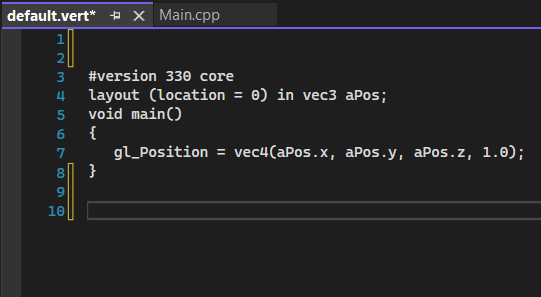
Think of shaders as functions that run on GPU

Since they are functions they can take input and also have outputs



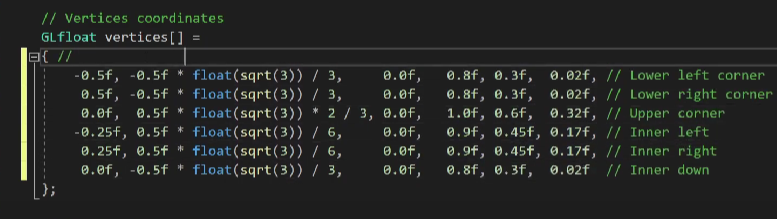
This is our default vertex shader

It might seem like a C code

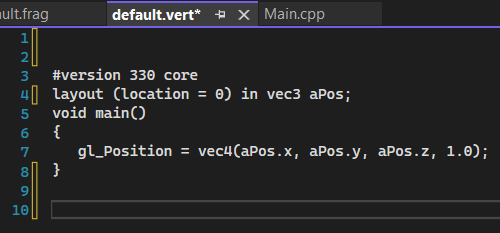
But

It is actually, OpenGL Shading Language ( GLSL )

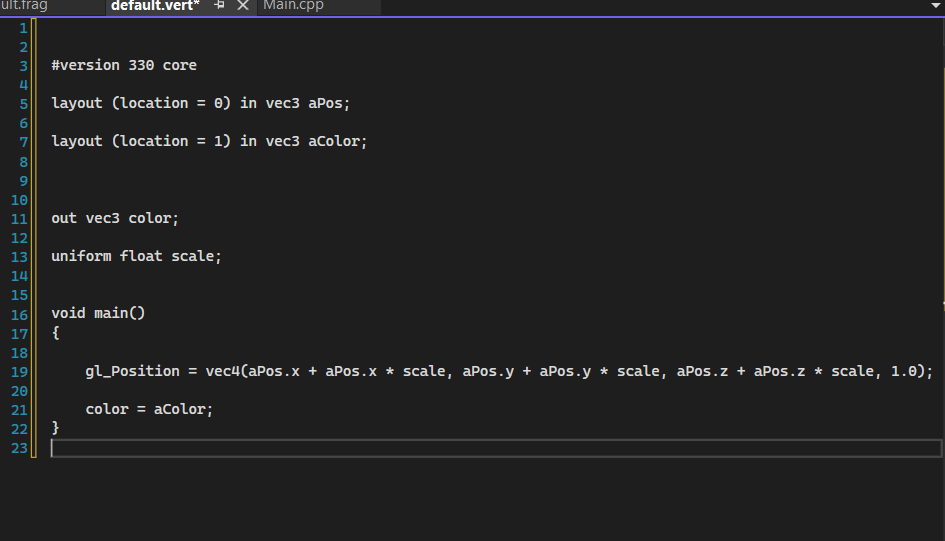
Let say we want to give color to the each vertices like this



For this to work we have to make certain changes to the vertex and fragment shader



