SWS3005: Real-Time Graphics Rendering (2022)

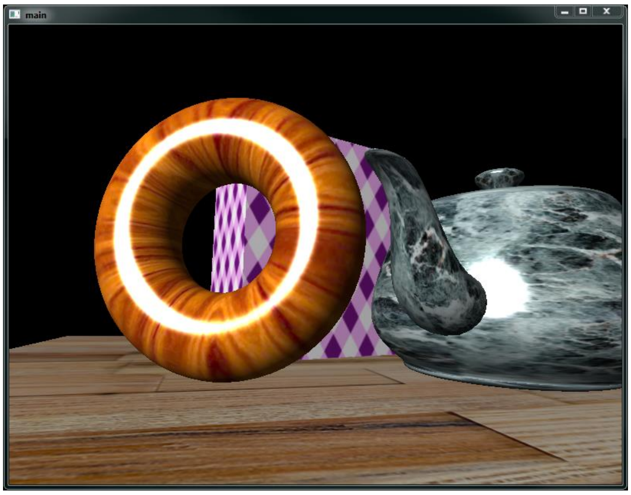
Assignment #4   (Individual Work)

Release Date: 20 July 2022, Wednesday

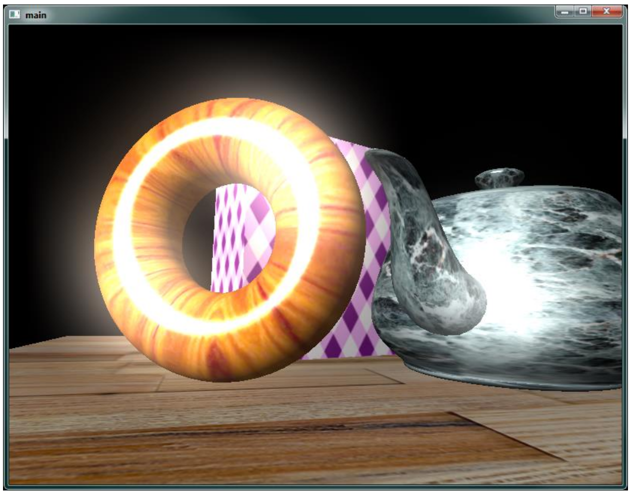
Submission Deadline: 25 July 2022, Monday, 11:59 PM

TASKS

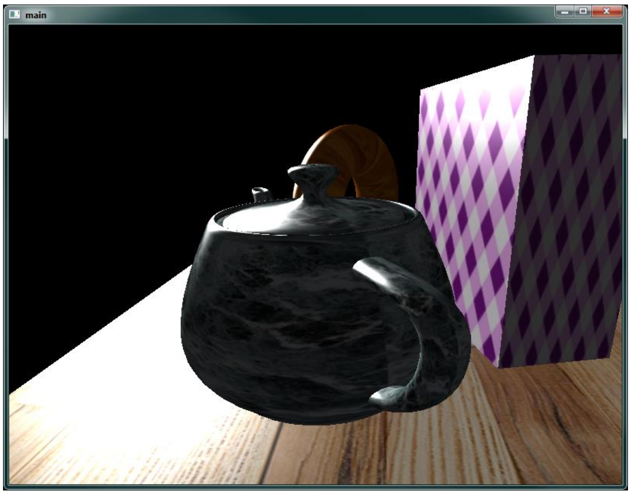
You are to complete an OpenGL program (a **fragment shader**) to add **bloom effect** to the rendered image of a scene. The following images show sample views of the result that your program is expected to produce:



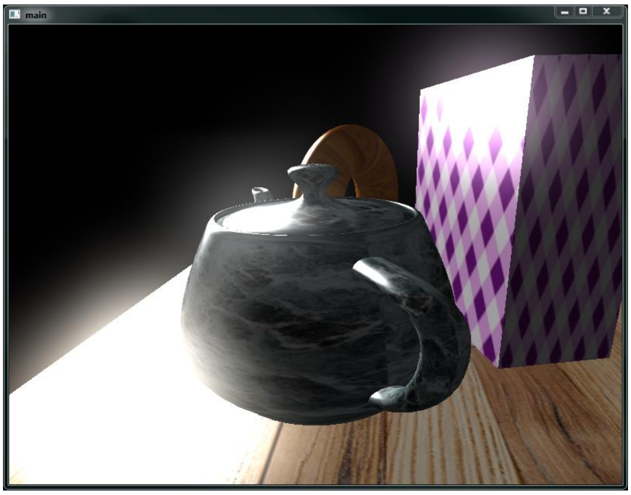
Without bloom effect.



