Assignment 1 - TDT4195

Nora Evensen Jansrud, Olaf Rosendahl

September 06, 2022

Task 1

c)

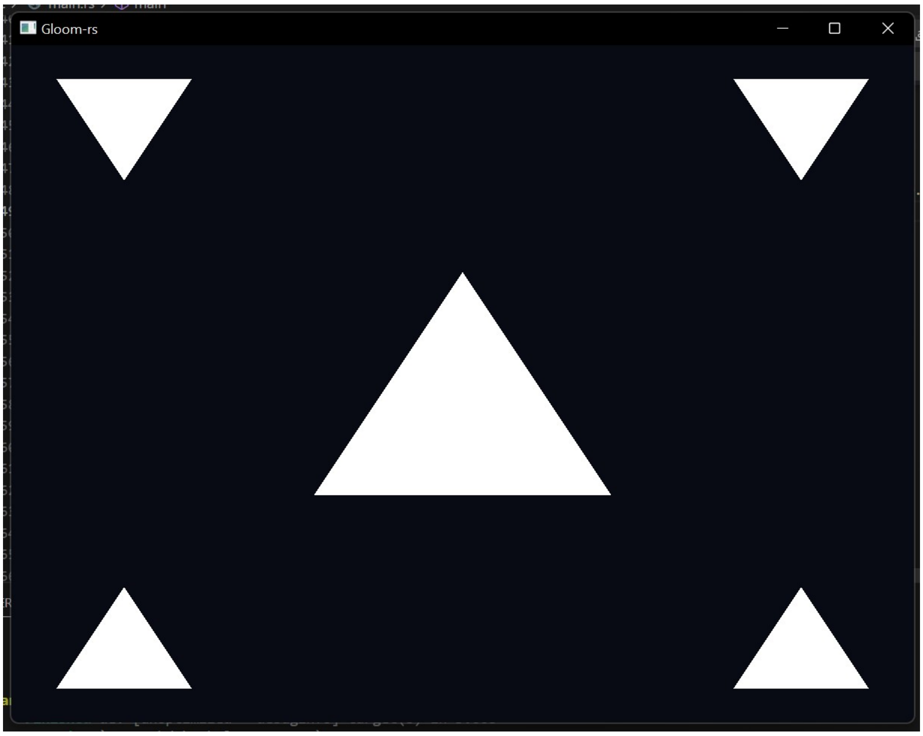


Figure 1: 5 distinct triangles

Task 2

a)

