

Computer Graphics – Assignment 1

Margot Sirdey & Balázs Orbán

Drawing your first triangle	1
Geometry and theory	2
Sources	5

1. Drawing your first triangle

- look at the source code
- look at the source code
- Result: 5 triangles, *Figure 1*

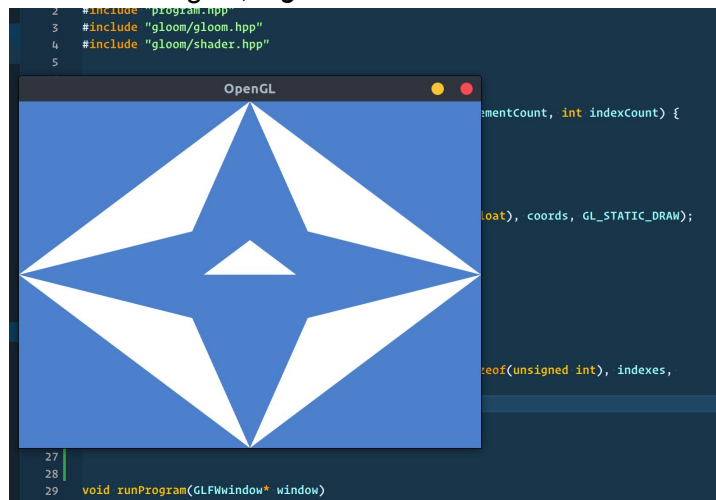


Figure 1 - The 5 triangles

2. Geometry and theory

a. Result: *Figure 2*

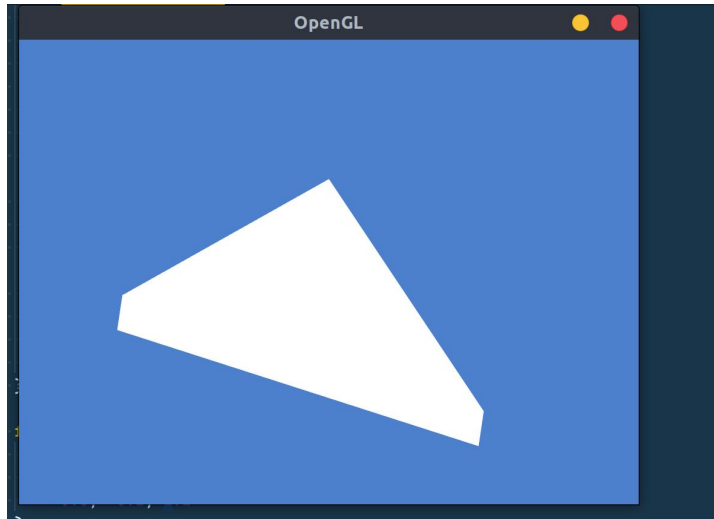
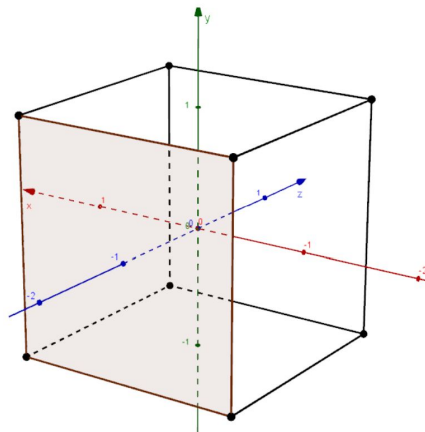


Figure 2 - The clipped triangle

- i. The name of this phenomenon is « clipping ».
- ii. This occurs when the coordinates of the object goes outside of the clipping volume. (In this case -1,2 and 1,2 on the z-axis). This is a $2*2*2$ cube, as *Figure 3 shows*. We changed the 1,2 to 1,0 on the z-axis. Then the triangle appeared entirely. The vertex shader is initialized at size 1,0.



