Project 7 – Shadow Mapping

Name:	Mukunth Balaramachandran Srinivasan
uNID:	u1467270

What’s implemented?

All requirements implemented. Implemented simple shadow mapping for an object including independent control of the light position (using CTRL + mouse button to move and zoom in/out) and the display of light position. The shadows are calculated for both the object and the plane. Instead of pure black, an ambient colour is given to the shadows.

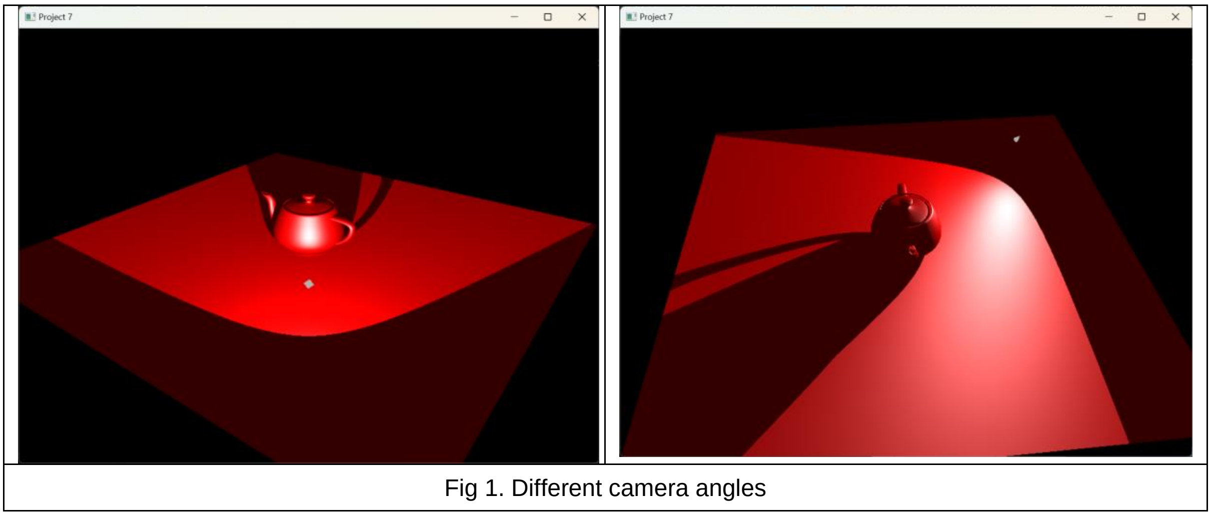

Fig 1. Different camera angles

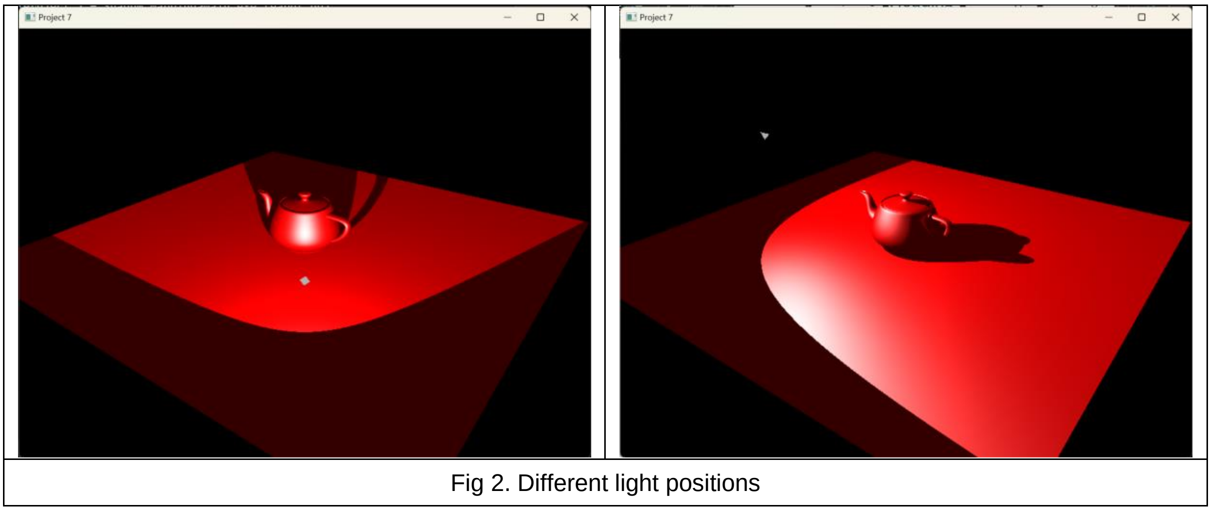

Fig 2. Different light positions

What could not be implemented?

The optional requirement:

- *Display the light as an object.*

Additional functionalities

Plane shadow:

The shadows cast by the plane are also calculated along with the shadows cast by the object.

Ambient shadow color:

Instead of pure black, an ambient color is given to the shadows of both the object and the plane.

Previous projects' functionalities:

- Left mouse button to rotate and right mouse button to zoom in/out (click and drag).
- Re-compiling shaders on pressing F6 key.
- Quitting the program on pressing Esc key.

How to use implementation?

```
g++ main.cpp -o main -lfreeglut -lglu32 -lopengl32 -lglew32
```

This command will generate the output file "main" ("main.exe" in Windows) in the working directory. This command includes the GLEW 32-bit linker. I didn't use an IDE and had all the libraries and headers globally installed, so I didn't have to use `-I` and `-L` tags to specify paths to headers and DLLs.

The folder structure for the headers in `include` is as follows:

```
-> include
      -> GL / all FreeGLUT and GLEW headers
      -> cyCodeBase / all cyCodeBase headers
```

OS and Compiler

Operating System	Windows 11 (x64)
Compiler	g++

External libraries and additional requirements

Apart from FreeGLUT, GLEW and cyCodeBase have been used for this implementation.