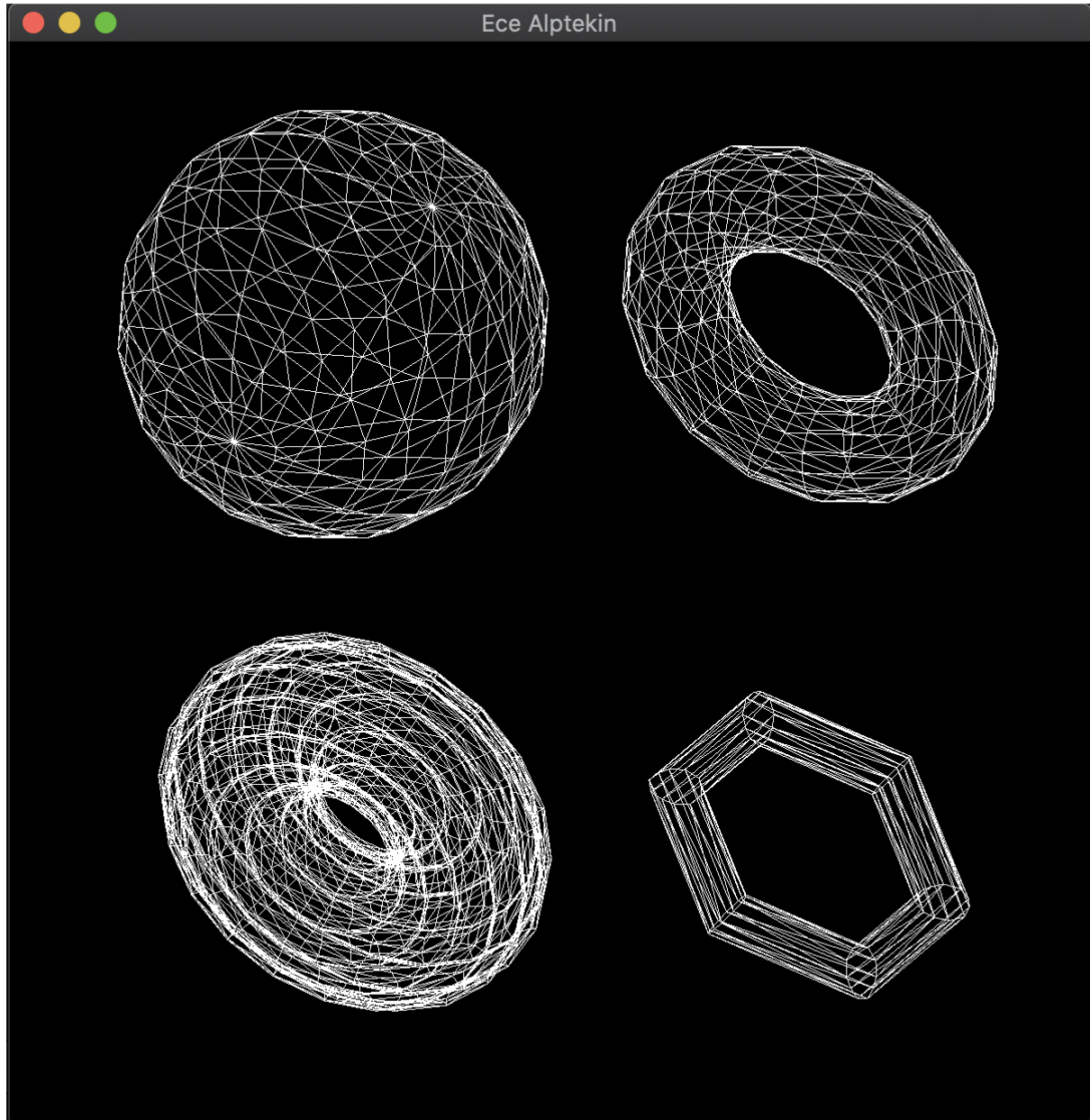


3D Project Part 1

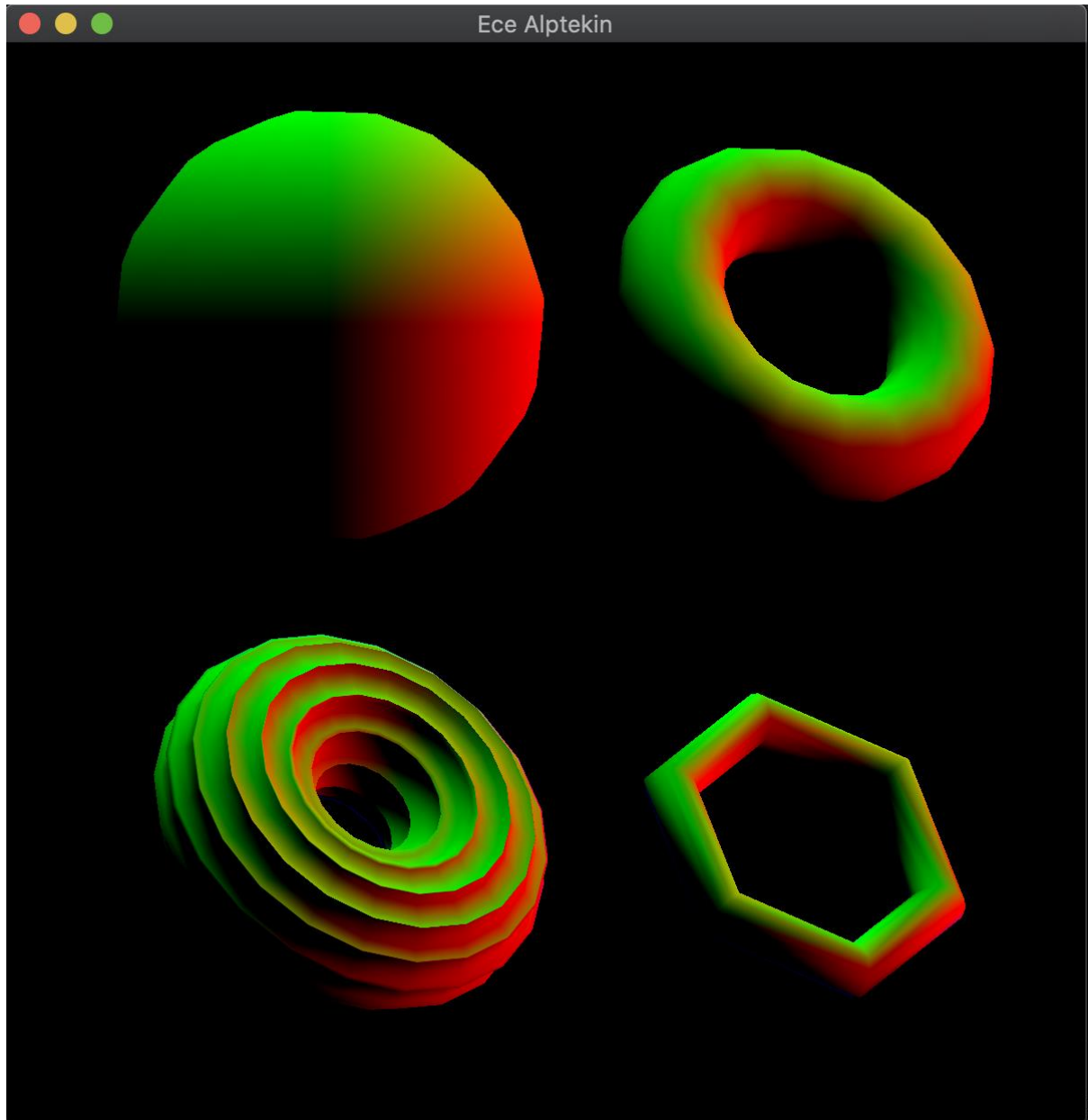
First Scene



The four meshes are located to different locations of the window with the transforms. The sphere and torus are created according to the instructions. The bottom meshes are created with `ParametricSpikes` and `ParametricSpikyCircle`. They are generated with 12×6 and 60×20 samples respectively. With the `glm:rotate`, all the meshes rotate around the axis $(1,1,0)$ and the rotation angle is $\text{current_time} \times 10$ degrees. Then, the meshes are rendered as wireframes

by utilizing the function `glPolygonMode(GL_FRONT_AND_BACK, GL_LINE)`; The color of the wireframes is set with `out_color = vec3(1, 1, 1)` in the shader implementation.

