Computer Graphics - Assignment 1

Christofer Gilje Skjveland

September 6, 2019

Task 1

c)

The five triangles are shown in figure 1.

Task 2

a)

The result is shown in figure 2.

i)

The name of this phenomenon is clipping.

ii)

This occurs when the value of x, y or z for a vertex is larger than 1.0 or lower than -1.0. These are the maximum and minimum values for x, y and z. Everything else is clipped and therefore not shown.

iii)

The purpose of clipping is to not render anything that is not shown on the screen.

b)

The result of scrambling the indices for the five triangles are shown in figure 3.

The triangles that is drawn has a certain ”front” side and ”back” side. When the back side is in front the triangle is hidden, when the front side is in the front the triangle is shown. If the indices are in the right order the triangle is shown if the vertices are drawn in a counter clockwise manner. This is called face culling. The reason for this phenomenon is because the back side has not been rendered, only the front.

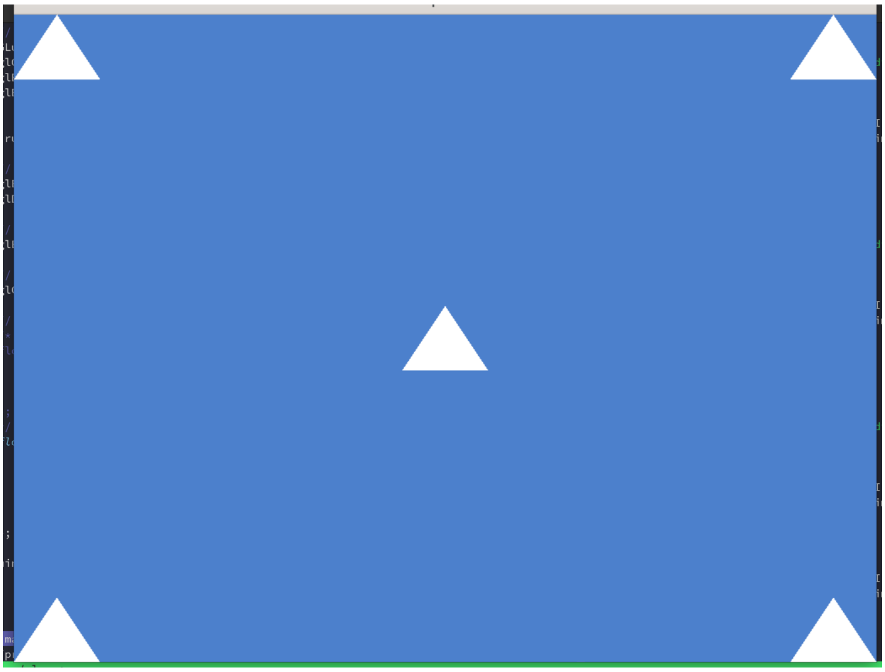

Figure 1: 5 triangles

c)

i)

Each fragment is compared to the depth buffer to determine what depth the fragment is at in the frame. If another fragment is in front, then the new fragment won't be rendered. At next scene the screen is cleared and everything has to be redrawn. The depth buffer will therefore have to be cleared since the fragments in the new scene should not have to be compared to the depths of the old scene.

ii)

When the scene is very complicated, several fragments migth occupy the same pixel. In this case the fragment shader is executed multiple times for the same pizel.

iii)

The two most common shaders are the freagment shader and the vertex shader. The vertex shader manipulates the attributes of the vertices. The fragment shader manipulates how the pixels between the vertices look.

iv)

Some times the vertices are part of several triangles. Instead of defining vertices several times, an index buffer can be defines instead. This way we can refer to a index instead and save quite a bit of memory usage.

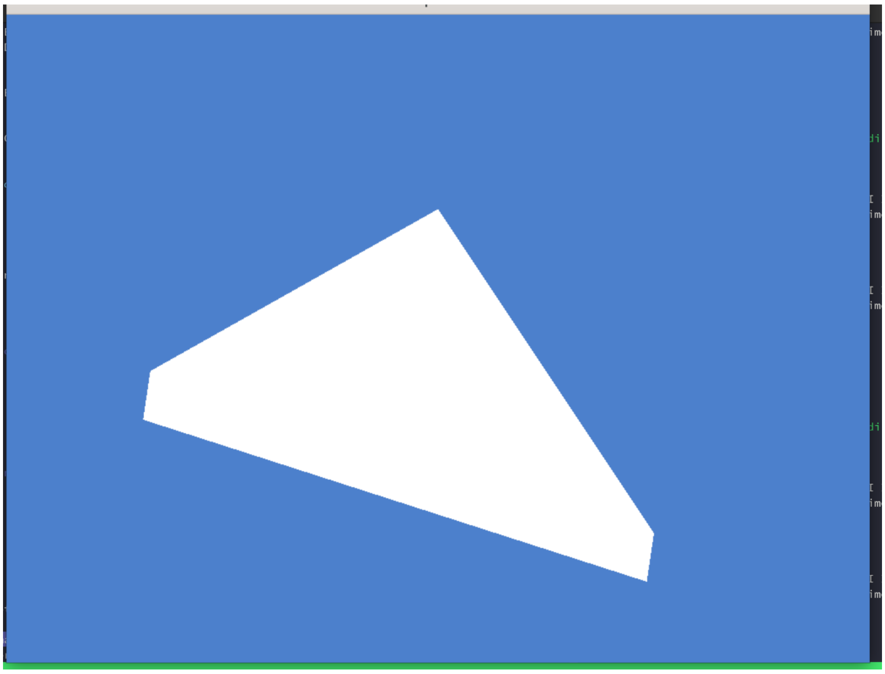

Figure 2: Clipping

v)

