

## CIS 441/541: Project #3A

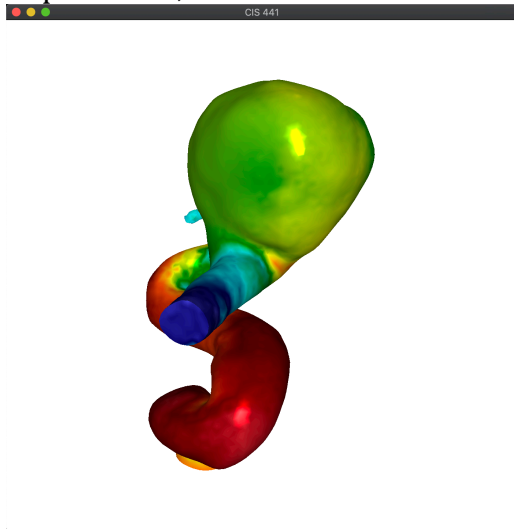
### Goal:

Extend your 2A code to use 1D textures. You will use 1D textures to achieve two effects:

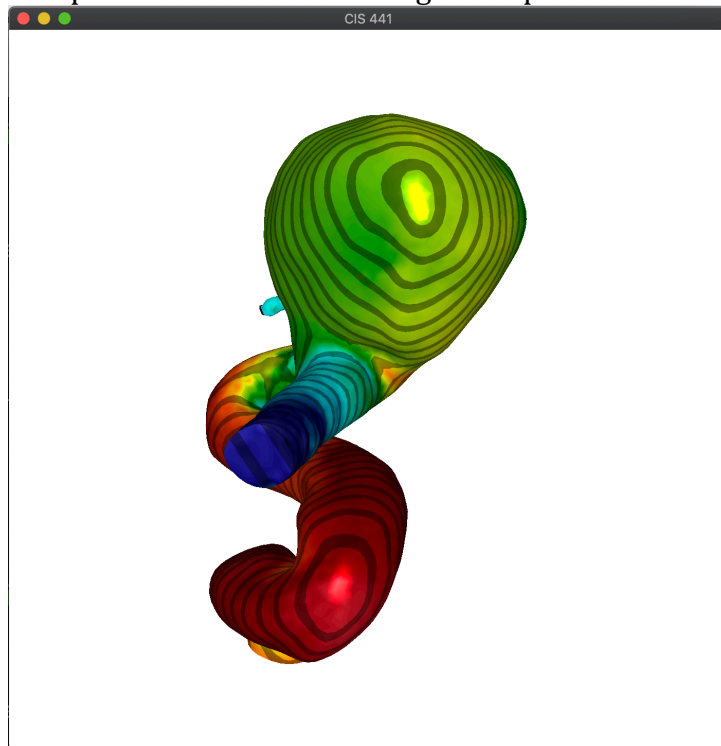
- (1) Improved coloring
- (2) Tiger stripes that show depth

### Steps:

1. Copy your project2A.cxx to a new file called project3A.cxx
2. Modify the Projection matrix to have near plane = 40, far plane = 60, i.e.,  
`glm::mat4 Projection = glm::perspective(  
 glm::radians(30.0f), (float)1000 / (float)1000, 40.0f, 60.0f);`
3. Download the functions on the course website and copy them into your project3A.cxx file. The two functions you need are “GetColorMap” and “GetTigerStripes”
4. Set the background color to be white (glClearColor before the glClear – do a Google). This will help the tiger stripes show up later.
5. Compile and run. You should see the same picture you saw when you completed 2A, except the background is white.. Note the color map is somewhat simple – blue, white, and red. We will improve this in this project.
6. Set up a 1D texture for the color map. Do not modify your shaders yet.
7. Compile and run. You should still get the same picture.
8. Modify your fragment shader to use the 1D texture. Note that the texture function assume texture coordinates between 0 and 1. But we are passing in data values. These data values range between 1 and 6. Therefore you will need to scale the data values to be between 0 and 1 and thus be valid texture coordinates:  $(data-1)/5.0$ . This should happen in your fragment shader.
9. Compile and run. You should get a new color mapping, similar to project 1F. In particular, it should look like this:



10. Add a second 1D texture, this time for the tiger stripes. Compile and run.  
The output should look the same. Note: this texture is just a single unsigned character for each element. So it should be `GL_RED`, not `GL_RGB`.
11. The tiger stripes give depth information. We don't have depth information readily available, but we can get it. In particular, modify your vertex shader to produce a new output called `depth`. The depth can be found from the `gl_Position`. Don't forget to divide by `W`. Also don't forget to modify your fragment shader to take the depth as an input.
12. Compile and run. The output should still look the same.
13. Modify your fragment shader to use the second texture. The texture will produce a value between 0 and 1, which is how darkened the fragment should be to produce the tiger stripes (0 meaning fully dark, and 1 meaning not darkened at all). This "darkening" is similar in spirit to shading. In fact, the correct way to apply it is just to treat it as a second shading value:  
`frag_color = frag_color * shading * tiger_stripe.x;`
  - a. **IMPORTANT:** the "texture" function does not have a good signature for our purposes. In particular "`float tiger_stripe = texture(...)`"; does not work. I did "`vec4 tiger_stripe = texture(...)`" and used `tiger_stripe.x` when adjusting the shading.
14. Compile and run. You should get this picture:



What to submit?: just one file – `project3A.cxx`.