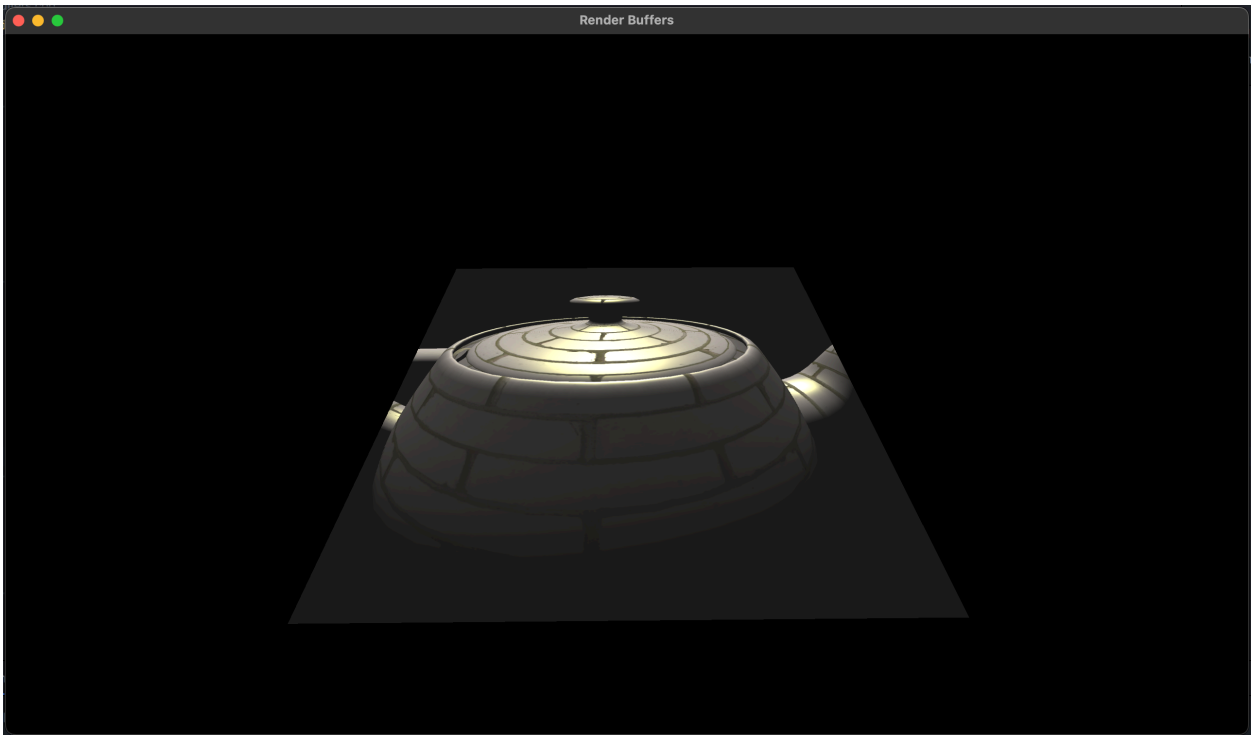
James Youngblood, CS 6610, Project 5



I am resubmitting this on Feb 24, having fixed the errors in my code.

What I implemented
I have created a render buffer (using the cyCodeBase, GLRenderTexture2D class), which I render the teapot into. I have created a new VAO, VBO, and shader program (with vertex and fragment shaders) for rendering the plane. This plane samples from the render buffer for the color of each fragment, with a small boost in lightness of the color so that the plane is distinguished from the background. Mipmapping, bilinear filtering, and anisotropy have been set for the texture sampling. I have also added controls and separate rotation variables for the plane, so that when you hold ALT, the plane will be transformed instead of the teapot.

What I could not implement
All required features have been implemented.

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. Simply run the executable that you compile, with the path to the obj file as a string argument. Note that it will look for a material and image files

associated with the obj file—these should probably be in the same directory as the code/obj file.

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 render_buffers.cpp -o render_buffers
```