

## Lab 8 PDF Submission

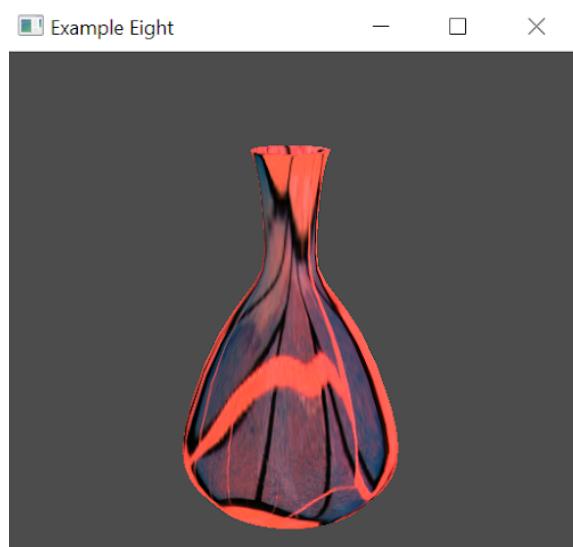
Name: Michael Loo  
Student ID: 100702210

Note: I set ambientLight to 0.3 because that's what the TA said was ok

### Exercise 1:

For exercise 1, I used example 8, then I used the code from example6c.fs to add specular highlight to the textured vase. The texture values were used as the ambient and diffuse components to give the lighting its look.

### Result:



### Code:

```

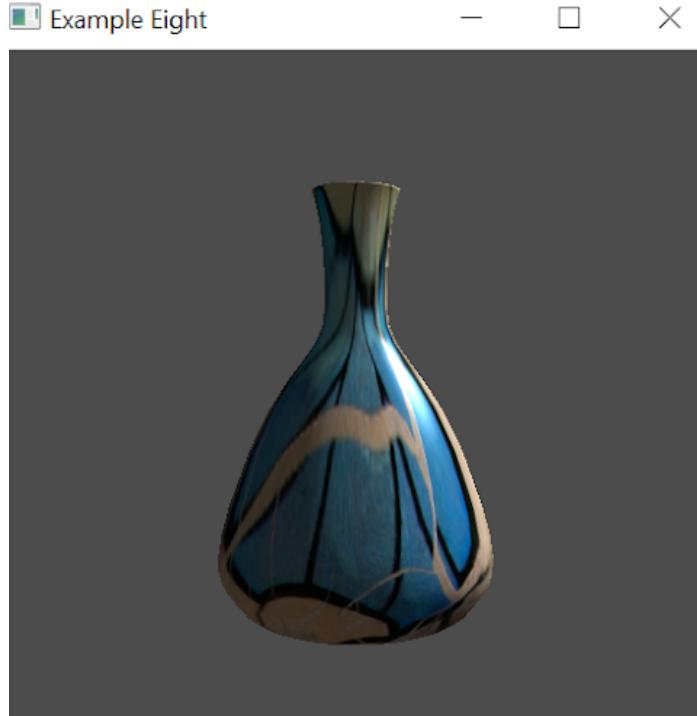
1 /* Simple fragment shader for the first
2  * texture example. Just look up the texture
3  * value at the current text coordinate
4  */
5
6 #version 330 core
7
8 in vec2 texCoord;
9 uniform sampler2D tex;
10
11 //for specular highlights
12 in vec3 normal;
13 in vec4 position;
14 uniform vec3 eye;
15
16 void main(void) {
17
18     //taken from example6c.fs
19     vec3 N;
20     vec3 L = vec3(1.0, 1.0, 0.0);
21     vec4 colour = vec4(1.0, 1.0, 1.0, 1.0);
22     vec4 Lcolour = vec4(1.0, 0.0, 0.0, 1.0);
23     vec3 H = normalize(L + vec3(0.0, 0.0, 1.0));
24     float diffuse;
25     float specular;
26     float n = 100.0;
27
28
29     N = normalize(normal);
30     L = normalize(L);
31     diffuse = dot(N,L);
32     if(diffuse < 0.0) {
33         diffuse = 0.0;
34         specular = 0.0;
35     } else {
36
37         specular = pow(max(0.0, dot(N,H)),n);
38
39         gl_FragColor = min(texture(tex, texCoord) * vec4(1.0, 0.0, 0.0, 1.0) +
40             (0.3*texture(tex, texCoord) + texture(tex, texCoord)*diffuse*Lcolour +
41                 0.3 * texture(tex, texCoord)), vec4(1.0));
42
43         gl_FragColor.a = colour.a;
44     }
45
46 }

```

## Exercise 2:

For exercise 1, I continued off of exercise 1. The changes that were made were that I used the value retrieved from the texture map as the object colour. I also set the color back to the original. The new result looks to have a lighting effect that is not shining evenly over the whole vase and not from all directions.