With bloom effect.

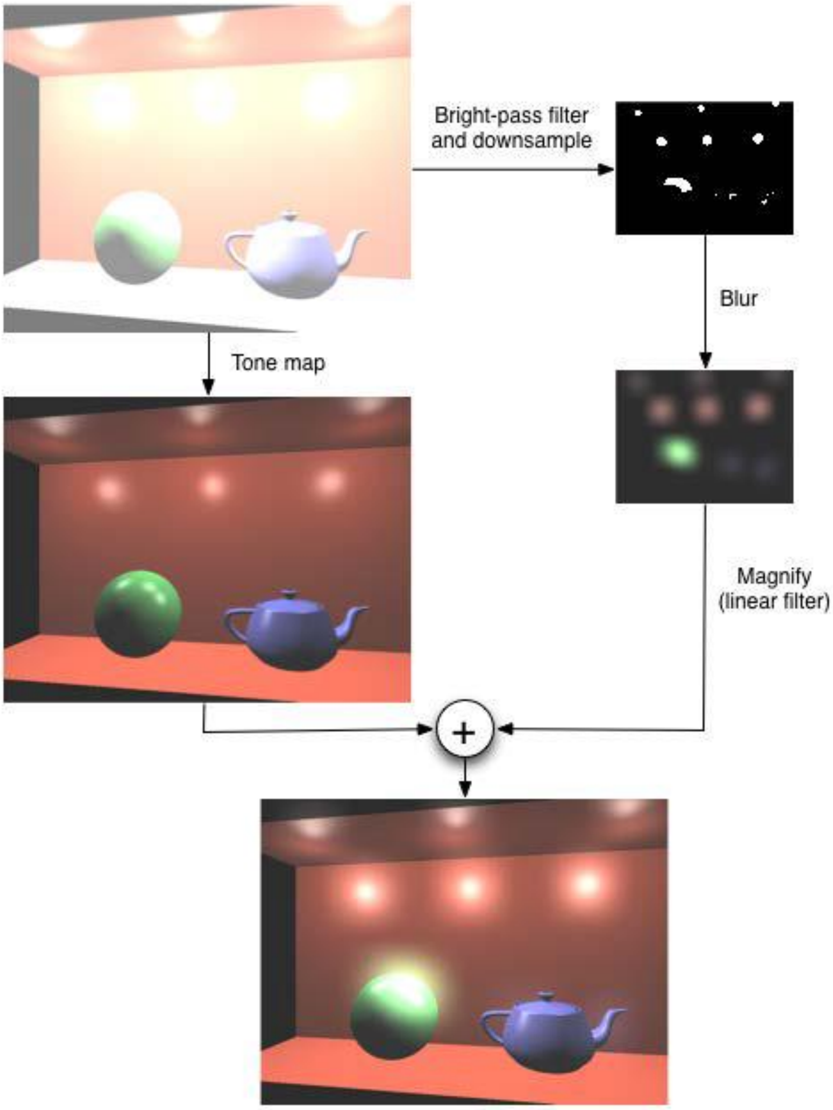


Without bloom effect.



With bloom effect.

You are to use the method presented in the lecture, which is outlined in the following diagram. In this assignment, however, there is no need to perform **tonemapping** of the original rendered image. The **bright-pass filtering** is performed by **image thresholding** based on pixel luminance. The image blurring is achieved by digital convolution with a separable 2D Gaussian filter, which is implemented as a 2-pass digital convolution using a 1D Gaussian filter.



Please download the ZIP file **`sws3005_2022_assign4_todo_(*).zip`** from the **Assignments** folder in LumiNUS Files.

You need to complete the fragment shader **`imageBloom.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 all the necessary shader source files, and use them in the rendering. It sets up the texture maps and framebuffers and 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 C/C++ source files**.

For this assignment, you are required to complete the functions **`ThresholdImage()`**, **`HorizBlurImage()`**, **`VertBlurImage()`**, and **`CombineImages()`** in **`imageBloom.frag`**.

You should study **`main.cpp`** carefully to help you complete the above functions in **`imageBloom.frag`**. You can run the executable application program **`main_done.exe`** (or **`main_done`** on macOS) directly to test your shader.

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.

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 **`imageBloom.frag`**.

You must put it/them in a ZIP file and name your ZIP file ***nus-user-id*`_A4.zip`**. For example, if your NUS User ID is **`t0912345`**, you should name your file **`t0912345_A4.zip`**.

Submit your ZIP file to the **Assignment #4 Submission** folder in LumiNUS Files. 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 ———