Graphics Programming

Coursework

Glasgow Caledonian University

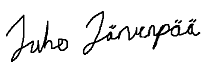
School of Computing, Engineering and Built Environment

Applied Computer Games

2021

Juho Järvenpää s2106881

*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*.



|  |  |  |
| --- | --- | --- |
|  | | Abstract |
| Author(s)  Järvenpää, Juho | Type of Publication  Research report | Published  2021 |
| Number of Pages  5 |  |
| **Graphics Programming**  **Coursework** | | |
| Name of Degree  Bachelor of Engineering, Information and Communications Technology | | |
| Name, title and organization of the client  - | | |
| Abstract  Topic of the coursework is to create shaders for 3D models using OpenGL API and OpenGL Shading Language. Visual studio and C++ are used as environment. Using OpenGL API requires some common knowledge about graphical programming and becoming familiar with OpenGL documentation. Shaders can be used to change the visuals of objects significantly. | | |
| Keywords  OpenGL, Shader | | |

Table of Contents

[1 Basic Lighting 1](#_Toc71382937)

[1.1 Ambient lighting 1](#_Toc71382938)

[1.2 Difuse lighting 2](#_Toc71382939)

[1.3 Combining effects 2](#_Toc71382940)

[2 Model Transparency 3](#_Toc71382941)

[3 Outcome 4](#_Toc71382942)

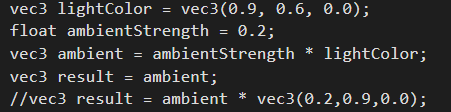
[Sources 5](#_Toc71382943)

1. Basic Lighting

To modify the appearance of models we can use lighting. In this case ambient lighting and diffuse lighting is combined into one shader that provides basic lighting effect. Following lines of code is written inspired by article provided by (Learn OpenGL Basic Lighting).

* 1. Ambient lighting

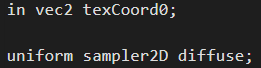
In picture 1 light color (lightColor) has been declared and RGB values are assigned. Each value (Red, Green, Blue) can be from 0 to 1. Strength of the effect can be modified through the (ambientStrength) value. This value multiplies the effect. This effect can be used by calling (result) value.



Picture 1. Ambient lighting.

* 1. Difuse lighting

Diffuse lighting is created using texture coordinates and (diffuse) sampler. See declaration for both attributes in picture 2.



Picture 2. Declarations for sampler and Texture coordinates.

Using texture2D functionality we can apply diffuse effect using texture coordinates. Color values are stored in (dColor) attribute for later use, see picture 3.



Picture 3. Diffuse lighting using texture2D functionality.

* 1. Combining effects

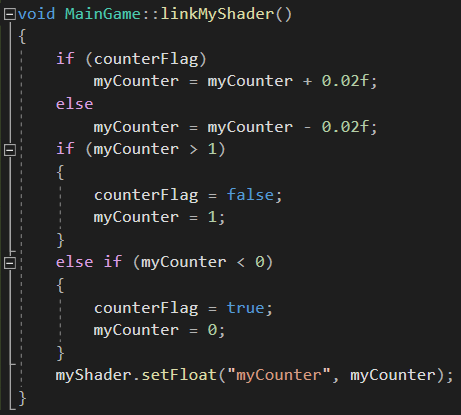
Ambient lighting and diffuse lighting effects can be combined into together to produce more complex/polished lighting effect. Combining effect is done in this case is done simply by adding diffuse and ambient color value to together (picture 4).



Picture 4. Combining effects.

1. Model Transparency

Uniform value (myCounter) is created to be able to make glowing effect. Counter changes its value between 0 and 1 in each call. Value is increased until it reaches the maximum value and then effect is reversed until it reaches minimum value. See picture 5.



Picture 5. Logic behind the “myCounter” value.

After “myCounter” is declared as uniform (float), it can be used as transparency value in Fragment color. See picture 6.



Picture 6. Using “myCounter” as transparency value.

1. Outcome

The result can be seen in the video attachment.



Sources

Learn OpenGL. Basic Lighting. Referenced 08.05.2021

Available <https://learnopengl.com/Lighting/Basic-Lighting>