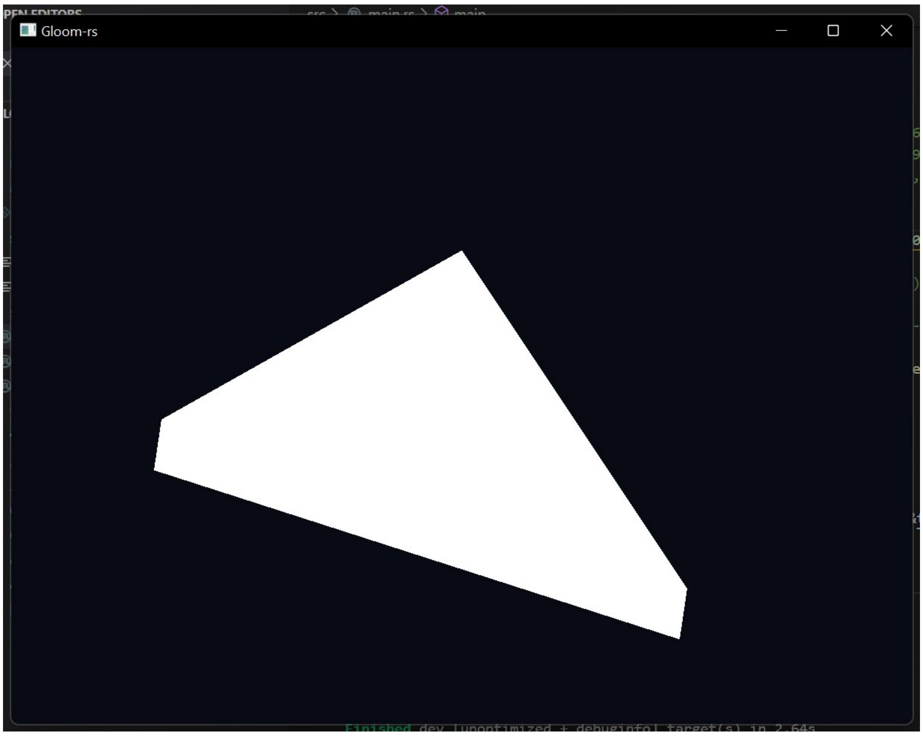


Figure 2: A single triangle passing through the given vertices

1. What is the name of this phenomenon?

   This is clipping. The OpenGL Clipbox is a three-dimensional box with a size of 2x2x2. The coordinates go from -1 to 1 in each dimension. Content that exceeds these coordinates will be clipped letting only the content inside the clipbox left and visible.

2. When does it occur?

   It occurs when a object like for example a point, line or triangle goes out of the clipbox.

3. What is its purpose?

   The purpose is to remove primitives like vertices outside the clipbox. This ensures that only primitives that should be rendered are part of rendering.

b)

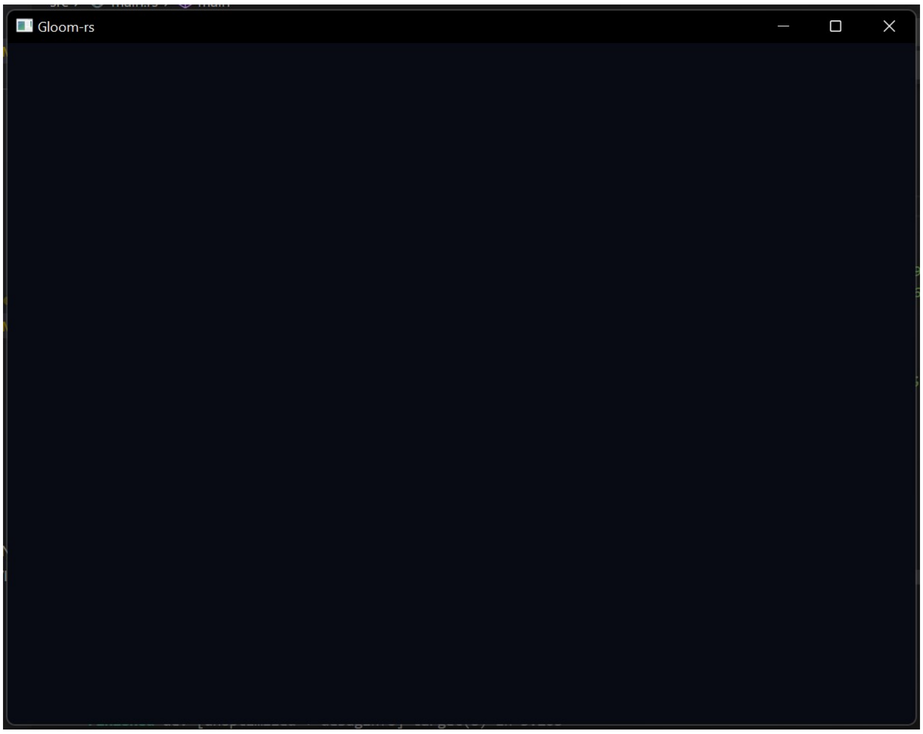


Figure 3: A single triangle with modified index buffer

1. What happens?

   This is called back-face culling.

2. Why does it happen?

   This happens when the direction of the vertices aren't clockwise. OpenGL uses the direction of the vertices to determine whether the triangle is front-facing or back-facing. Back-faced triangles are "culled"/removed from the rendering-process which increases performance since less have to be drawn.

   What is the condition under which this effect occurs? Determine a rule.

