CS3241 Computer Graphics (2022/2023 Semester 1)

Lab Assignment 3

Release Date: 14 October 2022, Friday

Submission Deadline: 30 October 2022, Sunday, 11:59 PM

LEARNING OBJECTIVES

Writing OpenGL program to simulate planar reflection using texture-mapping and a multi-pass rendering technique. After completing the programming assignment, you should have learned how to

- set up texture mapping in OpenGL,
- model and draw texture-mapped objects,
- set up off-center view frustum,
- read back image in framebuffer for texture mapping, and
- simulate planar reflection using a multi-pass rendering technique.

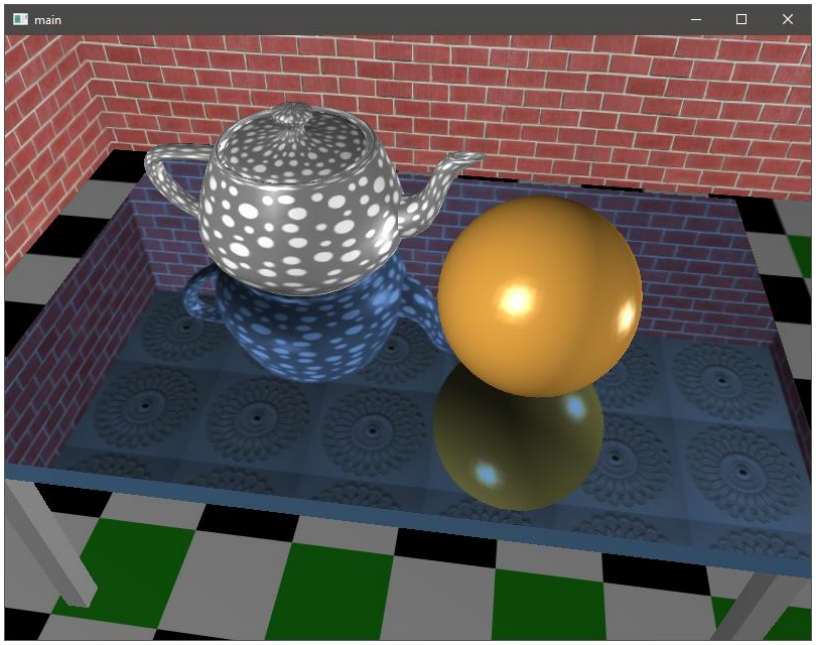
TASKS

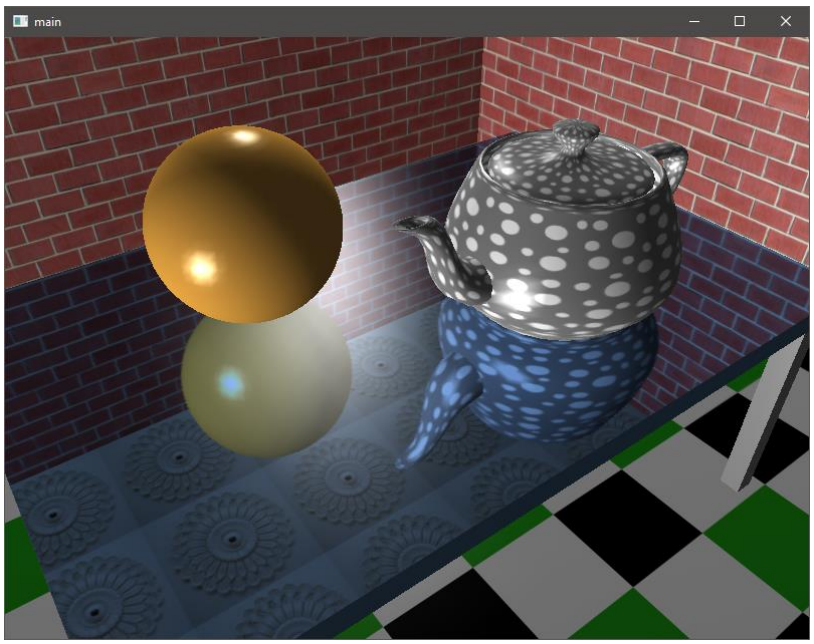
Please download the ZIP file **`Lab3_todo_(*).zip`** from the **Assignments** folder in LumiNUS Files.

You are to complete an **incomplete** C++ application program, so that it can simulate **planar reflection**, using a multi-pass rendering technique and the texture-mapping capabilities provided by OpenGL. You have to complete the program according to the following requirements.