Second Scene

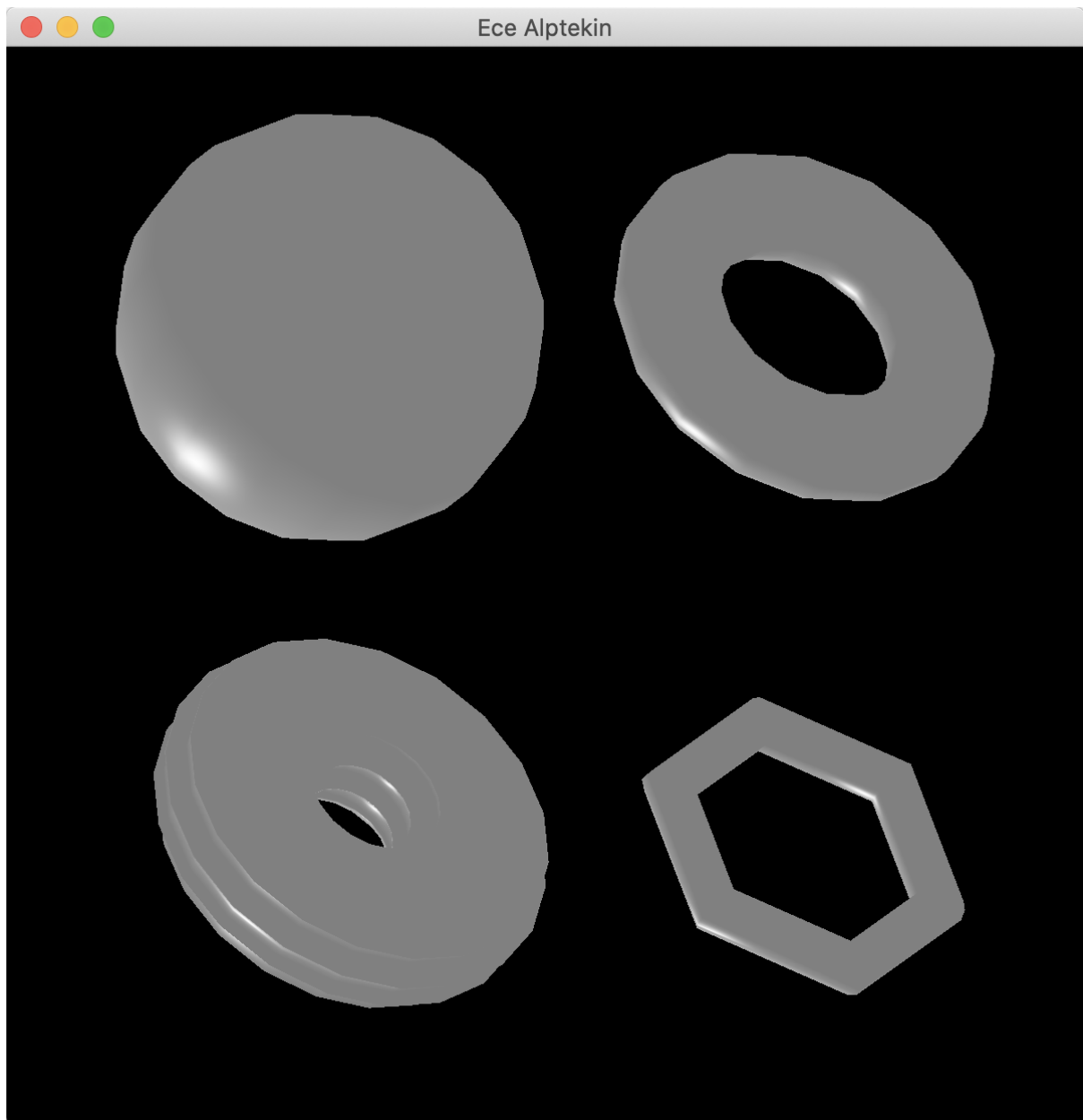


The four meshes are located to different locations of the window with the transforms. The sphere and torus are created according to the instructions. The bottom meshes are created with `ParametricSpikes` and `ParametricSpikyCircle`. They are generated with 12x6 and 60x20 samples respectively. With the `glm:rotate`, all the meshes rotate around the axis (1,1,0) and the rotation angle is `current_time*10` degrees. Then, the meshes are rendered by utilizing the function `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)`; To show the normal vectors as colors, the following implementation is made in the shader:

```
float ambient = 2;  
vec3 ambient_color = vec3(1);  
vec3 color = (ambient_color * ambient) * vertex_normal;
```

This implementation enables using the `vertex_normal`, which stores the normals, in the color.