   The effect occurs when the "right hand rule" isn't followed. OpenGL uses a right handed system which requires the normal to point towards the screen/eye to avoid back-face culling. The index buffer must therefore be in the correct order.

c)

1. Why does the depth buffer need to be reset each frame?

   To avoid comparing pixels in the current frame with the depth values from the previous frame.

In which situation can the Fragment Shader be executed multiple times for the same pixel? (Assume we do not use multisampling.)

When there is multiple shapes that overlap eachother, the fragment shader will be executed once per shape and therefore multiple times for the same pixel.

2. What are the two most commonly used types of Shaders? What are the responsibilities of each of them?

   Fragment shader is responsible for giving colors to the pixels.

   Vertex shader determines the positions of the screen elements by transforming the input to the 3d-space.

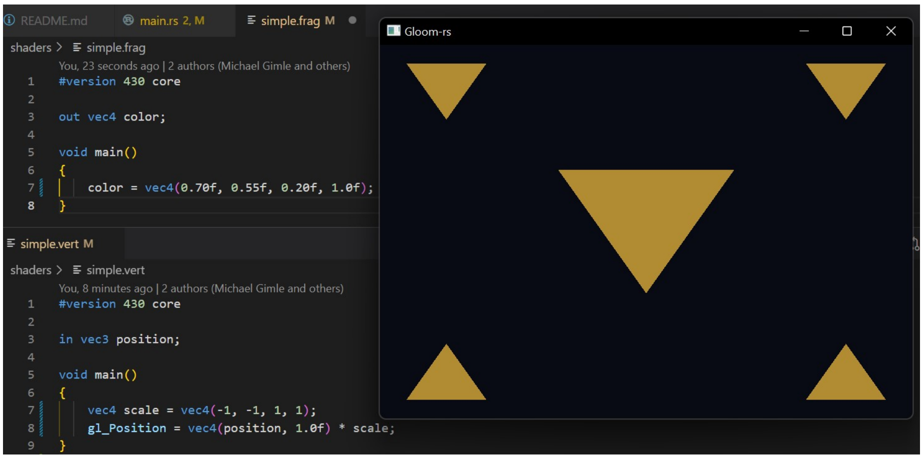
3. Why is it common to use an index buffer to specify which vertices should be connected into triangles, as opposed to relying on the order in which the vertices are specified in the vertex buffer(s)?

   An index buffer helps optimizing shapedrawing. Triangles is used since it requires the lowest amount of vertices. In order to draw for example a circle or a square, one can easily combine mulitple tringles. Since multiple triangles is combined, the index buffer makes them share vertices instead of having duplicates. This make the program more effecient, especially when your're create more complex shapes.

4. While the last input of gl::VertexAttribPointer() is a pointer, we usually pass in a null pointer. Describe a situation in which you would pass a non-zero value into this function

   When we store more than just coordinates in the buffer, for example texture coordinates. In those cases we have to tell OpenGl where this extra information starts.

d)


README.md
main.rs 2, M
simple.frag M
shaders > simple.frag
You, 23 seconds ago | 2 authors (Michael Gimle and others)

```
#version 430 core

out vec4 color;

void main()
{
    color = vec4(0.70f, 0.55f, 0.20f, 1.0f);
}
```

simple.vert M
shaders > simple.vert
You, 8 minutes ago | 2 authors (Michael Gimle and others)

```
#version 430 core

in vec3 position;

void main()
{
    vec4 scale = vec4(-1, -1, 1, 1);
    gl_Position = vec4(position, 1.0f) * scale;
}
```

Figure 4: The whole scene flipped both horizontally and vertically

In order to flip the scene we used a matrix in the vertex shader, as seen in Figure 4. This matrix scaled the scene with a value of -1 in both x- and y-direction which flips the objects. Since the scene is flipped twice, back-face culling isn't a problem because the triangles which faced towards the camera still faces the camera.

In order to change the colors of the drawn colors, we simply changed the RGB-color in `shaders/simple.frag`-file by editing the color-variable. As seen in Figure 4.