CS 535 Computer Graphics
Homework Assignment 2 Solution Set (40 points)
Due: 9/17/2024

1. (6 points)

   OpenGL has four buffers: *color buffer*, *depth buffer*, *accumulation buffer* and *stencil buffer*. Each buffer has a specific function/purpose. For instance, a stencil buffer is used to mask pixels in an image, to produce special effects. A color buffer, the so-called frame buffer, is where the image to be shown on screen stored. What are the functions of the depth buffer and the accumulation buffer? Put your answer in the following text boxes for each of them.

   Depth buffer:

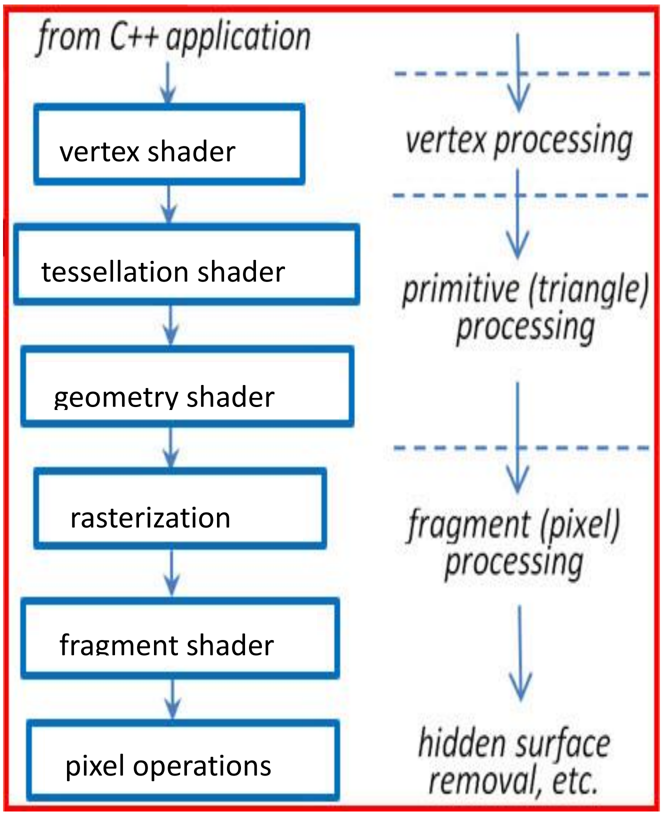
   The depth buffer is used by the graphics pipeline to implement the **Z-buffer method** to eliminate hidden lines/surfaces.
   Initially, each entry of the depth buffer is set to -1 (why?). During the rasterization step, the depth value (z-coordinate, a floating point number) of each point (fragment) on the edges of or inside a triangle is computed and stored in the corresponding entry of the depth buffer so that the pixel operations step can use the depth information to determine which object's color should be used for each entry of the color (frame) buffer.

   Accumulation buffer:

   The accumulation buffer is used by the graphics pipeline for two things: **double buffering** and **composing a sequence of successive images** to generate special effects such as **anti-aliasing**, **motion blur**, or **depth of field**.

2. (8 points)

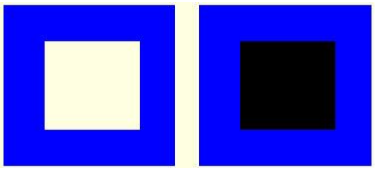
   Modern 3D graphics programming utilizes a pipeline. Each stage of the pipeline is done by a specific hardware. In the following chart, fill out the blanks for the names of those stages. (6 points)



Some of the stages are programmable and some are not. Which one(s) are not programmable and why? (2 points)

The **rasterization** and **pixel operations** are not programmable. These two stages use pre-defined algorithms to perform primitive (lines, triangles) scan-conversion process and hidden line/surface elimination process.

3. Based on example programs 1-6 in the notes "OpenGL and Shaders", write a simple C++/openGL program to show **one** of the following hollow blue squares of dimension 50x50 (pixels) at the center of your glfw window, the white (black) portion is of dimension 25x25 (pixels). Use "HW2 – Question 3" as the title of your glfw window and use black as the background color of your glfw window. Your vertex shader and fragment shader can be hard-coded in the C++/openGL program or written in separate glsl files.



Take a screen shot of the output of your program (make sure the window title is included in the screen shot) and include the screen shot in your HW2 submission, or turn in the screen shot as an attachment to your HW2 submission. Turn in your program as an attachment too. (10 points)

Sol:
Two approaches. You can either decompose the hollow square into four separate regions and draw each of them using the fixed (blue) color, or check the values of "**gl_FragCoord.x**" and "**gl_FragCoord.y**" to determine if a point should be painted in blue of white (black, in the second case).

4. "Event handling" used to be done by the openGL command "glutMainLoop ( )". But now is handled by the command "glfwPollEvent( )" in a while loop. The first instruction in that while loop is the "display( )" function. What is the advantage of this kind arrangement? Put your answer in the following blank. (6 points)

With the "*display( )*" function arranged as the first step in the while loop, **animation is automatically supported by a modern C++/openGL application** – simply render the entire scene each time *display( )* is called.

5. What would happen to your program for Question #3 once it is executed if an event other than clicking the "x" symbol at the upper-right corner of your glfw window, such as "pushing a keyboard key" or "clicking at an arbitrary point inside the glfw window" is generated and why? (5 points)

Nothing. As there are no callbacks defined for any other event types.

6. Depth buffer and color buffer should both be cleared when doing hidden surface elimination (slide 61 of the notes "OpenGL and Shaders"). Why? (5 points)

The depth buffer is cleared each frame **to ensure hidden surfaces are removed properly.** The color buffer is cleared to **avoid a moving object leaving a trail as it moves**. This is mainly useful for moving objects. Note that if the depth buffer is not cleared, a moving object will still be shown in the non-overlapping area from the previous frame. Why?