## Result:



## Code:

```
1  /*  
2  * Simple fragment shader for the first  
3  * texture example. Just look up the texture  
4  * value at the current texture coordinate  
5  */  
6  
7  #version 330 core  
8  
9  in vec2 texCoord;  
10 uniform sampler2D tex;  
11  
12 //for specular highlights  
13 in vec3 normal;  
14 in vec4 position;  
15 uniform vec3 eye;  
16  
17 void main(void) {  
18  
19    //taken from example6c.fs  
20    vec3 N;  
21    vec3 L = vec3(1.0, 1.0, 0.0);  
22    //vec4 colour = vec4(1.0, 1.0, 1.0, 1.0);  
23    vec4 Lcolour = vec4(1.0, 1.0, 1.0, 1.0);  
24    vec3 H = normalize(L + vec3(0.0, 0.0, 1.0));  
25    float diffuse;  
26    float specular;  
27    float n = 100.0;  
28  
29    N = normalize(normal);  
30    L = normalize(L);  
31    diffuse = dot(N,L);  
32    if(diffuse < 0.0) f  
33  
34    //vec4 colour = texture(tex, texCoord);  
35    //gl_FragColor = min(0.3*colour + diffuse*colour*Lcolour + Lcolour*specular, vec4(1.0));  
36    //gl_FragColor = min(texture(tex, texCoord) * vec4(1.0, 0.0, 0.0, 1.0) +  
37    //    ((0.3*texture(tex, texCoord) + texture(tex, texCoord)*diffuse*Lcolour +  
38    //    0.3 * texture(tex, texCoord)), vec4(1.0));  
39    gl_FragColor.a = colour.a;  
40  
41    }  
42  
43    N = normalize(normal);  
44    L = normalize(L);  
45    diffuse = dot(N,L);  
46    if(diffuse < 0.0) f  
47  
48    N = normalize(normal);  
49    L = normalize(L);  
50    diffuse = dot(N,L);  
51    if(diffuse < 0.0) f  
52
```

## Exercise 3:

For the first part of exercise 3, I multiplied the texture coordinates by 4 and saw that the textures were repeated more times within the space of the vase on the outside. Then I replaced GL\_REPEAT with GL\_CLAMP and saw that the textures were now mapped on the inside of the vase rather than the outside.

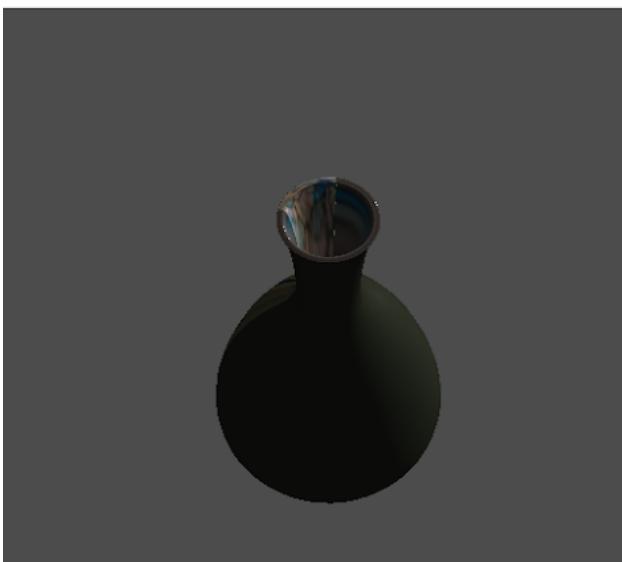
### Part 1 results:

Example Eight



### Part 2 results: (zoom in to see better)

Example Eight



## Code:

### example8.vs changes:

```
1  /*
2   * Simple vertex shader for the first
3   * texture example.
4   */
5
6 #version 330 core
7
8 in vec4 vPosition;
9 in vec3 vNormal;
10 in vec2 vTexCoord;
11
12 uniform mat4 modelView;
13 uniform mat4 projection;
14 uniform mat3 normalMat;
15
16 out vec2 texCoord;
17 out vec3 normal;
18
19 void main(void) {
20
21     gl_Position = projection * modelView * vPosition;
22     normal = normalMat * vNormal;
23     texCoord = vTexCoord * 4;
24
25 }
```

### Main.cpp changes

```
265
266     glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, texture->width, texture->height,
267                 0, GL_RGB, GL_UNSIGNED_BYTE, texture->data);
268     glGenerateMipmap(GL_TEXTURE_2D);
269     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
270     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
271     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
272 //     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
273     glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR_MIPMAP_LINEAR);
274
275 }
276
```