*Figure 3 - The clipping volume (a $2*2*2$ cube, ranging from -1 to 1 on each axis)*

- iii. The purpose is to improve performance by ignoring the parts the user can not see.
- b.
 - i. If we change the order in which a single triangle's vertices are drawn, the triangle disappears.
 - ii. Backface culling
 - iii. The rule is that while drawing an object, the indexes in the index buffer must be increasing. (when we come to the end of the object's last

index, we jump to the first one.) Example with a single triangle, with indexes 0, 1, 2, According to the defined rule, the possible orders, which will successfully draw the triangle are: 0, 1, 2 or 2, 0, 1 or 1, 2, 0.

c.

- i. Depth buffer, also named Z-buffer, is a two dimensional array which stores the z-value for each pixel. Its purpose is to determine if an object is visible or not. (Source 1)



ComputerHope.com

Figure 4

- ii. The Program Object groups the Shader Objects together, which are compiled codes.
- iii. The two most commonly used shaders are: the vertex shader and the fragment shader. The first one is to perform transformations. The other one is responsible for determining the color of each fragment.
- iv. The index buffer kept separate from the vertex attributes in order to define vertices only once so they can be reused.

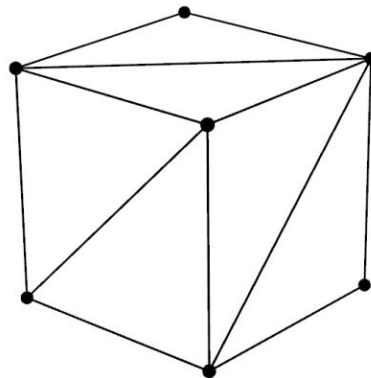


Figure 5

- v. Vertex Attributes, contained in the VAO, are inputs of the Vertex Shader. Then the Vertex Shader only specify which input should be drawn or not.
- d.
- i. To mirror the whole scene vertically and horizontally, we multiplied the position by -1 in the vertex shader. For the result, look at *Figure 6* and *Figure 7*.

- ii. We changed the color values in the fragment shader. Look at *Figure 6* or *Figure 7*.

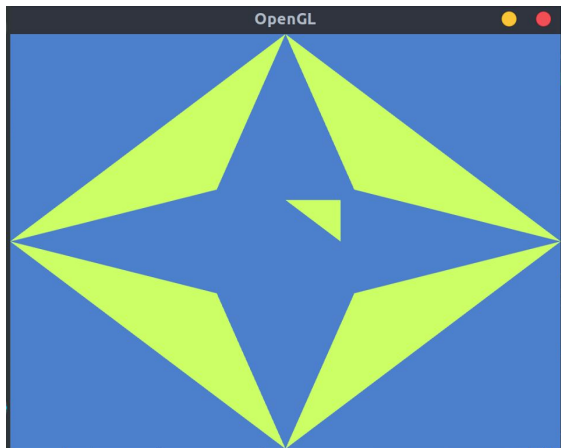


Figure 6 - before

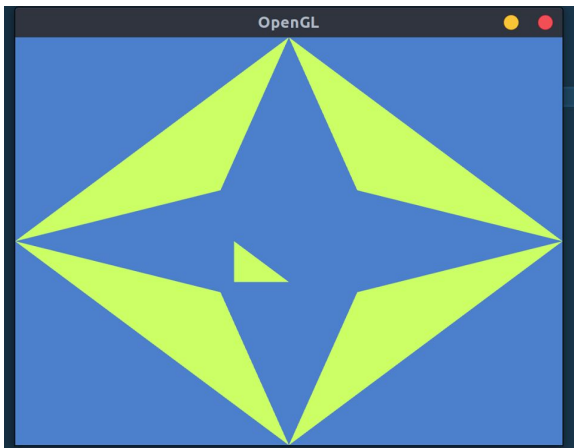


Figure 7 - after

Sources

1. <https://www.computerhope.com/jargon/z/zbuffering.htm>
2. Hitchhiker's guide to OpenGL