Lab 04: 3D Object transformation - Texture mapping

1. Execution

- The source is implemented on Ubuntu 18 with OpenGl 3.0 Mesa 19.2.8
- To compile all the source code. Run `make all`
- The target execution file is `main`
- To run the program `./main`
- Github: https://github.com/nnmhuy/Lab04-OpenGL-3D-Object-Transformation-Texture-Mapping

2. Algorithms

Object drawing

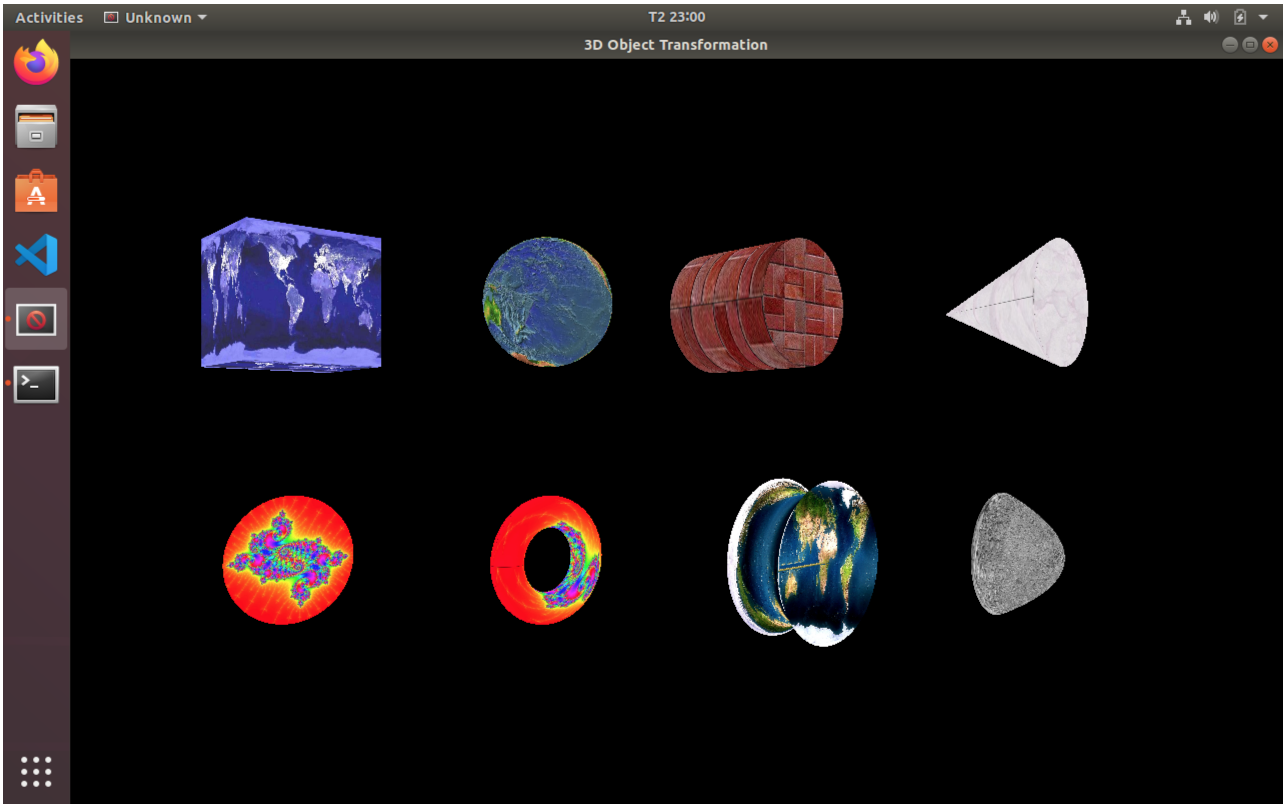
- The implementations allow drawing 8 different 3D objects which are:
  - Cube
  - Sphere
  - Cylinder
  - Cone
  - Disk
  - Torus
  - Hyperboloid
  - Paraboloid
- Each object is inherited from base class Object with attributes like texture index and rotation angle; main method is drawScreen
- Depending of object type there are some additional attributes like radius, height, nStack, nSlice, ...
- Each object base on each shape and equation is divided in to smaller primitives to combine into a large object
- I use two types of primitives in this lab which are GL_QUAD_STRIP and GL_TRIANGLE_FAN:
  - GL_QUAD_STRIP: for surfaces cylinder, cone shape, torus, ...
  - GL_TRIANGLE_FAN: for circle top/bottom base of shapes

Texture mapping

- First all texture are loaded and stored using SOIL library
- Then before drawing any object, bind the corresponding texture of that object
- For the correct texture mapping coordinate, for each point on the mesh, I set the appropriate normal vector of the surface at that point using `glNormal3d` and the corresponding texture coordinate by `glTexCoord2d`

3. Demo

- 8 different objects with different textures for each object


Activities
Unknown
T2 23:00
3D Object Transformation