SWS3005: Real-Time Graphics Rendering (2024)

Assignment #3 (Individual Work)

Release Date: 11 July 2024, Thursday

Submission Deadline: 16 July 2024, Tuesday, 11:59 PM

TASKS

You are to complete an OpenGL program to render a scene as if it is lit by an **image projected from a light projector**. The following images show sample views of the result that your program is expected to produce:

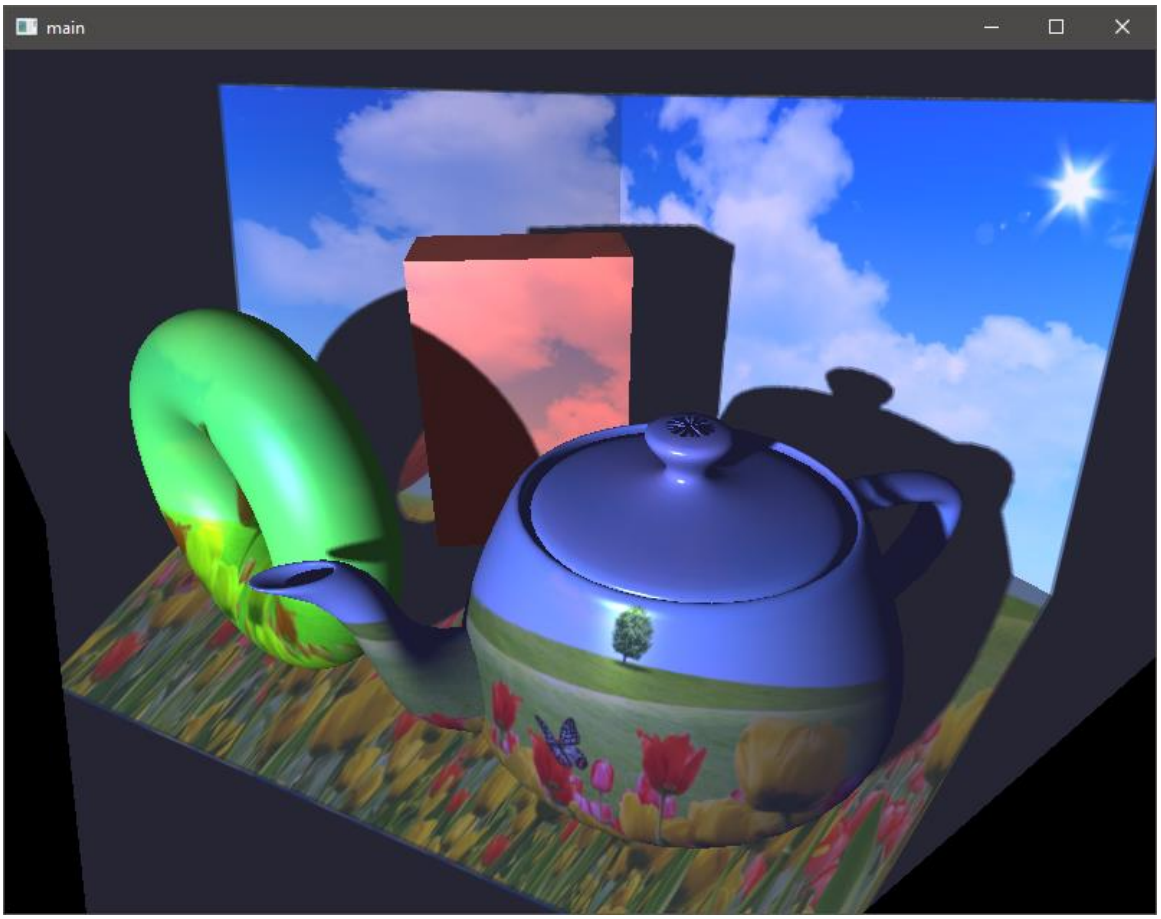

Figure 1

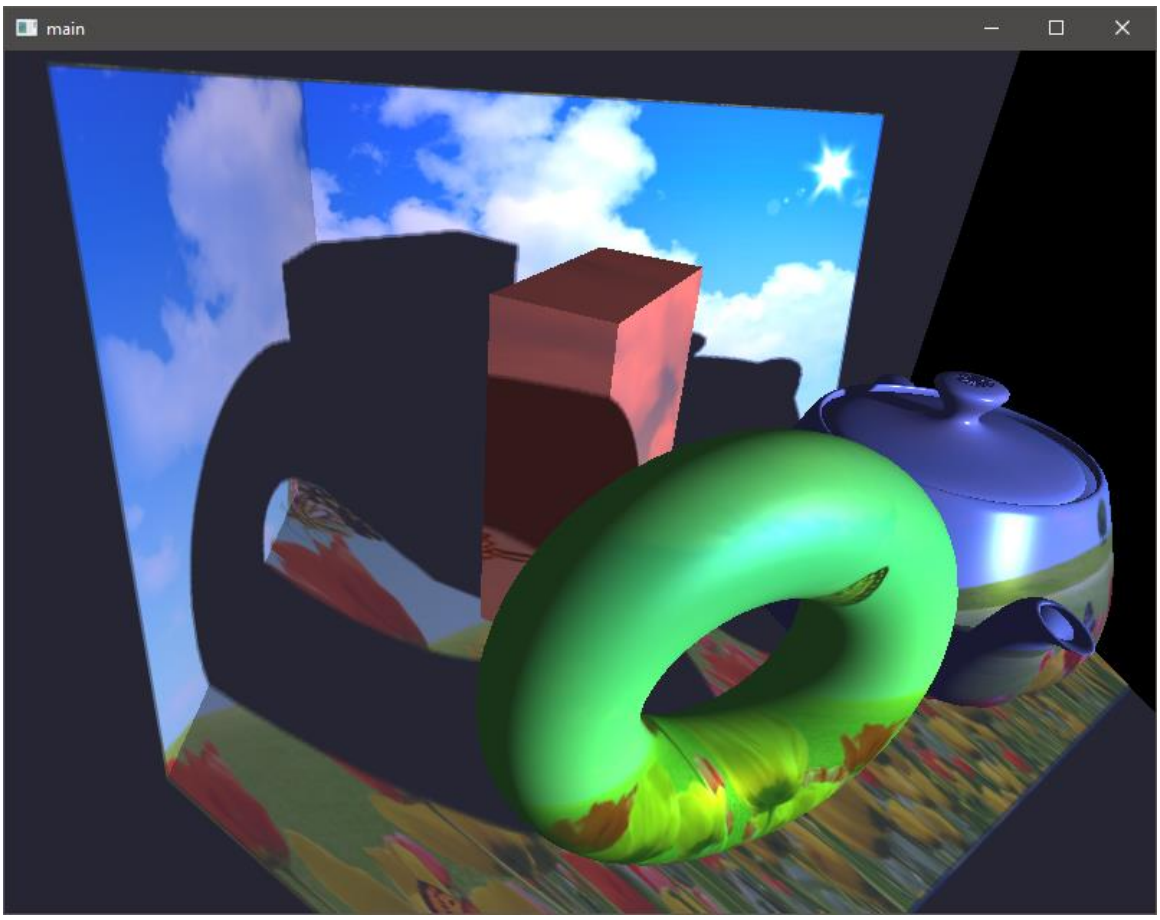

Figure 2

The shadow must be produced using the **shadow mapping** technique described in Lecture Topic B06. **Percentage-closer filtering (PCF)**, as described in Lecture Topic B06, must also be applied to smooth the shadow boundaries.

Please download the ZIP file **`sws3005_2024_assign3_todo_(*).zip`** from the **`Canvas > SWS3005 > Files > Assignments`** folder.

You need to complete the C++ application program **`main.cpp`** and the fragment shader **`shader.frag`**. In the fragment shader, all necessary **uniform variables**, and **global input/output variables** have already been declared, and **you must not add new ones**. You can add new functions in your shader. Note that you should adhere to the **variable naming convention** where the prefix "**`ec`**" is used to indicate that the entity is expressed in the eye space, the prefix "**`wc`**" to indicate world space, and the prefix "**`tan`**" to indicate tangent space.

A Visual Studio 2017 solution **`main.sln`** (or Xcode project **`main.xcodeproj`** on macOS) is provided for you to build the executable program. The application program loads the shader source files **`shader.vert`** and **`shader.frag`**, and use them in the rendering. It also provides the values for the **vertex attributes** and **uniform** variables to the shaders. In this assignment, **you are not required and must not change any other C/C++ source files** besides **`main.cpp`**.

