CSC8820 Advanced Graphics Algorithms
Spring 2018

Project #3

Due date: 11:59 pm, April 9, 2018 (Monday)

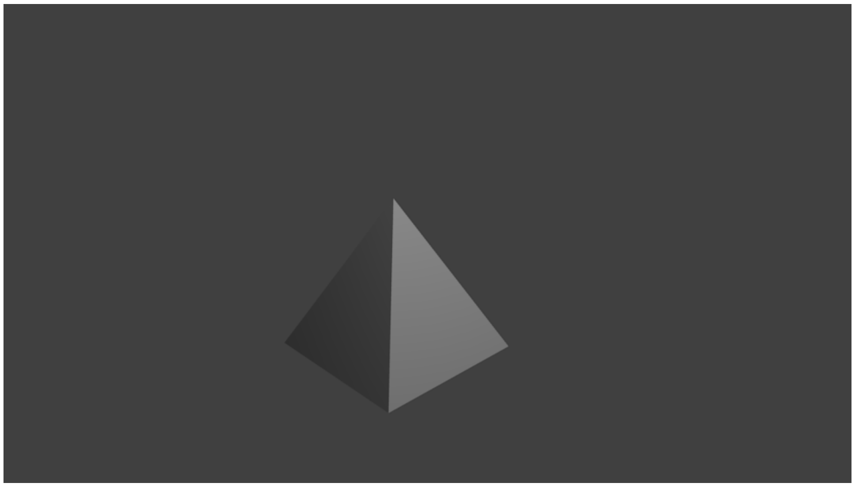
In this project, you will learn the basics of drawing and lighting using Vulkan API.

Requirements (100 points total):

1. Write a C or C++ program that draws the 3D object in the attached OBJ file. The output of the program should be like the picture below.
    a. The 3D object file pyramid.obj is attached.

    b. Use Vulkan to develop your program. You can download Vulkan SDK from https://vulkan.lunarg.com. Detailed instructions on how to install and compile Vulkan SDK is included with the package or can be found online. Additional instructions can be found at https://sites.google.com/site/csc8820/educational/vulkan-basics.

    c. (60 points) The object shall be lighted. You must write a vertex shader and a fragment shader. The lighting must be implemented in the fragment shader. There should be at least one light source.



2. (20 points) Draw two data flow diagrams. One diagram shows how the lighting parameters are transferred from the host program to the shader program. The other diagram shows how the vertex data is transferred from the host program to the shader program.
    a. Use free online diagramming tools such as Google Drawing.

3. (20 points) Answer the following questions:

a. Compare Vulkan and OpenGL in three areas: pipeline set, data transfer, and data processing. In which part is Vulkan more complicated than OpenGL?

b. What design decisions make Vulkan more efficient than OpenGL?

c. What is SPIR-V? Why use SPIR-V? What is the relationship between SPIR-V and GLSL?

4. Students whose computers do not support Vulkan still need to fulfill the requirements 1, 2, and 3. However, it is understood that the code may not compile but the code will be inspected for features. For these students, requirement #1 will be worth 40 points. The students need to answer the following questions (20 points):

   a. Describe how Vulkan layers can be used for debugging purposes.

5. Name your source file (lastname)_proj3.cc. In the source code, write your name and any special instructions for building and running your program. Either embed your shaders in your program or save them in separate files. Name your shaders (lastname)_vshader.glsl and (lastname)_fshader.glsl.

6. Submit your source code as well as a report. The report should contain your diagrams and answers to the questions (requirement 2, 3, or 4).

7. Upload your source code and report in a ZIP file to iCollege under the folder Project 3. Do not submit via email.