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.

Albert Mas Compés

Coursework

Graphics Programming

Albert Mas Compés (s2008093)

2020

# Extra Shader

For the extra shader I have added the combination of the Fog, Toon and Rim shaders.

This shader calculates the position and orientation of each polygon of the mesh in relation to a light source. Depending on the result assigns a colour to that polygon. Then calculates the distance to the camera and obscures this colour if the distance is over the minimum fog distance. At maximum fog distance, the colour will be darkened completely.

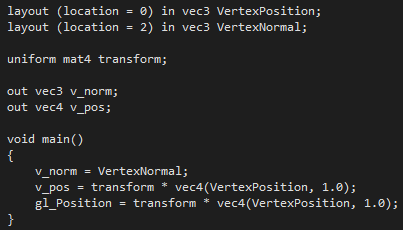


Mesh near the camera. Colour is not darkened.



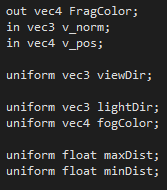
Mesh far from the camera. Colour is greatly darkened, though not completely.

**Vertex Shader**



Normal, Position and FragColor are passed onto the fragment shader.

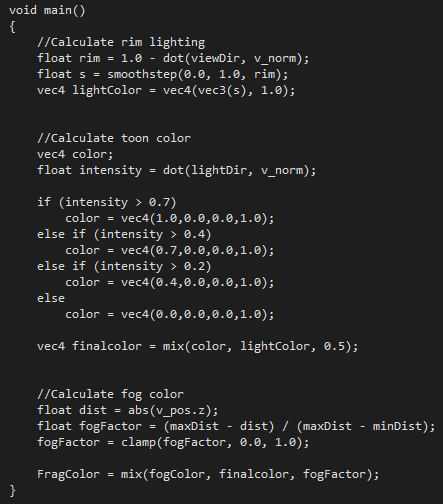
**Fragment Shader**



Receives Normal, Position and FragColor from vertex shader.

On every draw, uniforms are passed onto the shader with updated values.

* viewDir contains the direction where the camera is looking at.
* lightDir tells where is the light coming from.
* fogColor sets the colour that will be mixed with the toon colour depending on the distance. Black will darken the colour for a somewhat realistic fading of the mesh. Grey will make it look like the mesh is entering a thick fog or smoke.
* maxDist sets the distance where the mesh’s colour will have completely faded and just the fog colour will remain.
* minDist sets the distance at which the model’s colour starts to fade, mixing the fog’ colour into it.



The main function has 3 parts:

* Calculate Rim lighting

Calculates the amount of light it receives, depending on the angle with the camera view.

* Calculate Toon colour

Calculates the base colour, depending on the angle with the light source; and mixes it with the light colour.

* Calculate Fog colour

Calculates the factor at which the final mesh colour should be mixed with the fog colour, depending on the minDist, maxDist and distance.

If distance is smaller than minDist, factor will be over 1.0 and only the final mesh colour will be used.

If distance is greater than maxDist, factor will be under 0.0 and only the fog colour will be used.

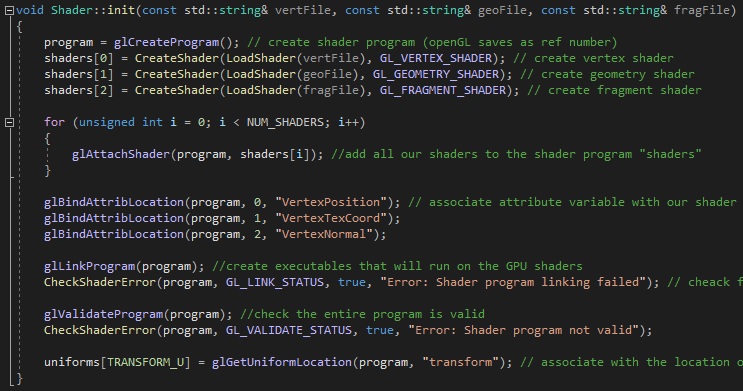
Then, the factor is clamped between 0.0 and 1.0, as these are the min/max values to mix the colours.

Finally, both colours are mixed with the obtained factor.

# Shader Class

This is the class in charge of Loading, Creating, Binding and Updating shaders.

**Init function**

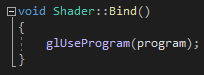


This function is called at the start of the application to load and create the vertex, geometry and fragment shaders for a new Shader program.



An overload of this function exists for shader programs without geometry shader.

**Bind function**

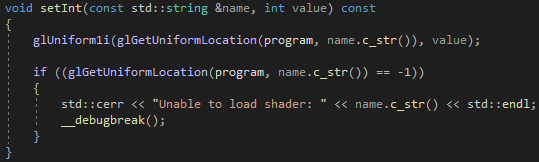


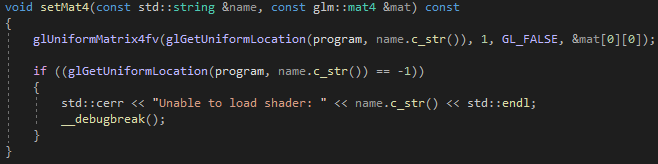
Before rendering a mesh, the desired Shader program will have to be bind by telling GLSL to use it in the rendering process. To unbind the current program, the same function must be called with a 0 as parameter. glUseProgram(0);

**Uniform Setter functions**

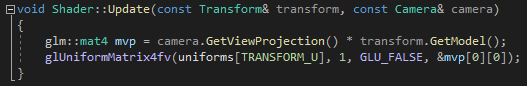
Before rendering a mesh, the desired Shader program might have variables that need to be updated (e.g. camera position, light colour, app timer). For that reason, lots of setter functions are needed to modify uniform variables in the shader.

Some of those are:





**Update function**



Because all shader programs need the transform uniform updated, there is a specific function in the shader class that does this.