There are **three tasks** in this assignment:

- **Task 1:** Complete the function **`DrawSceneWithProjection()`** in **`shader.frag`**.

  You can use the finished application program **`main_done.exe`** (or **`main_done`** on macOS) to test your fragment shader. The program does not produce correct rendering right now since it is using the incomplete fragment shader.

- **Task 2:** Complete the function **`SetUpShadowMapAndFBO()`** in **`main.cpp`**.

- **Task 3:** Complete the function **`RenderShadowMap()`** in **`main.cpp`**.

The detailed requirements for each task can be found in the source code.

GRADING

The maximum marks for this programming assignment is **100**, and it constitutes **20%** of your total marks for the course. The marks are allocated as follows:

- **Task 1 — 50 marks,**
- **Task 2 — 30 marks,**
- **Task 3 — 20 marks.**

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 (adhere to the new variable naming convention), 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 s**`hader.frag`** that contains code for **Task 1**;
- Your completed **`main.cpp`** that contains code for **Task 2** and **Task 3**.

You must put it/them in a ZIP file and name your ZIP file ***nus-user-id*`_A3.zip`**. For example, if your NUS User ID is **`t0912345`**, you should name your file **`t0912345_A3.zip`**.

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

Submit your ZIP file to **`Canvas > SWS3005 > Assignments > Assignment #3`**. Before the submission deadline, you may upload your ZIP file as many times as you want. **We will take only your latest submission**.

DEADLINE

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

——— End of Document ———