If there are several entry types in the vertex buffer object, then the pointer must be non-zero to refer to the next entry type.

d)

The effects are shown in figure 4.

Each effect is accomplished by modifying simple.frag and simple.vert. The changes can be viewed in the code.

Sources

`https://www.khronos.org/opengl/wiki/Face_Culling https://stackoverflow.com/questions/19469194/why-do-we-have-to-clear-depth-buffer-in-opengl-duri`
`https://stackoverflow.com/questions/4421261/vertex-shader-vs-fragment-shad`

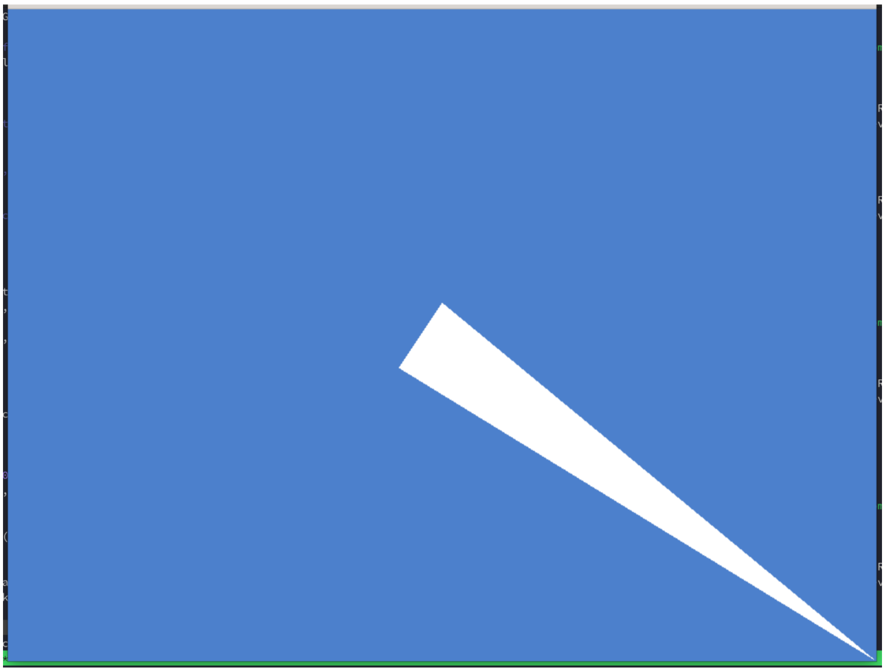

Figure 3: Scrambled indices

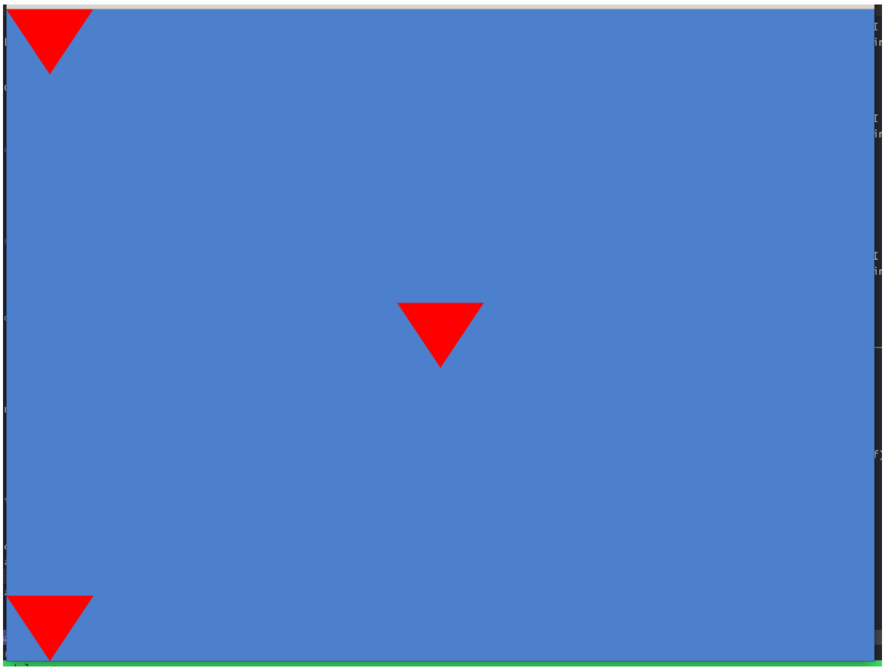

Figure 4: Modified shaders