|  |
| --- |
| Computer Games: Software Development |
| Graphics Programming |
| S1828592  *I confirm that the code contained in this file (other than that provided or authorised) is all my own work and has not been submitted elsewhere in fulfilment of this or any other award*.  *Ritchie Alasdair MacLean*. |

|  |
| --- |
| Ritchie Maclean  Session 2019/2020 |

Contents

[1. Combo Shader 2](#_Toc40023692)

[1.1 Light Source 2](#_Toc40023693)

[1.2 Fog Shader 2](#_Toc40023694)

[1.3 Toon Shader 3](#_Toc40023695)

[1.4 Final Colour Combo 3](#_Toc40023696)

# 1. Combo Shader

For the combo.vert file there are two vector 3s used. Firstly, the vertex position and then the normal of the vertex.

Out vec3 lets us send the variable v\_norm form the vertex shader to the fragment shader.

This is also the same for the out vec4 in which v\_pos is sent from vert to frag.

In the main void we set the v\_norm to be the VertexNormal.

The vertex position (v\_pos) is set to be the transform multiplied by the vertex position.

## 1.1 Light Source

The first uniform in the fragment shader is a vec3 called lightDir. This will store the position of the light source. LightDir will be set in the maingame.cpp. For this shader, the light source will move in a circle to fully illustrate the shader effects. This has been down using the following formula see fig1.

A close up of a map

Description automatically generated

Figure 1 Circle formula

The X position is set to cos(counter2) \* 0.5. The 0.5 is equal to the radius of the circle the light source will move in and counter2 is an incrementing variable that is equal to the current angle.

The same again is then done for the Z position but this time is it sin of the angle.

Two effects will be applied by this shader. There will be a fog effect and a toon shading effect.

## 1.2 Fog Shader

To make the fog effect there will be a combination of the colour of the fog, the colour of the light and finally how strong the fog is.

The fog colour is a vec3 uniform that is set in the maingame.cpp. It is currently set up as yellow is colour rgb, = (1.0, 0.8, 0.0). The light colour is another vec3 that is set in the frag shader. Rbg = (0.1, 0.1, 0.1)

Finally, the fog factor is set with a distance calculation. What we want to happen is for the object to get darker as it gets further away. To do this we take the position of the vertex on the z-axis, this will set as a variable call dist. The calculation is the (minimum distance – dist) / (minimum distance – maximum distance). minDist and maxDist are both unifroms that are set in the maingame.cpp.

The three values are then mixed to give be us a colour. Mix(fogColor, lightColor, fogFactor). As the mesh moves further down in z axis, t will alter the fogfactor making it stringer the further away it gets.

To apply this, we would make this colour mix equal to FragColor, but the toon effect must also be adding in before that.

## 1.3 Toon Shader

To the get the toon shading effect the intensity of the light on the vertex must be calculated. To work this out we need the Light direction and we need to know the surface normal of then vertex.

The intensity is equal to the dot product of the light direction and the surface normal.

After that, the colour is set depended on the level of intensity. The intensity will be a value from 0 to 1. See fig2.

A screenshot of a cell phone

Description automatically generated

Figure 2 Intensity

## 1.4 Final Colour Combo

Now to get the final colour that we want applied to the mesh, the fog and the toon colour must be combined. This s set simply be multiplying the two colours together. FragColour = fogcolour \* tooncolour.