HW1

P76091551 譚至斌

- OS: Ubuntu 20.04
- Environment Setup: `$ sudo apt install build-essential cmake libglfw3-dev libglfw3 libglm-dev #(I just follow the guild slides)`
- Project Directory:

```
$ tree ./hw1
./hw1/
├── build -> may not exist
├── .clang-format
├── cmake
├── CMakeLists.txt
├── .git
├── .gitignore
├── resources
├── run.sh -> this is a script for quick demonstration
├── src
└── thirdparty
```

- Build: `$ mkdir -p build && cd build && cmake .. && make -j`
- Executable file: `$ ./build/bin/Homework01`
- Input format: `./Homework01 [model name] [texture name] [vertex shader file name] [fragment shader file name]`
- Run:

```
$ cd build/bin && ./Homework01
resources/model/Utah_teapot_\(solid\)_texture.obj
resources/texture/uv.png Shader/BasicVertexShader.vs.glsl
Shader/BasicFragmentShader.fs.glsl
```

  - or just run ./run.sh

```
$ ./run.sh #(this script will build and run the program automatically)
```

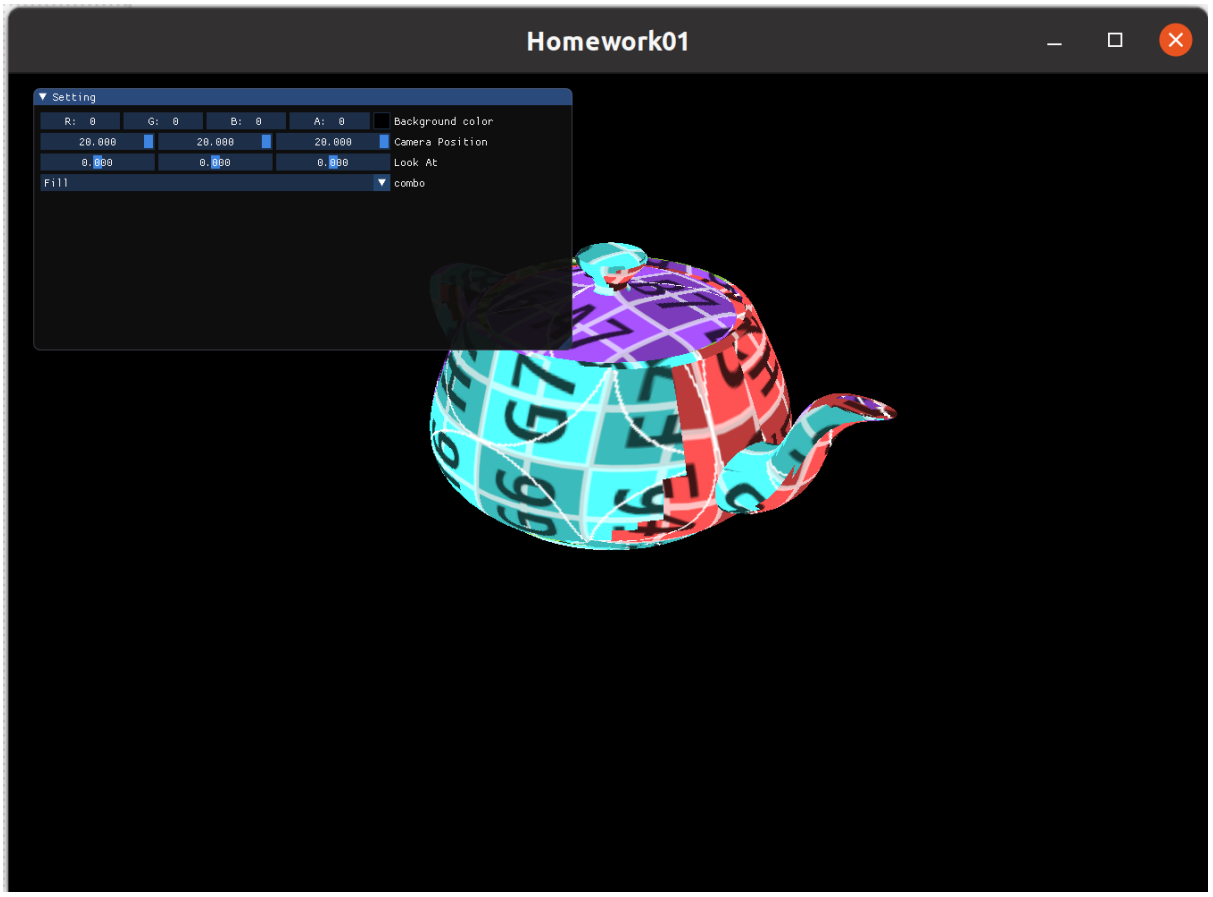
- Program Overview: I refer to the tutorials on learnopengl.com and modify the sample code by filling in the blank lines to complete this assignment.

```
src/
├── CMakeLists.txt
```

```
├── Main.cpp
├── Model
│   ├── Mesh.cpp
│   ├── Mesh.hpp
│   ├── TextureFactory.cpp
│   ├── TextureFactory.hpp
│   └── Vertex.hpp
├── OpenGL
│   ├── OpenGLBufferObject.cpp
│   ├── OpenGLBufferObject.hpp
│   ├── OpenGLException.cpp
│   ├── OpenGLException.hpp
│   ├── OpenGL.hpp
│   ├── OpenGLShader.cpp
│   ├── OpenGLShader.hpp
│   ├── OpenGLShaderProgram.cpp
│   ├── OpenGLShaderProgram.hpp
│   ├── OpenGLShaderProgram-inl.hpp
│   ├── OpenGLTexture.cpp
│   ├── OpenGLTexture.hpp
│   ├── OpenGLVertexArrayObject.cpp
│   └── OpenGLVertexArrayObject.hpp
├── OpenGLWindow.cpp
└── OpenGLWindow.hpp
```

- Main.cpp:
  - the entry point of the program which contains IO, window creation and rendering loop
- OpenGLBufferObject:
  - the class of the OpenGL buffer object, with two types, vertex buffer and index buffer
- OpenGLShader:
  - the class of the OpenGL shader, which contains vertex shaders and fragment shaders
- OpenGLShaderProgram:
  - the class of the OpenGL shader program, which stores shaders and manipulates life cycle of shaders
- OpenGLTexture:
  - the class of the OpenGL texture
- OpenGLVertexArrayObject:
  - the class of the OpenGL vertex array buffer

- Operation Manual



  - Backgroud color:
    - check right box to show background color
  - Camera position:
    - the coordination of the camera
  - Look At:
    - the direction which the camera face to
  - combo:
    - Line: only render mesh line
    - Fill: render the whole model including texture