IMAT2908 CW: Lighting

|  |  |
| --- | --- |
| **Full Name: Robert Quick** | **P\_Number: P2658455** |
| **GitHub Username: rbtq** | **Word Count:** |

## Introduction

A statement of the reports purpose

The purpose of this report is to show the different effects of attenuation factors, shininess, light intensity and material types on a teapot and a surface plane as well as provide a brief description of the shader used to create the result and the implementation process in which this shader was implemented.

## Brief History of shaders

### Pre-1980s

Gouraud shading was one of the first shading techniques published in 1971 by Henri Gouraud in which ‘a surface represented by a patch is approximated by polygonal planar facets. Gouraud computes information about the curvature of the surface at each vertex of each of these facets’. (Phong, 1975)

### 1980s-2000

In the 1980s, developing computer graphics was a pain; every hardware needed its own custom software(Hergaarden, 2011). This meant developing computer graphics took a lot of time and resulted in a program that could only run on specific hardware, limiting code re-useability.

## An illustrated explanation of the theoretical principle of full Phong shading

History of shaders

History of phong shading

How to make a phong shader – formula’s

Real world examples

## Methodology

## An annotated explanation of the sections of program code specifically needed to produce full Phong Shading including the structures and the functions

Talk about all code changes, structures and functions

### Structures & Functions

Below is an explanation of all the structures and functions I have added to the program, split into categories based on the file they are implemented in.

#### phong.vert

Phong.vert calculates the model’s position on-screen using the values stores in the InputData structure below.

##### InputData

Text

Description automatically generated

This structure contains all the values that need be set in the main program for the vertex shader to work.

##### ProcessData

Text

Description automatically generated

This structure caches all the values that would be calculated more than once when the vertex shader is set up to reduce the number of operations.

##### LightData

Text

Description automatically generated

This structure contains all the data calculated from the input data in the vertex shader to be used for the fragment shader.

### Phong shading

#### Diffuse

Scene with only diffuse lighting


Diffuse lighting was already implemented into the program, resulting in a teapot that looked like above.

#### Ambient

#### Specular

### Controls

## Results

Output screen captures showing different effects of attenuation factors, shininess factor, light intensity and material types on the result. This should have a proper discussion and justification

Make a table

## Conclusion

Talk about how the project went

## References

Use IEEE style (does not count towards 2000 words limit)

[Illumination for computer generated pictures | Communications of the ACM](https://dl.acm.org/doi/abs/10.1145/360825.360839)  
  
[Continuous Shading of Curved Surfaces | IEEE Journals & Magazine | IEEE Xplore](https://ieeexplore.ieee.org/abstract/document/1671906)

[3388769.3407523 (acm.org)](https://dl.acm.org/doi/pdf/10.1145/3388769.3407523) - Physically Based Shading in eory and Practice

M. Hergaarden, “Literatuur shaders - CS.VU.NL,” *Graphics shaders*, Jan-2011. [Online]. Available: https://www.cs.vu.nl/~eliens/download/literatuur-shaders.pdf. [Accessed: 22-Mar-2023].

B. T. Phong, “Illumination for computer generated pictures,” *Communications of the ACM*, 01-Jun-1975. [Online]. Available: https://dl.acm.org/doi/10.1145/360825.360839. [Accessed: 22-Mar-2023].