Third Scene

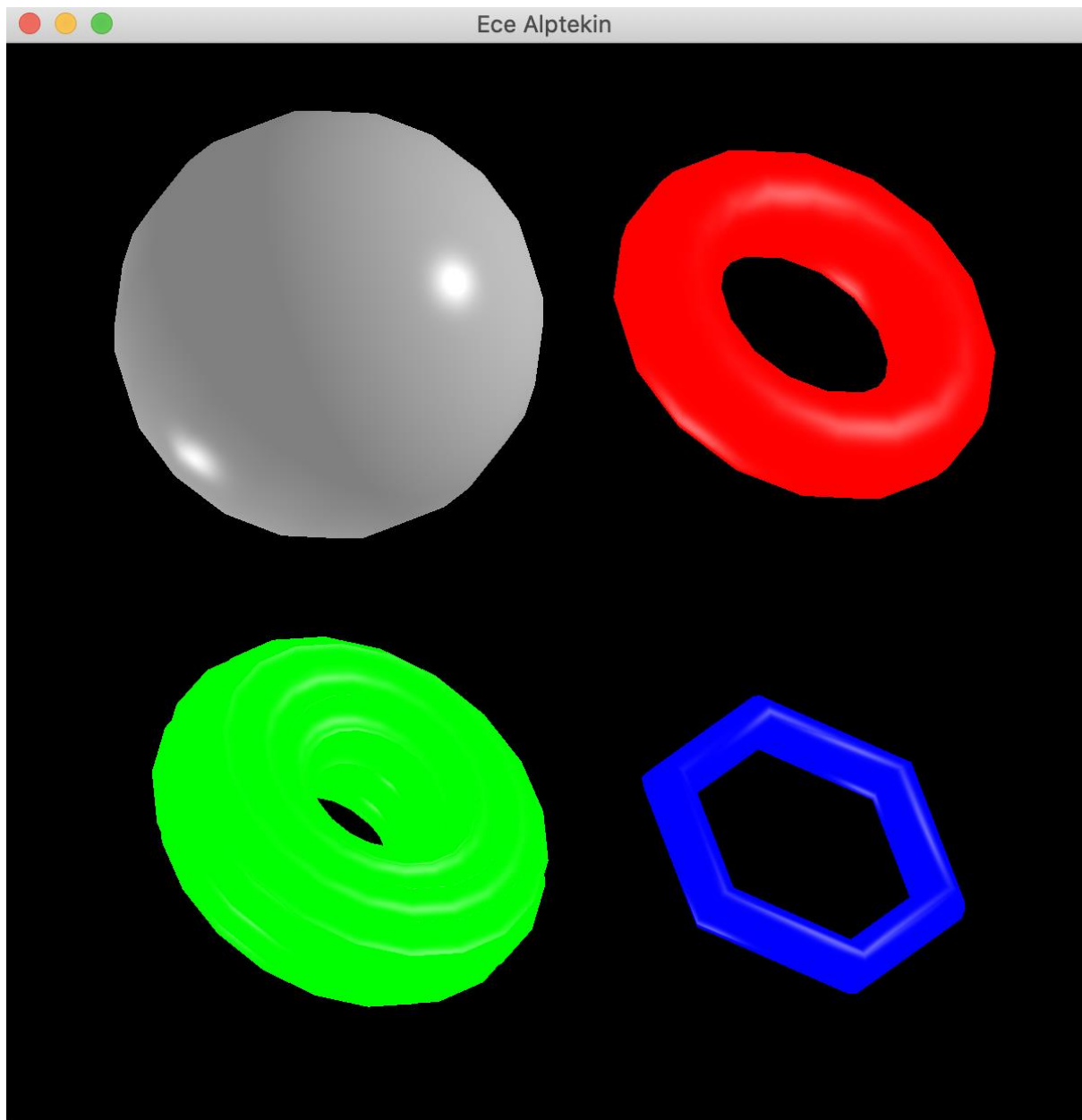


The four meshes are located to different locations of the window with the transforms. The sphere and torus are created according to the instructions. The bottom meshes are created with `ParametricSpikes` and `ParametricSpikyCircle`. They are generated with 12x6 and 60x20 samples respectively. With the `glm:rotate`, all the meshes rotate around the axis (1,1,0) and

the rotation angle is $\text{current_time} \times 10$ degrees. Then, the meshes are rendered by utilizing the function `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);`

The Blinn-Phong reflection model is implemented in the shader. A directional light as a light source is added to the scene. The color of the directional light is set to (0.4, 0.4, 0.4) and its direction is (-1, -1, 1). The surface color is gray for each mesh. The meshes have the same shininess value 64.

