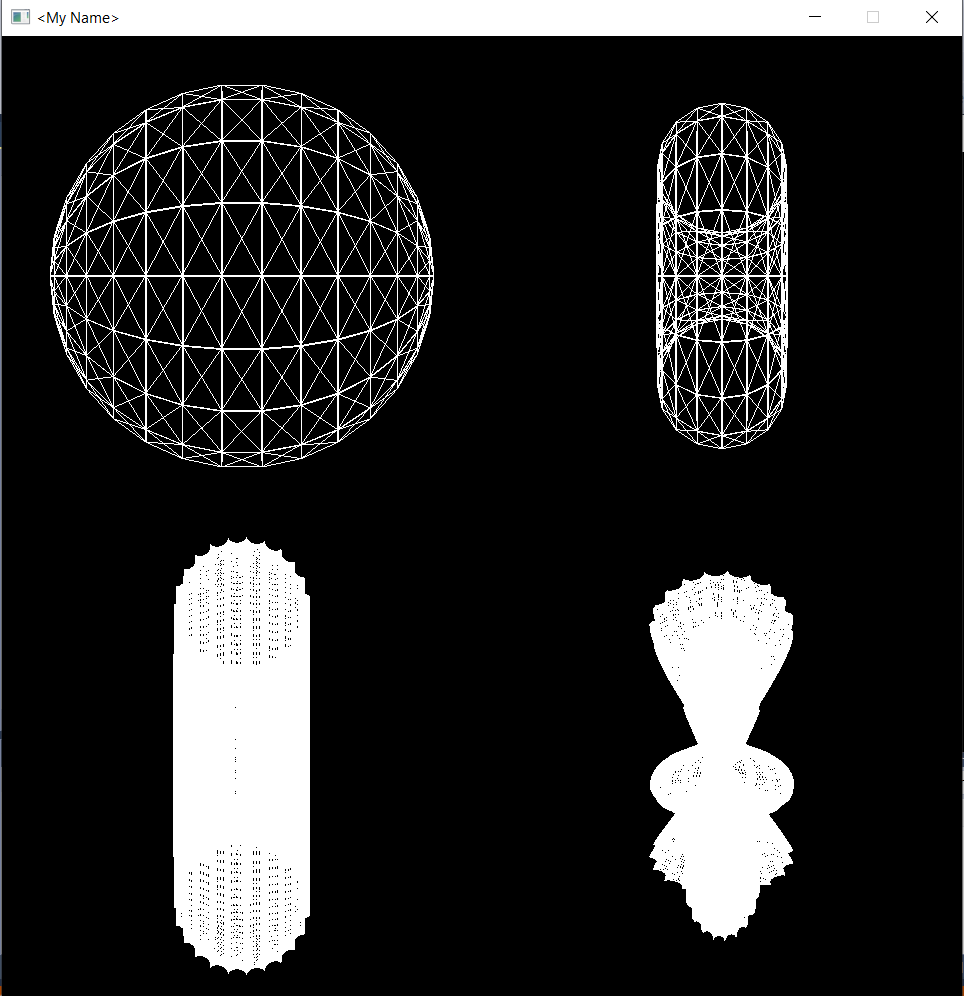
**3D Project Part 1 Report**

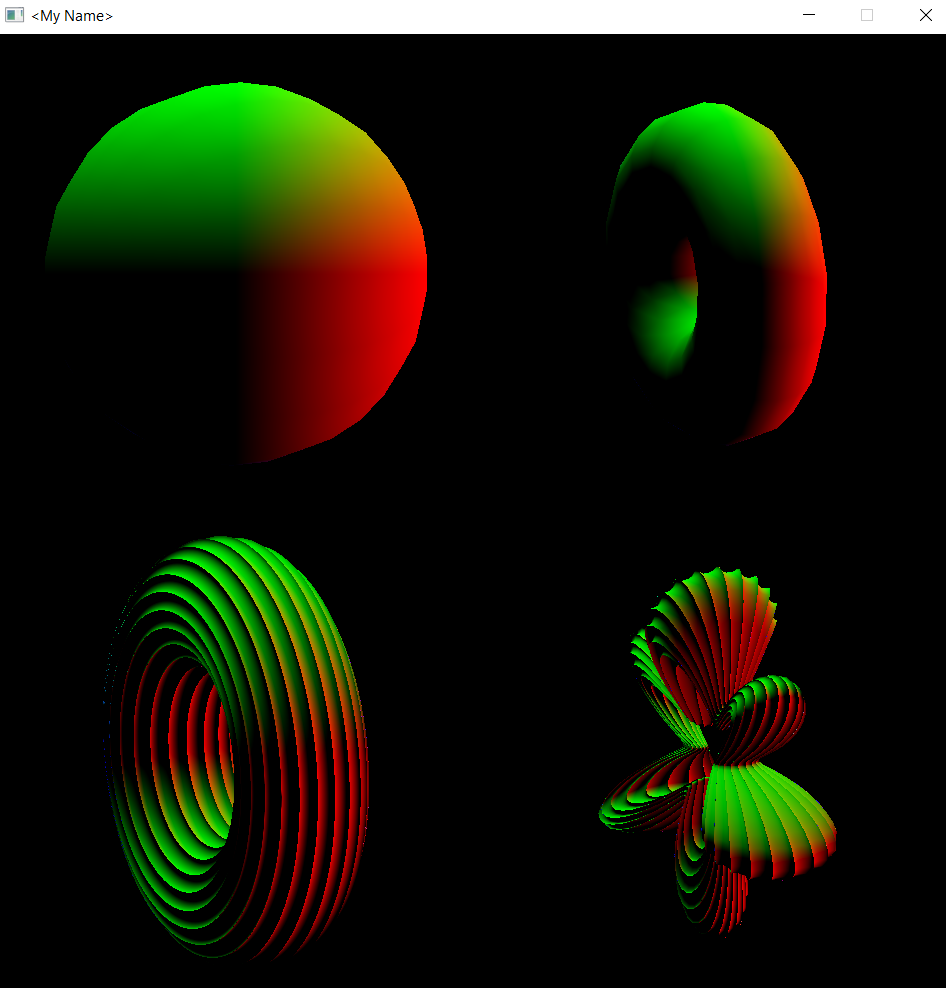
Nilay Irem Gucin 25464

*First Scene:*



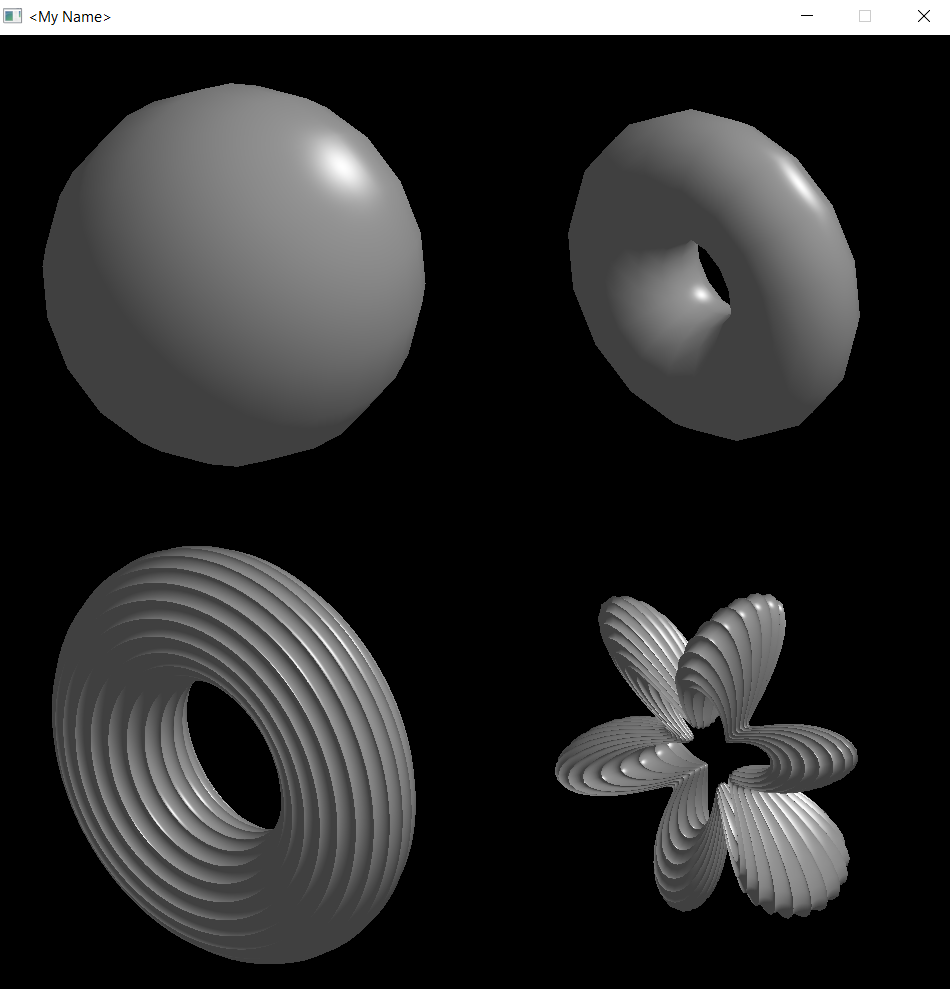
I used the GenerateParametricShapeFrom2D class for the first 3 shapes and used the GenerateParametricShapeFrom3D class for the last one. Upper left shape is done by ParametricHalfCircle function, upper right by ParametricCircle and lower left by ParametricSpikes function. I used the GenerateParametricShapeFrom3D like I used the parametric\_surface from GenerateParametricShapeFrom2D but instead of vec2 it receives vec3 as a parameter and makes the same computations as its counterpart. Only exra thins it does is that it multiplies p with sin of itself. I don’t know if it was the intended purpose of this extra class, but I did used it this way. For the wireframe look, I used the glPolygonMode as it was advised.

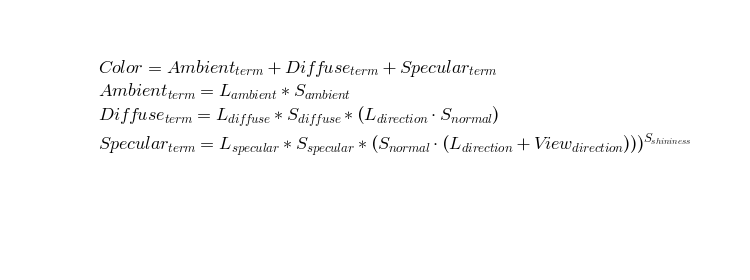
*Second Scene:*



The objects are the same for this scene. But the color component is added. The color of every point calculated as the shape’s normal normalized instead of white.

