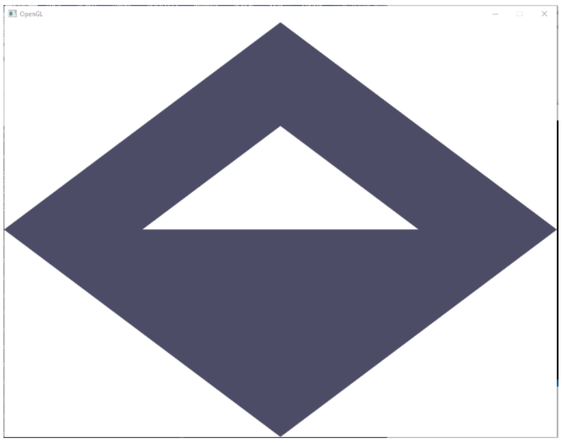
Exercise 4 – Graphics
Mikkel Svagård

Primitive Rendering

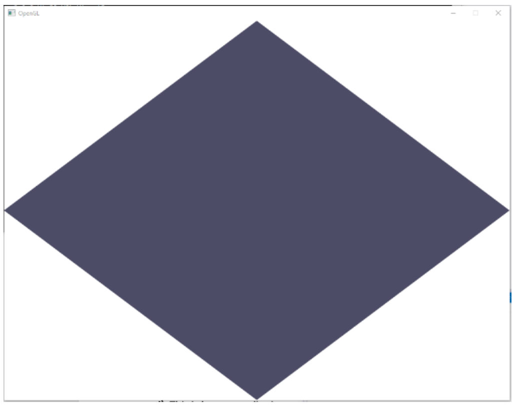
b) 5 triangles →



c)

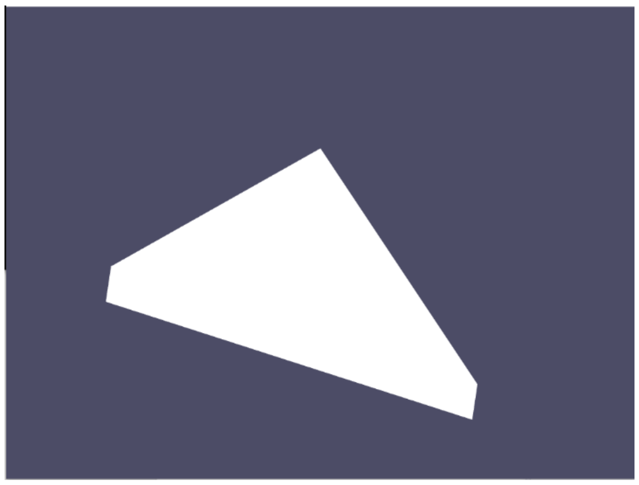
Changing the order of the vertices of a triangle sometimes causes it to be rendered away. This is due to the program interpreting the triangle to be faced away from the viewer in some of the cases, and its therefor optimized away as the viewer only would view it's not-existing backside.

A rule to this is given by clockwise and counter-clockwise positions of the vertices making up the triangle. Vertices numerated in a counter-clockwise manner gives the face of the structure, while clockwise gives the backside. Hence all triangles to be shown on the screen has to have vertices in a counter-clockwise order.



An example of this is that the triangles in the centre above. Those were numerated clockwise. By reversing the order of the vertices the triangle disappears.

d) This is known as clipping, as some of the triangle is outside the clipbox and then not suppoes to be viewed. This is caused by the z-dimension of the vertices, where bottom right and bottom left has 1.2 and -1.2 z-coordinates accordinlgy.



A trinagle entierly outside of the clipping box gets completely removed, which is known as culled.

The reason for this is enchansing performance, as unneccesary pixels wont get rendered.

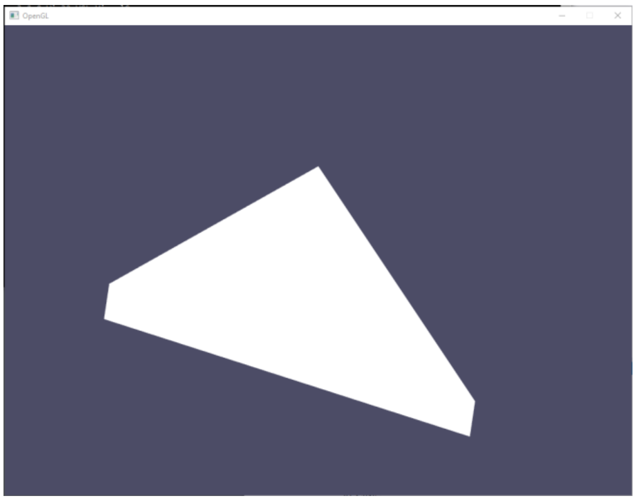
Exercise 4 – Graphics
Mikkel Svagård

Simple Shaders

a)

Shaders are small programs on the CPU where they take part in the rendering pipeline, while Program Objects are bundles of shaders. The two most commonly used shaders are the Vertex and Fragment Shaders. The Vertex Shader transforms the different vertices around the scene (operations such as translating, scaling and rotating) while also responsible for projecting the scene to the camera. Fragment Shader determines the color of each fragment defined by the vertices.

c)

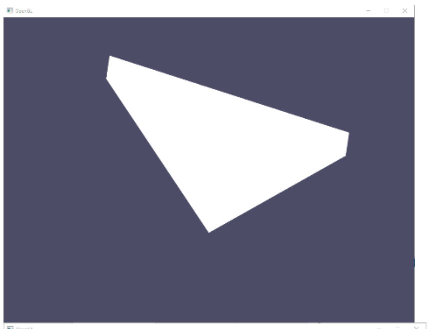


Applying both shaders didn't make any difference. The reason for this is that the vertex shader only forwarded the position while the fragment shader recolored with the color it was in the begginning: white.

Layout qualifiers is paramaters giving the shaders to know which buffers the values its working with originate from.

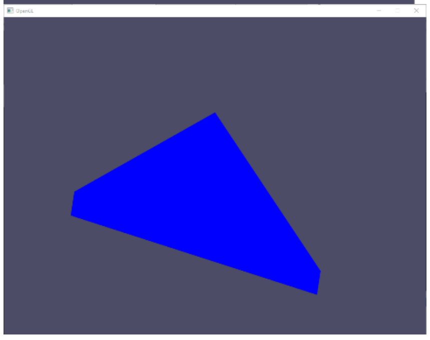
While applying shaders*, Vertex attribute* is parameters regarding individuel vertices. *Uniform variables* on the other hand are constant parameters that are identical for all instances of the shaders being executed.

d)



Flip

Achieved by manipulating the positions in the Vertex Shader. This was done by flipping over the axis by setting the x and y coordinate to (-position.x) and (-position.y)



Change color:

By editing the Fragment Shader I changed the initial color from (1.0f, 1.0f, 1.0f, 1.0f) to (0.0f, 0.0f, 1.0f, 1.0f), only leaving the color blue and visibility to 1.

Both applied together with green →