Fourth Scene

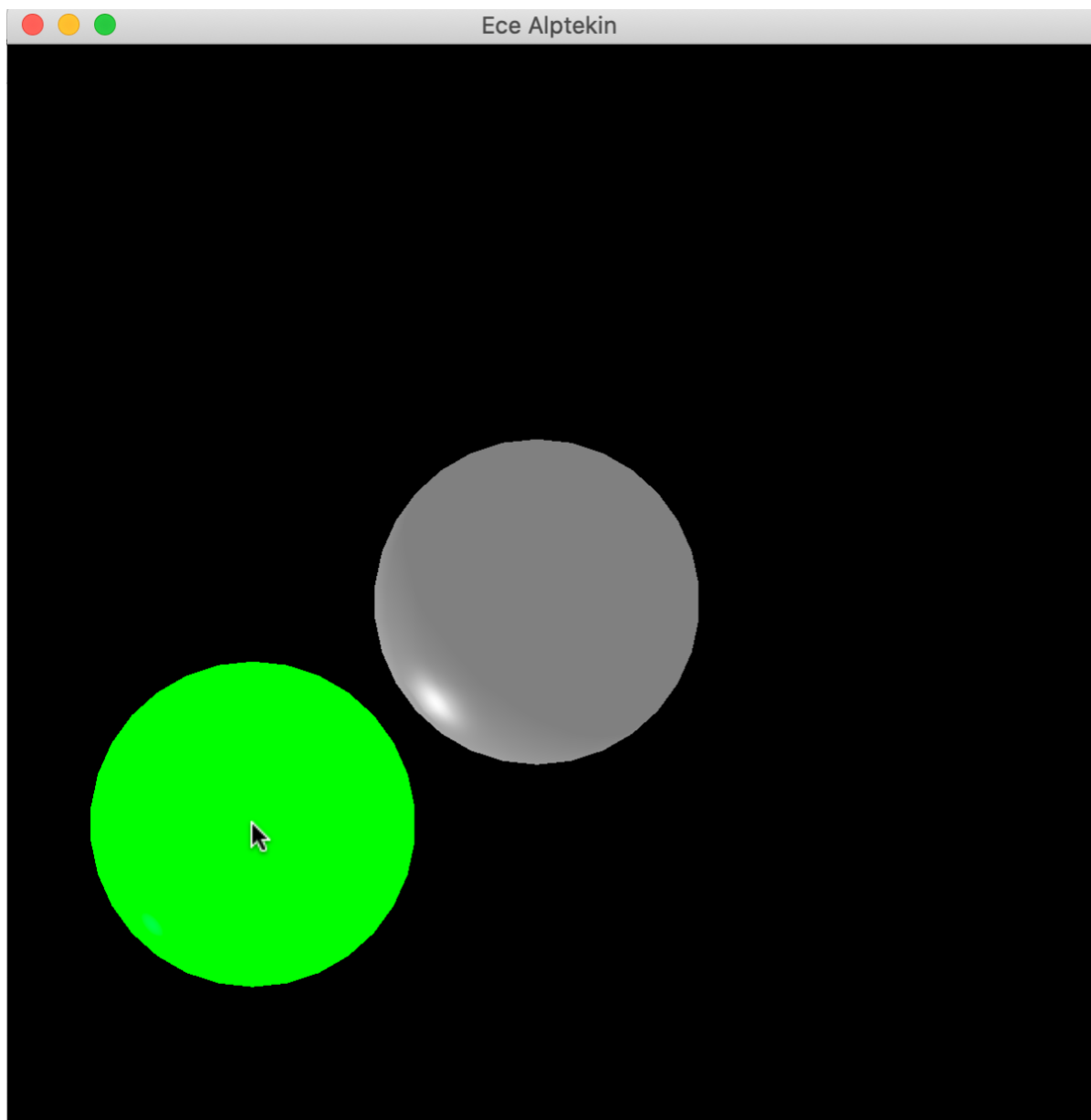


The four meshes are located to different locations of the window with the transforms. The sphere and torus are created according to the instructions. The bottom meshes are created

with ParametricSpikes and ParametricSpikyCircle. They are generated with 12x6 and 60x20 samples respectively. With the glm:rotate, all the meshes rotate around the axis (1,1,0) and the rotation angle is $\text{current_time} \times 10$ degrees. Then, the meshes are rendered by utilizing the function `glPolygonMode(GL_FRONT_AND_BACK, GL_FILL)`;

