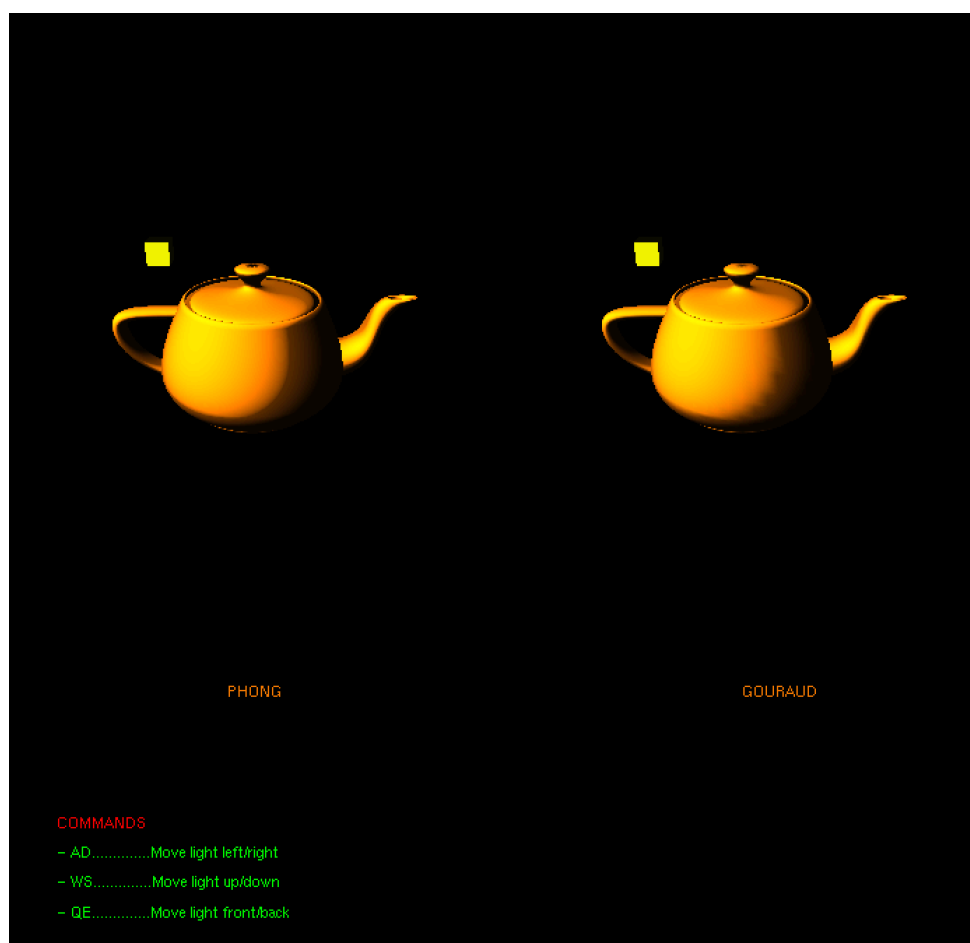




# Projeto - meta 3: Shaders

*Relatório realizado no âmbito da unidade curricular Computação Gráfica, inserida no plano de estudos do curso de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra*



Dylan Gonçalves Perdigão

2018233092

[dgp@student.dei.uc.pt](mailto:dgp@student.dei.uc.pt)

# Introdução

Neste projeto foi implementado uma comparação entre dois modelos de shading: O modelo de Phong e o modelo de Gouraud. A janela do programa é dividida em 3 viewports:

- Um para a representação do modelo de Phong.
- Outro para a representação de modelo de Gouraud.
- Um ultimo que permite apresentar os comandos relativos à luz.

Temos uma luz pontual representada por um cubo amarelo que é possível mover pelas três dimensões do espaço com ajuda das seguinte teclas

- A e D permitem deslocar a luz respetivamente para a esquerda e para a direita.
- W e S permitem deslocar a luz respetivamente para cima e para baixo.
- Q e E permitem deslocar a luz respetivamente para frente e para trás.

Os principais viewports (para modelo de Phong e Gouraud) vão desenhar exatamente os mesmo objetos (ou seja o cubo da luz e uma chaleira) mas mudará o modelo de shading aplicado.

## Componentes

O projeto está dividido em vários ficheiros permitindo uma melhor organização do trabalho.

### **COLORS.H**

É somente composto por definições de cores recorrendo a defines.

### **CUBE.H**

Implementa o desenho de um cubo por vertex array de tamanho definido pela parâmetro size. É efetuado o desenho de 6 polígonos que representam as respetivas faces do cubo. Tendo cada face um vetor normal associado.

### **READER.H**

Implementa a leitura de um ficheiro com texto, no nosso caso servirá para ler os ficheiros GLSL de shaders. É calculado o comprimento do ficheiro para efetuar a alocação de memória correta. Isto permite armazenar numa variável o texto do ficheiro GLSL.

### **MAIN.CPP**

O programa vai criar, compilar, linear e usar os ficheiros GLSL e passar o valor de variáveis "uniform" como a posição da luz, do observador, um booleano que permite saber se é aplicado o modelo de Phong ou o modelo de Gouraud e constantes para o calculo da intensidade da luz. Neste ficheiro é efetuado também o handling das teclas para o deslocamento da luz e o desenho dos objetos e textos relativos a cada um dos viewports mencionados anteriormente.

### **VERTEXSHADER.GLSL**

O VertexShader ocupa-se de calcular a cor nos vertices para o modelo de Gouraud.

### **FRAGMENTSHADER.GLSL**

O FragmentShader ocupa-se de calcular a cor nos fragmentos para o modelo de Phong.

## Calculo da cor

Para o calculo da cor são usadas variáveis recebida do main que são variáveis "uniform" que são passadas tanto para o VertexShader como para o FragmentShader. Consoante o modelo escolhido é calculado no ficheiro adequado os vetores da luz, de reflexão e do observador. O vetor normal apenas é definido nos vertices (ou seja no VertexShader) e é passado por "varying" para o ficheiro FragmentShader.

A intensidade ambiente é definida por uma constante arbitrária. A intensidade difusa é definida pelo produto escalar positivo entre vetor normal (N) e vetor de luz (L). A intensidade especular é definida pelo produto escalar positivo entre vetor de reflexão (R) e vetor do visualizador (V) sendo esse produto elevado a uma constante arbitrária. A soma dessas intensidade é usada para multiplicar pela cor. Sendo o resultado diferente se calculado nos vertices o nos fragmentos.