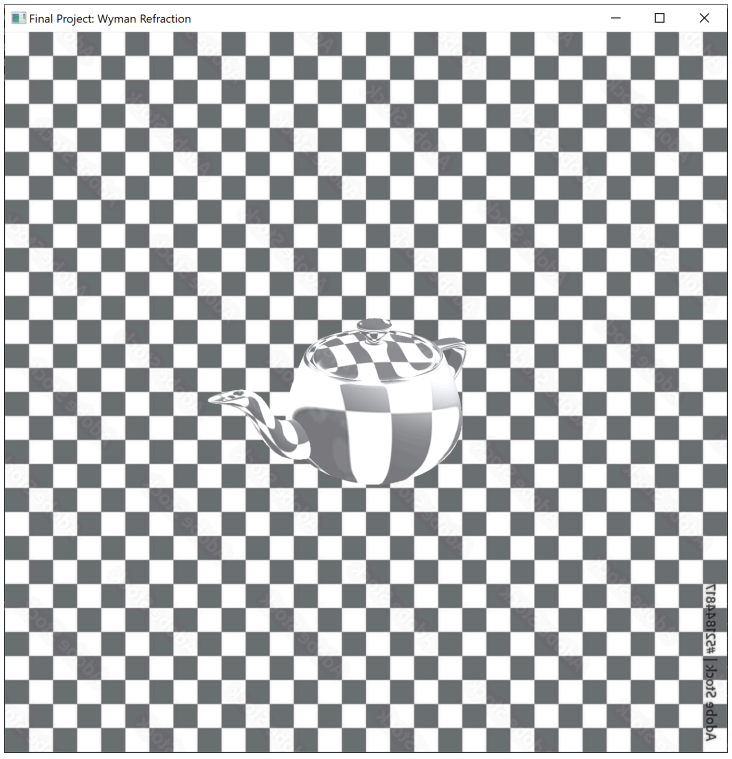
- **Operating System**: Windows 10 (also tested on Windows 11)
- **Compiler**: Visual Studio C++

External Libraries and Additional Requirements (OpenGL 3.3)

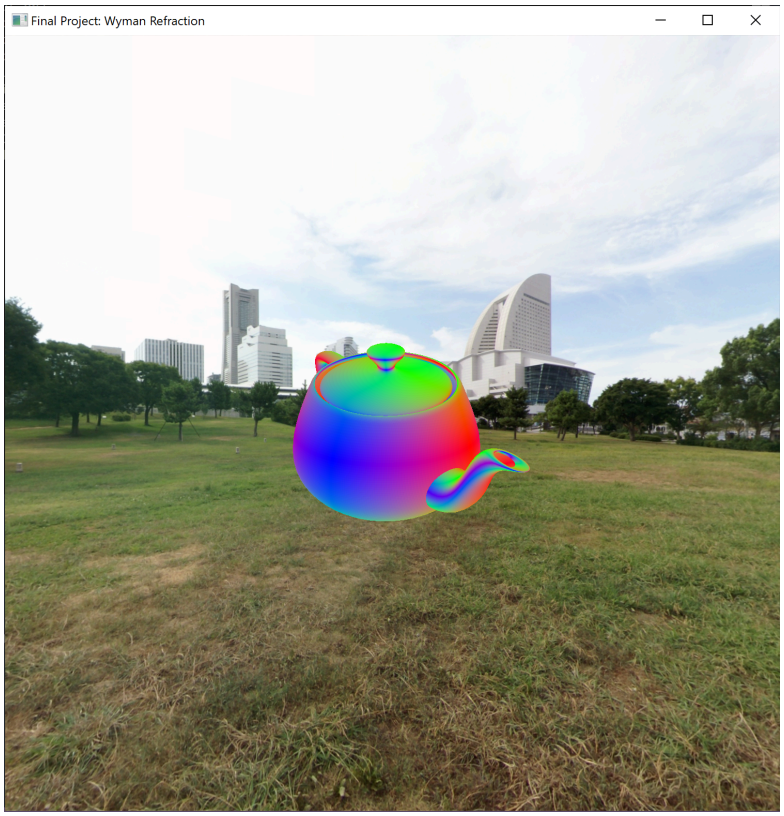
GLFW, glad, glm, cyCore, cyTriMesh, cyVector, cyGL, stb_image (in gl folder)

Screenshots shown below:

Single surface refraction (first pass attempt)

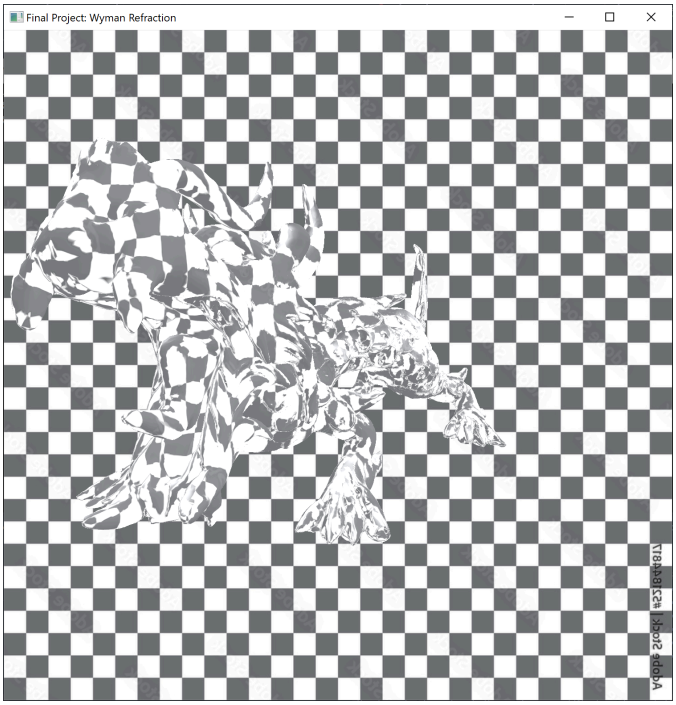


Grid for debugging



abs(RefractionDirection) visualized as color for debugging

High-Res mesh with fixed refraction direction



Debugging background for high-res mesh