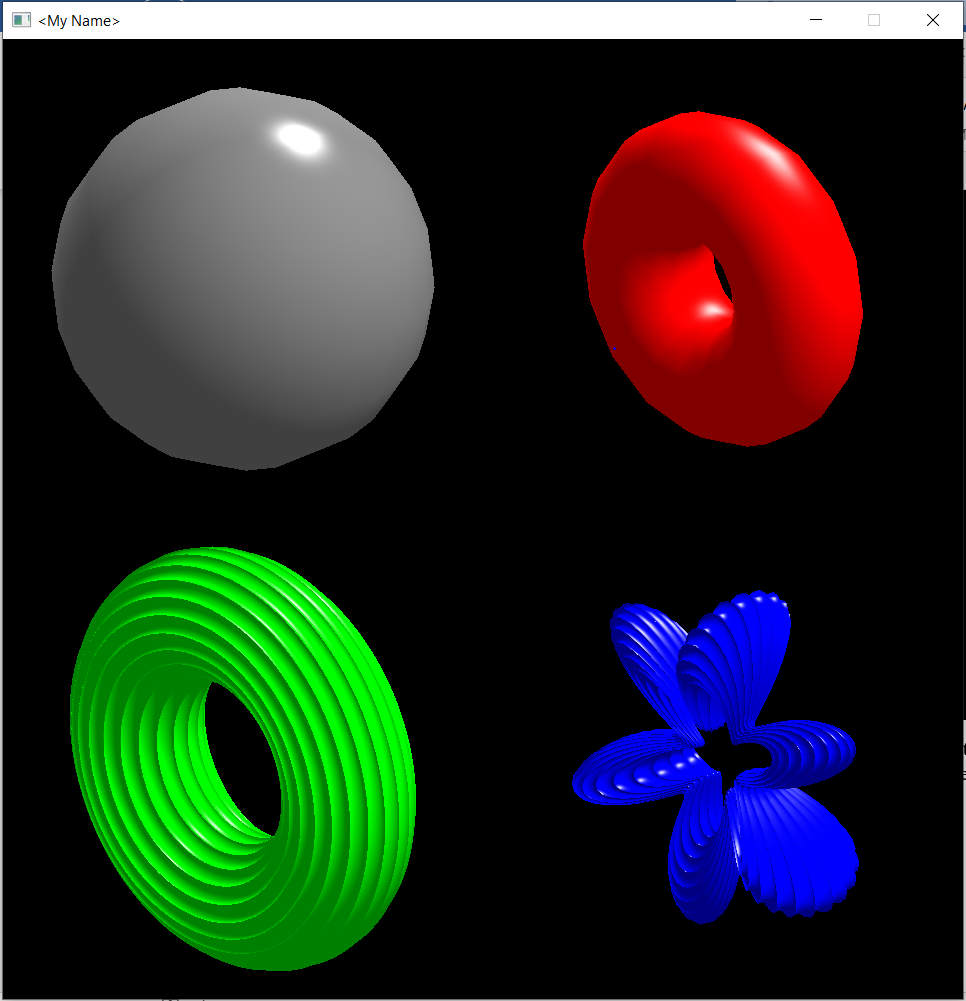
*Third Scene:*





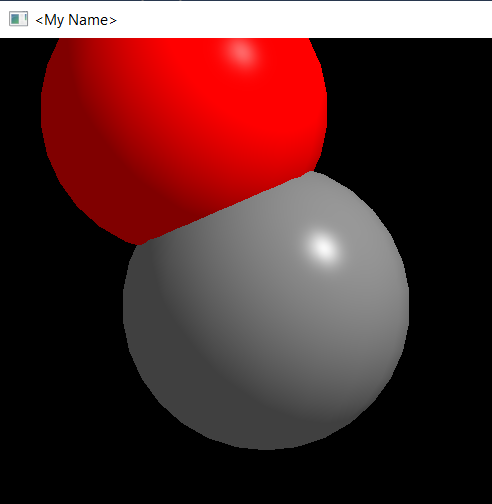
This scene has the same objects as well. But all the objects are grey. I initialized the surface\_color as grey and for it to be not too dark I initialized the ambient light as grey too. This way it looks more realistic. I used the formula in the picture. Diffuse light coefficient has calculated as the angle between the surface and the light which is the dot product. Specular light is about the viewer, so the specular coefficient is calculated with the angle between the camera, light and the surface. The shininess is about the material of the object and this case they are all made of the same material.

