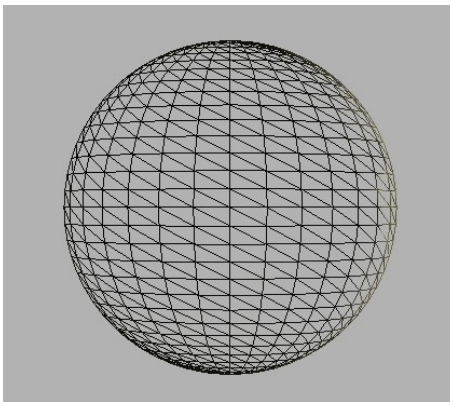
Yassine MANKAI

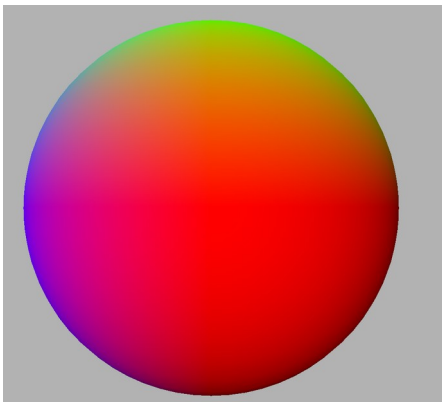
IGR201a Practical – OpenGL Porgramming

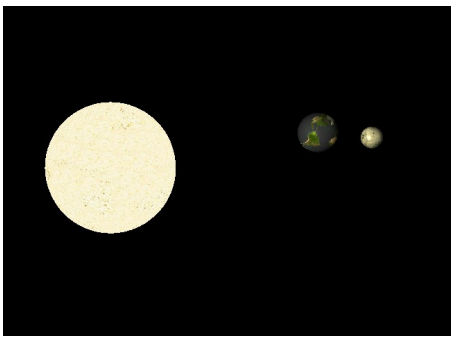
## Objectives successfully achieved:

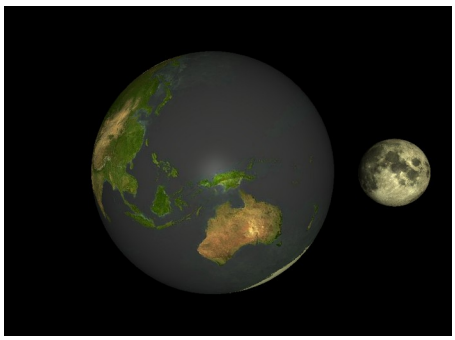
→ rendering a single triangle : using vertex buffer and index buffer.
→ coloring the triangle : adding a color buffer and updating the shaders
→ rendering correctly a sphere : transforming a plane mesh to a sphere mesh
→ creating the Mesh class
→ implementing the Phong shading model
→ implementing different model matrices to the same mesh to render multiple spheres
→ modifying the model matrices to animate the sphere (using glfwGetTime)
→ mapping a texture to the model instead of the color buffer
→ adding a camera mode feature and implementing a simple version of fly like camera

## Screenshots :









Yassine MANKAI

# Details :

- Determination of model Matrices :

There are two things to be taken into consideration when calculating model matrices:
→ The first one is the decomposition of the movement of each sphere into simple movements (translation, rotation and scaling). For example, the moon movement around the earth could be seen as a translation according to the x axis + simple rotation or a rotation + translation according to a rotating vector (sin(f*time),cos(f*time),0)
The second thing is the order in which we compose these transformations. The general way is to scale then rotate then translate.

- A note on the Sphere mesh :

A correct geometry could be achieved by connecting the vertices with phi= 2pi * (resolution -1)/ resolution directly with those having phi=0. This mesh have a drawback when mapping texture as you can't map two different sets of texture coordination to the same vertex. The solution adopted is to add a copy of the first column of the mesh at the end of the vertex array.

- Shaders :

For learning purposes, I decided to use two different shaders one for the lighting source and one for the other objects. I find this choice more flexible for future changes.

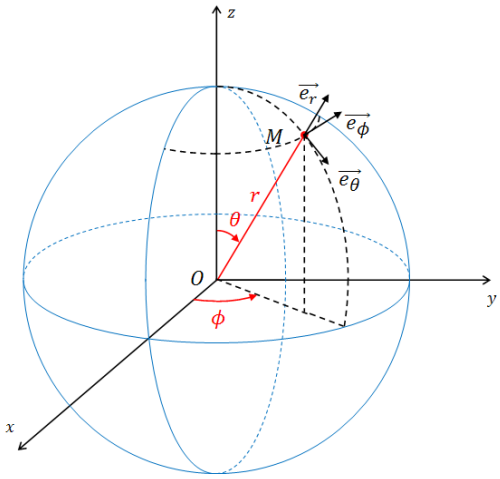
- Camera :

This app presents different camera modes determined by the camera position and the look at point . The user is free to fixate one of the sphere (center) and to examine it from the point of view of either one of the other spheres or from the outer space (free camera). These settings are configured via the set of commands presented below :

default : camera centered in the earth looking at the center of the moon

To change the camera position :
→ press 'C': outer space (free camera mode)
→ press 'V': earth's center
→ press 'B': moon's center
→ press 'N': sun's center

To change the look at point :
→ press 'J': earth's center
→ press 'K': moon's center
→ press 'L': sun's center




In the free camera mode : press (and maintain if wanted)
→ 'LEFT' or 'RIGHT': to decrease or increase (respectively) phi
→ 'UP' or 'DOWN': to decrease or increase (respectively) theta
→ 'Q' or 'S': to decrease or increase (respectively) r