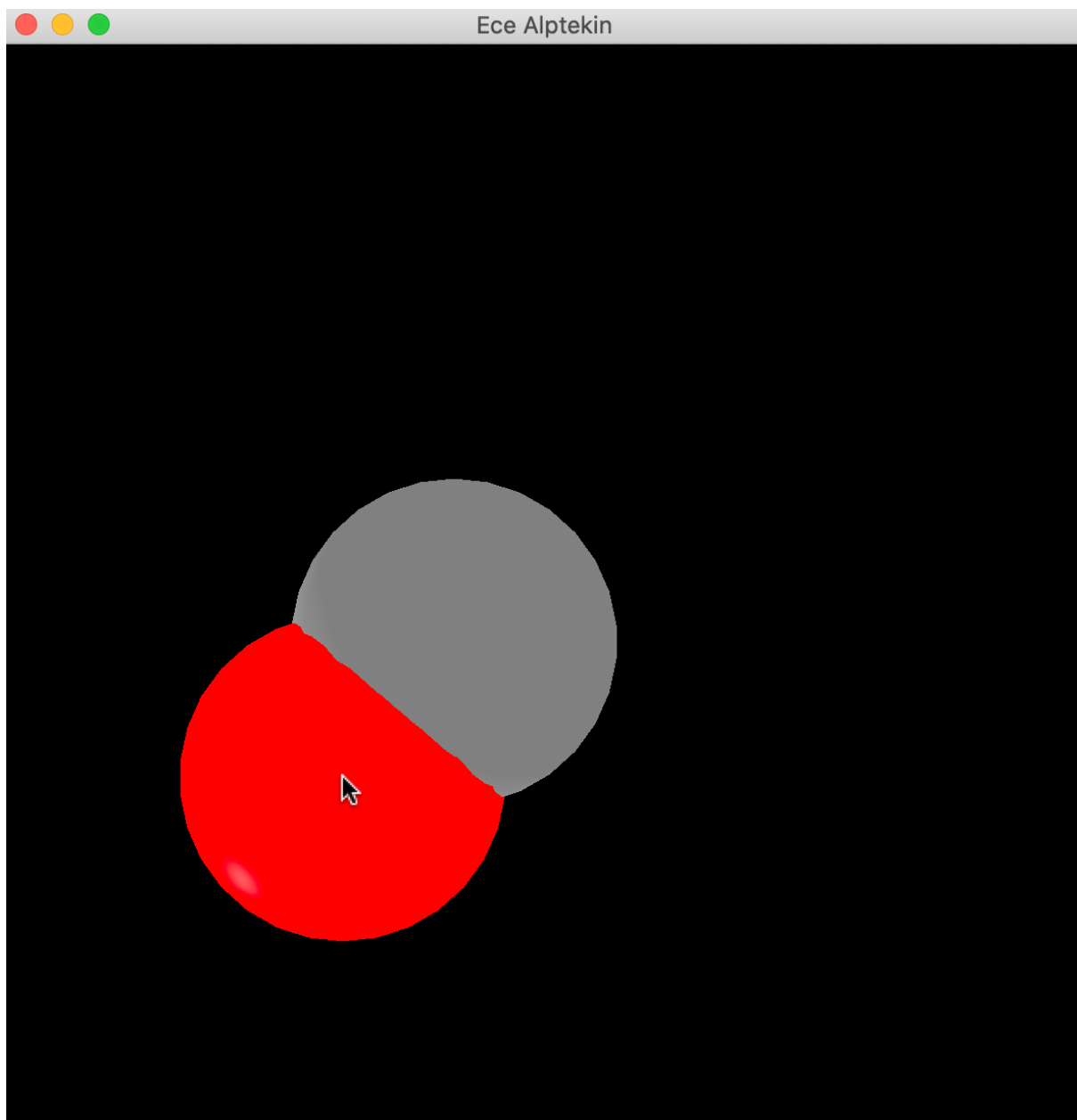
The Blinn-Phong reflection model is implemented in the shader. A directional light as a light source is added to the scene. The color of the directional light is set to (0.4, 0.4, 0.4) and its direction is (-1, -1, 1). A point light, which is controllable by mouse, is added to the scene, while the directional light from the third scene is used. The color of the point light is (0.5, 0.5, 0.5). The shininess and the surface color is set for each mesh. Sphere has the shininess 128 and torus has 32. The colors are the same with the demo.