This was previous vertex shader

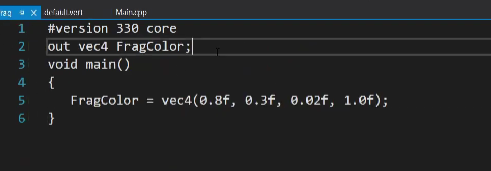


This is latest vertex shader

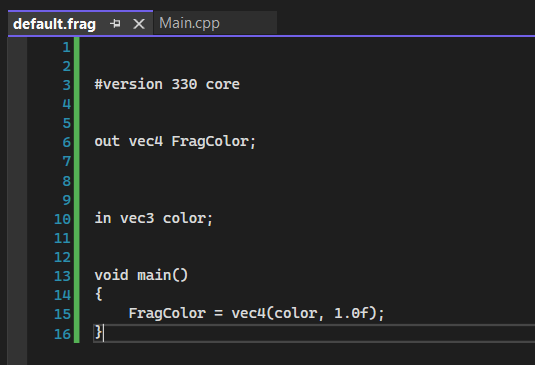
Here

uniform float scale;

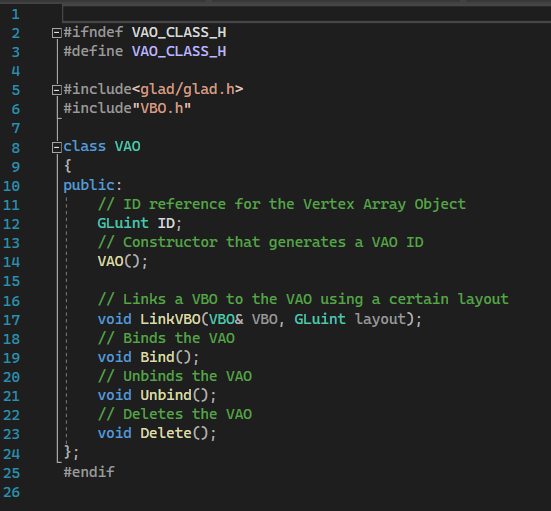
has another function



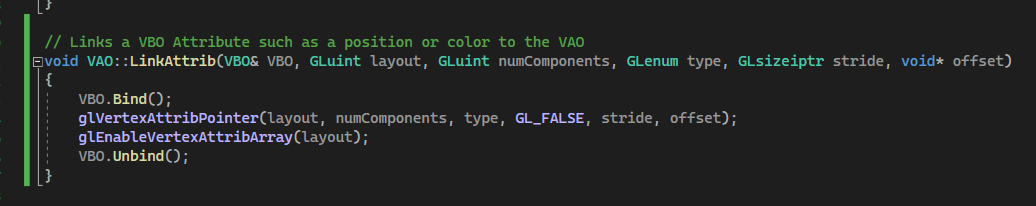
Previous fragment shader



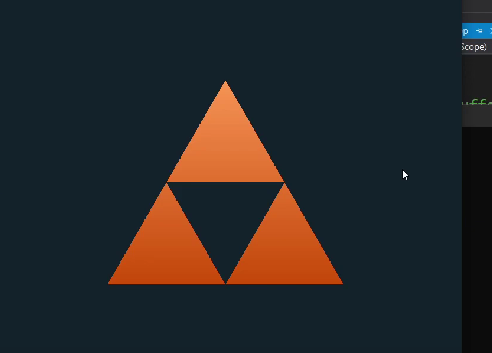
Latest fragment shader



We also need to make changes to the VAO class, VAO.h



After in VAO.h class we also made similar changes to the VAO.cpp class

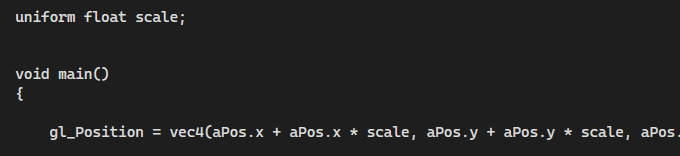


We will get similar output

Now why are we getting a gradient of color when we only specify a couple of colors?

Because if the primitive has different colors for it’s vertices than open gl will auto create a gradient from one color to another

This is called interpolation



This scale is called uniform in open gl

Here it will increase or decrease our shapes