*Fourth Scene:*



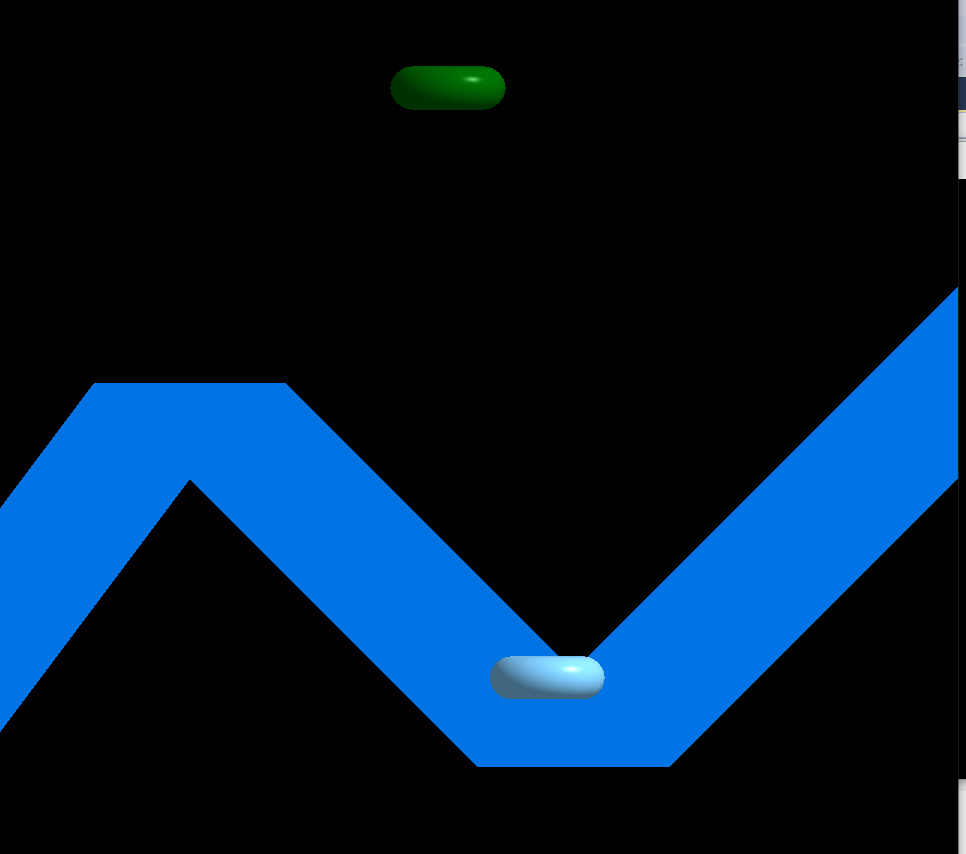
In this case, the color is not pre-determined in the program. I adjusted the colors in the main after the program starts. In terms of light, the calculation is the same, but there are 2 light sources so, the diffuse part changes. Specular also changes because there are 2 different angles. The sphere has shininess of 128 and the donut has the shininess of 32 therefore the sphere looks shinier and has a bigger light point. The shininess of the bottom two is 64.

*Fifth Scene:*



The objects in this scene are different. In this case there are 2 spheres. The light source is the same as the Third Scene. The position of the red ball is the same as the mouse\_position. The grey ball’s position is calculated as the chasing\_pos which chases the mouse but in a slightly slower manner. The distance is calculated with glm::distance() function and if it’s bigger than 0.3\*2 then the red ball becomes green.

*Sixth Scene:*





My goal in this was to create a game people can play with their mouse. It was inspired by the story of Orpheus and Eurydice. In this story Eurydice follows Orpheus in order to leave the underworld. But Orpheus should follow some rules or Eurydice, his love, dies. In this game the rule is not to leave the river. I created the river and the cross from triangles. The cross means you lost. I couldn’t use Bezier curves to check if the trajectory is correct. So I check if the distance between some points in the curve and the green ball’s (Orpheus) position is bigger than 2. If the aforementioned distance is bigger than 2 then the cross appears.