Task 1

You have to complete **`main.cpp`** to produce the **planar reflection** that you see in the following sample images:





You may try the **completed executables:** **`main_done.exe`** (for Windows), and **`main_done`** (for macOS) found in the same ZIP file. (On macOS, you may need to use the command **`sudo chmod +x main_done`** to give the file execute permission before running it.)

Please read the instructions shown in the console/terminal window to learn how to operate the program.

The 3D scene contains a table with a **flat rectangular semi-reflective table-top**. The scene is also populated with other objects, at least some resting on the table-top. The table-top must **reflect the scene**. Here are some additional requirements:

- The reflection on the table-top is created by texture mapping a reflection image onto the table-top rectangle. The reflection image is generated by drawing the scene seen from an **imaginary viewpoint**, which looks through the table-top from under the table. This rendered image is then copied from the color buffer to a texture object, to be used for texture mapping the table-top rectangle.

- The reflection on the table-top should not be 100% (it is not a perfect mirror), and the underlying diffuse color and lighting on the table-top must still be visible. (Hint: use the correct texture function/environment.)

- **Mipmapping** must be used for all texture mapping, including for the reflection texture mapping. For the texture object that contains the texture image copied from the color buffer, you have to set the texture object using

```
glTexParameteri(GL_TEXTURE_2D, GL_GENERATE_MIPMAP, GL_TRUE);
```

- You are **not allowed** to use the **stencil buffer** for this assignment.

- Write your code immediately below the locations marked "**`WRITE YOUR CODE HERE`**". There are **three** of such locations.

- You are allowed to modify only **`main.cpp`**. You are not required and must not change any other source files.

Task 2

In addition to the given teapot and sphere, you have to **add at least one new texture-mapped object** into the scene. The new object(s) should be positioned above the table-top, and can float in the air. You may not use the 3D models in the GLUT library (e.g. `glutSolidTorus()`) for your new object(s), and must write your own function(s) to provide the polygons, vertex normal, texture coordinates, and material. Each new object must have at least 4 polygons.

You are allowed to modify only **`main.cpp`**. You should use your own new image(s) to texture-map your new object(s). As before, **mipmapping** must be used for all texture mapping.

Besides your completed **`main.cpp`**, you also need to submit the **new texture image(s)**.

This task will be assessed based on the fulfillment of the basic requirements, on the technical difficulty and object's complexity, and on the aesthetics and creativity.

DO NOT HARD-CODE VALUES. You should write your code in such a way that when the values of the named constants (defined in the beginning of the program) are changed to other valid values, your program should function accordingly. For example, if the table's height is changed, the reflection from the table-top should still look correct.

A Visual Studio 2017 solution **`main.sln`** (or Xcode project **`main.xcodeproj`** on macOS) is provided for you to build the executable program. In this assignment, **you are not required and must not change any other C/C++ source files** besides **`main.cpp`**.

Besides GLUT (or FreeGLUT) and what is provided in the ZIP file, you should not use any other third-party libraries. Your code must compile with either the MSVC++ 2017 (or newer) compiler on Windows, or Clang on macOS.

GRADING

The maximum marks for this programming assignment is **100**, and it constitutes **8%** of your total marks for the module. The marks are allocated as follows:

- **Task 1 — 70 marks**

- **Task 2 — 30 marks**
  - 20 marks — basic requirements,
  - 5 marks — technical difficulty and object's complexity,
  - 5 marks — aesthetics and creativity.

Note that marks will be deducted for bad coding style. If your program cannot be compiled and linked, you get 0 (zero) mark.

Good coding style. Comment your code adequately, use meaningful names for functions and variables, and indent your code properly. You must fill in your **name**, and **NUS User ID** in the **header comment**.

SUBMISSION

For this assignment, you need to **submit only**

- Your completed **`main.cpp`** that contains code for both **Task 1** and **Task 2**;
- File(s) of your **new texture image(s)** for **Task 2**. They must be in the **`images`** subfolder. Total **image files' size** must not exceed **5 MB**.

You must put it/them in a ZIP file and name your ZIP file ***nus-user-id*`_lab3.zip`**. For example, if your NUS User ID is **`e0123456`**, you should name your file **`e0123456_lab3.zip`**.

Note that you may be penalized for submitting non-required files.

Submit your ZIP file to the **Lab 3 Submissions / Group T0*x*** folder in LumiNUS Files, where **T0*x*** is your officially allocated Tutorial group number. Before the submission deadline, you may upload your ZIP file as many times as you want to the correct folder. **We will take only your latest submission.** Once you have uploaded a new version to the folder, you **must delete the old versions**.

DEADLINE

Late submissions will NOT be accepted. The submission folder will automatically close at the deadline.

——— End of Document ———