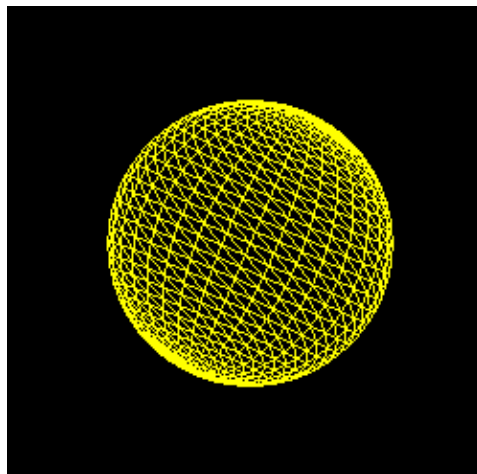


# Solar system

## Structure of the code

For this project, I decided to create 2 classes, a “Mesh” class according to the lab questions, which handled the creation of the sphere and variables such as the VAO and VBO, as well as a “Stellar” class which contained a shared pointer of a Mesh as well as many attributes such as the origin, the color, the light source, the rotation speed... This way, I was able to write `update()` and `render()` methods which worked for every Stellar body at once. To keep the number of files low, I decided to put both classes in “`mesh.h`” and “`mesh.cpp`”.

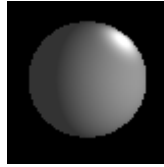
## Creating a sphere



The wire render of my sphere with resolution = 32.

The first major step of this project was to construct a sphere we could use for our stellar bodies. For this, I chose to use spheric coordinates and generated a sphere with every vertex at the same distance from the center, and by increasing  $\Theta$  and  $\phi$  with a constant step. I just needed to manually add the top and bottom vertices, and link them together with the EBO.

## Shading



An example of shading on a white sphere under white light.

After testing different possibilities, I decided to implement the Blinn-Phong shading, because it gave more satisfying results in my opinion, and a more realistic look. I tried different combinations for the coefficients in each component of the shading, and I think the result I came up with is good enough, even though there might be better options in some cases.

## Texture and movement



My sphere with the texture of Earth.

For the texture, since I had used constant variations of  $\Theta$  and  $\varphi$  to define my sphere, I decided that  $u = \varphi/2\pi$ ,  $v = \Theta/\pi$  would give a good result. In the end, I had to change the formula for  $u$  to  $1 - \varphi/2\pi$  to get the result I wanted.

Furthermore, since I needed to bind the textures around the z-axis, but I wanted an orbit in an horizontal plane, I rotated the whole scene around the y-axis, by putting a rotation matrix after (i.e left of) the translation matrix.

Finally, I decided to add a texture for the Sun, from Creative Commons.

## Attempts

In my project, you may find traces of attempts to go further into the project, such as an attempt of skybox, using a cubemap, which sadly did not give satisfying results in time.