|  |
| --- |
| Computer Games: Software Development |
| Graphics Programming |
| Games Programming 2 |

|  |
| --- |
| Module Leader: Bryan Young ([bryan.young@gcu.ac.uk](mailto:bryan.young@gcu.ac.uk))  Sam Clements S1828589 |

*Session 2019 - 2020*

***Plagiarism***

*Attention is drawn to the University regulations on plagiarism. Whilst discussion of the coursework between students is encouraged, the actual work has to be undertaken individually. Collusion may result in a zero mark being recorded for the coursework for all concerned and may result in further action being taken.*

*Sam Clements*

Contents

[1. Custom Shader 2](#_Toc40213236)

[2. Vert 2](#_Toc40213237)

[3. Frag 2](#_Toc40213238)

# Custom Shader

The custom shader is an edited version of toon and fog, the aim of the shader was to develop a custom shader that combined both aspects of the aforementioned shaders to produce a new effect.

As a combination of two previous effects certain aspects of the code remain similar to the code from the uncombine versions.

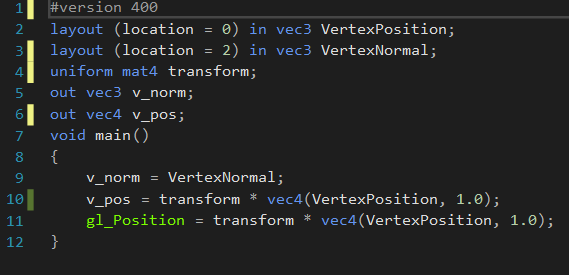
# customShader.Vert

The customShader.vert file utilises two vector 3 components, the first for the vertex position and second for the vertex normal (these can be seen at lines 3 and 4 of figure1).

To enable integration with the frag shader the variables v\_norm and v\_pos (these can be seen at lines 8 and 9 of figure1) these variables are sent to the customShader.frag shader from the customShader.vert shader.

The v\_norm and v\_pos variables are set to to VertexNormal and transform multiplied by the VertexPosition respectively (This can be seen lines 13 and 14 of figure 1)

(Figure 1: customShader.vert screenshot)



# customShader.Frag

The customerShader.frag takes in in v\_norm and v\_pos from the customShader.vert file (This can be seen at lines 3 and 4 in figure 2).

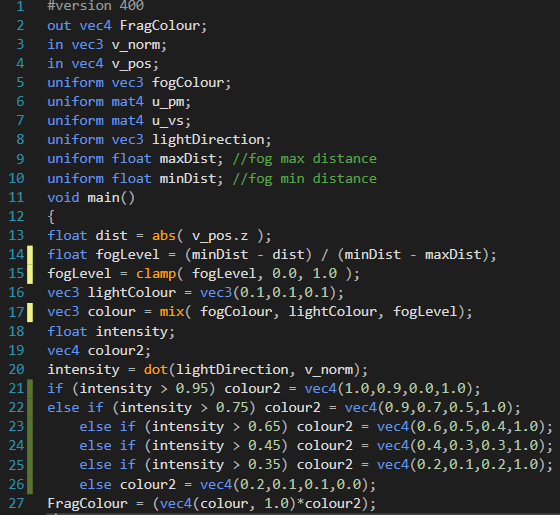
Next is the fog colour labelled in visual studio as fogcolour (line 5 in figure 2) this is a vec3, this vec 3 is actually set in maingame.cpp. This colour can be altered by changing the rgb values seen at line 336 in figure 3, the current fog colour is set to green.

To calculate the fog a light direction is needed in this case lightDirection (line 8 in figure 2), lightDirection stores the current position of the light source. LightDirection is set in maingame.cpp and is also a vec3.

To achieve the desired fog effect the amount of fog is set via the float fogLevel, fogLevel is set via the calculation minDist – dist divided by minDist – maxDist (as seen in line14 in figure 2).

minDist, maxDist are both floats, whilst Dist is the position of the object upon the Z-Axis and the above calculation is used to evaluate how far away the object in this case the custom shaded egg is from the available light source. The combination of fogColour, lightDirection and fog level allow the Toon colouring to darken and lighten depending upon how far away the object is and how much light is hitting each Vertex, the toon shading is then calculated via the intensity of the light on each vertex (as seen in lines 21 through 26 in figure 2) the intensity is the dot product of the light direction and v\_norm the final part is to multiply colour (the combined product of fogColour, lightColour and fogLevel and colour2.

(Figure 2: customShader.frag screenshot)



(Figure 3: mainGame.ccp screenshot)