Fifth Scene

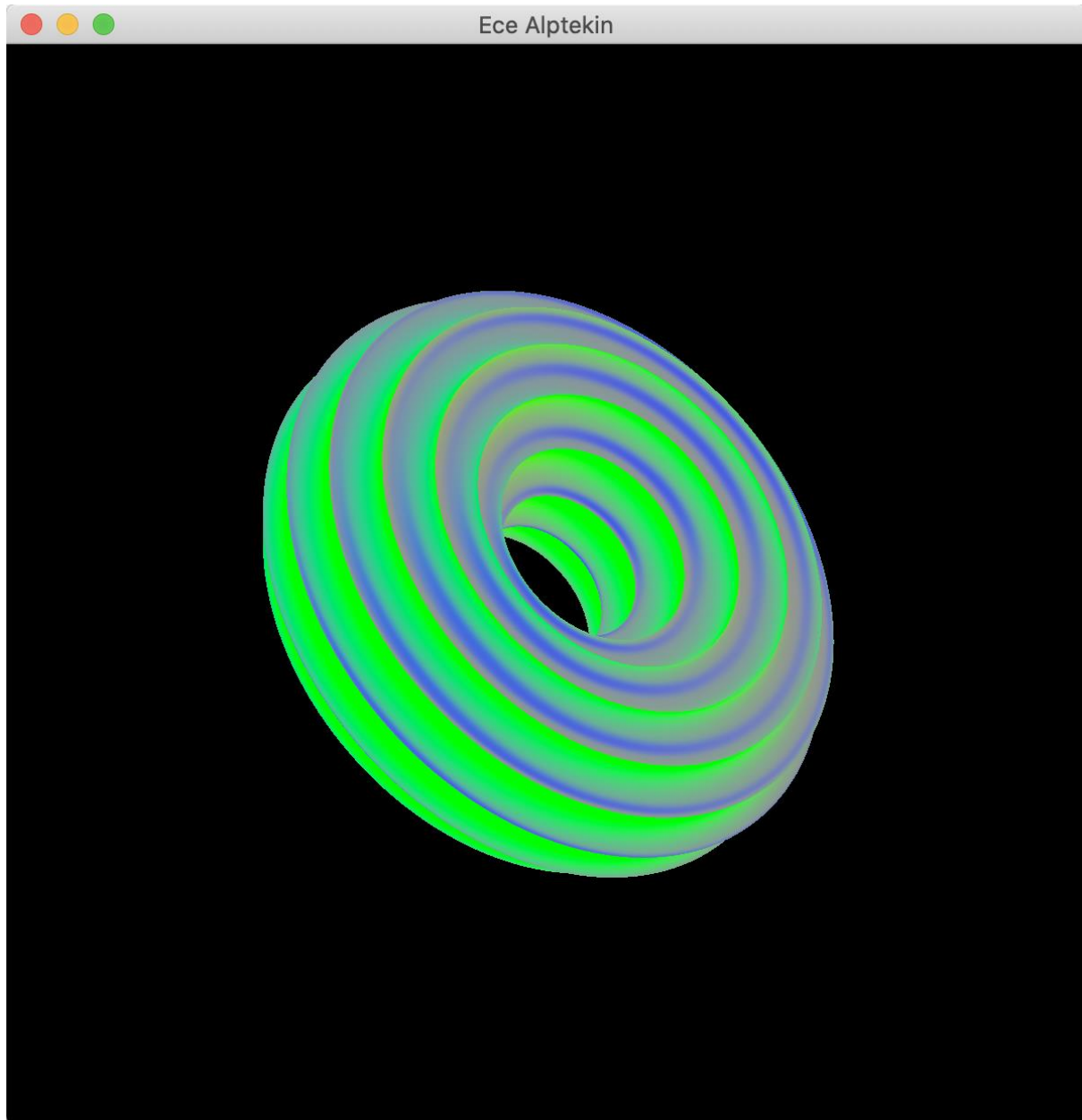


Two sphere, which are the same with the scene 1, are generated. One of the sphere is controlled by the mouse. The gray sphere is chasing the mouse-controlled sphere. A chasing_transform is generated for the chasing sphere. The translate of the chasing sphere is set with the function `glm::translate(chasing_pos);`

When the distance between two spheres bigger than $0.3 * 2$, the mouse-controlled sphere is green, otherwise it is red. The Blinn-Phong shading is used in the implementation. A directional light is added to the scene and set its direction to $(-1,-1,1)$. The light color is $(0.4, 0.4, 0.4)$. The shininess of the spheres is 64.



Sixth Scene



The mesh is drawn by ParametricSpikyCircle and scaled by 0.6. The rotation axis is $(1,1,0)$. The Blinn-Phong shading is used. A directional light and a point light are added. The shininess is set to 64. The directional light has the color $(0,0,1)$ and the point light has the color $(1,0,0)$. The ambient color is $(0,1,0)$ so that the mesh seems colorful. The mesh is generated with 1024×1024 samples. The mouse-controlled point light position is set with `vec3 point_light_position = vec3(u_mouse_position,-1);`