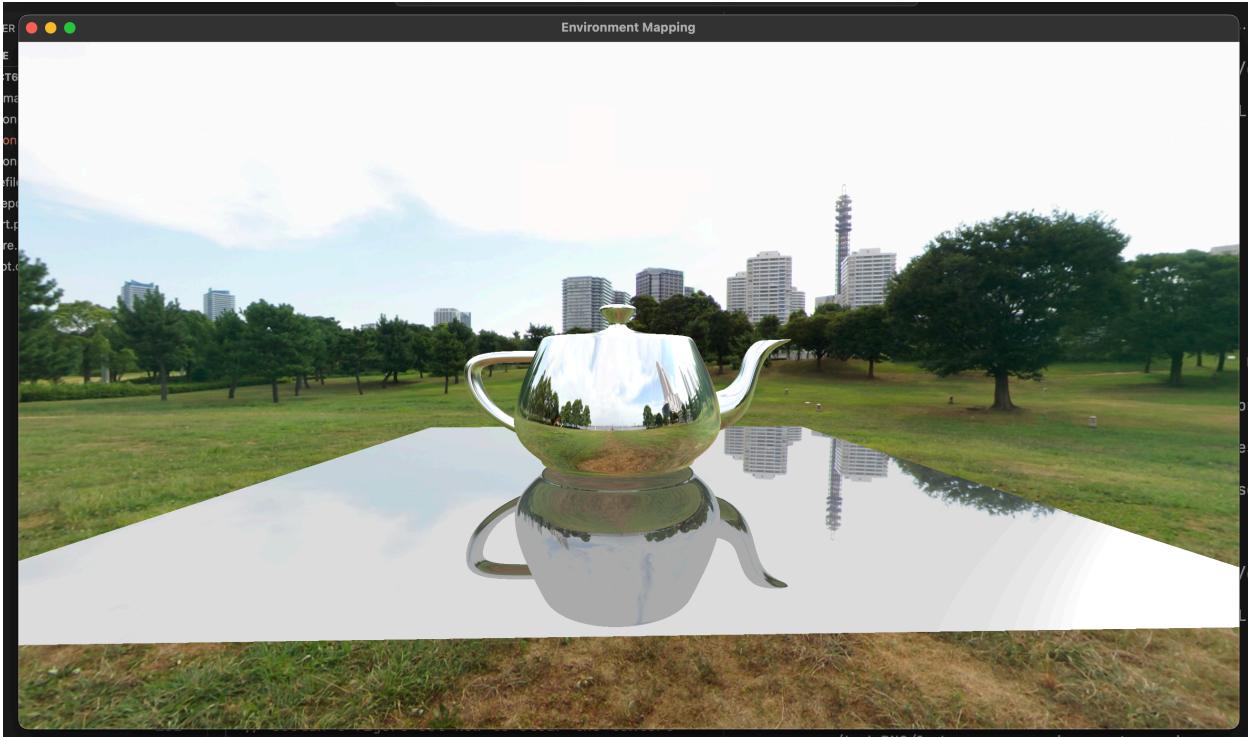


James Youngblood, CS 6610, Project 6



I am submitting this on Mar 9, having fixed the errors in my code.

What I implemented

I load in the cube map textures from a local directory, and use the cyCodeBase implementation of a cube map texture to link those with OpenGL. I've created shaders, VAO's, and VBO's for the background, the teapot, and the ground plane. I render the background first, as a single triangle, then the teapot and the teapot reflection (using a rotated MVP matrix, which then goes into a render buffer the size of the screen), and finally the ground plane. The fragment shaders transform reflection rays back into world coordinates to sample accurately from the cubemap, and the ground plane fragment shaders gets the screen size so that it can sample the reflection render buffer using the fragment coordinate divided by the screen size.

What I could not implement

I could not figure out how to do alpha blending for the reflection of the teapot and the background properly, so instead I simply check if the color from the reflection texture is black: if it is not, I render the reflection of the teapot, otherwise I render every other color. This is kind of a "hacky" fix, but it works well for this specific scene.

Also, currently the specular highlights are brightest in the direction of the light in "real" world space. They should be brightest in the direction of the light in the reflection space. I didn't find this effect too jarring, however, so I did not bother to fix it.

Additional functionalities

None.

How to use the code

My code is a single .cpp file, with shaders included as string literals in the code, so it should be easy to compile. It will expect a path to an obj file as an argument, and the cube map textures in a folder called “cubemap” in the same directory as the executable you compile. These textures should be named “cubemap_negx.png”, “cubemap_neny.png”, “cubemap_posx.png”, and so on.

What operating and compiler system did I use?

I used gcc as the compiler, on the latest version of macOS.

External libraries and other requirements for compilation

The use the following dependencies.

- GLFW (include GLFW/glfw3.h)
- GLEW (include GL/glew.h)
- OpenGL >= 3.3
- C++11 standard lib
- cyCodeBase headers cyVector.h, cyTriMesh.h, cyGL.h, and cyMatrix.h
- LodePNG

To compile on my Mac M1, I installed GLFW and GLEW using homebrew (a package manager for Mac), including them and linking to their libraries using flags -I, -L, -l, for each when compiling with gcc. I also included the cyCodeBase headers in a similar way. I included the LodePNG header, and compiled with the LodePNG .cpp file alongside my own. I linked to the pre-installed OpenGL distribution on macOS using the flag -framework OpenGL. Finally, I included the C++ standard lib using the flags -std=c++11 -lc++.

Here is the compilation command I used:

```
gcc -std=c++11 \
-I /opt/homebrew/Cellar/glfw/3.3.8/include \
-L /opt/homebrew/Cellar/glfw/3.3.8/lib \
-l GLFW \
-I /opt/homebrew/Cellar/glew/2.2.0_1/include \
-L /opt/homebrew/Cellar/glew/2.2.0_1/lib \
-l GLEW \
-I ../cyCodeBase/ \
-I ../LodePNG/
-framework OpenGL \
-lc++ \
../LodePNG/lodepng.cpp \
```

```
environment_mapping.cpp -o environment_mapping
```