

# Relatório CG - Meta 3 Shaders

Duarte Dias 2018293526 20/21

Opção escolhida - Alternativa 1 (Implementação de Gouraud e Phong)

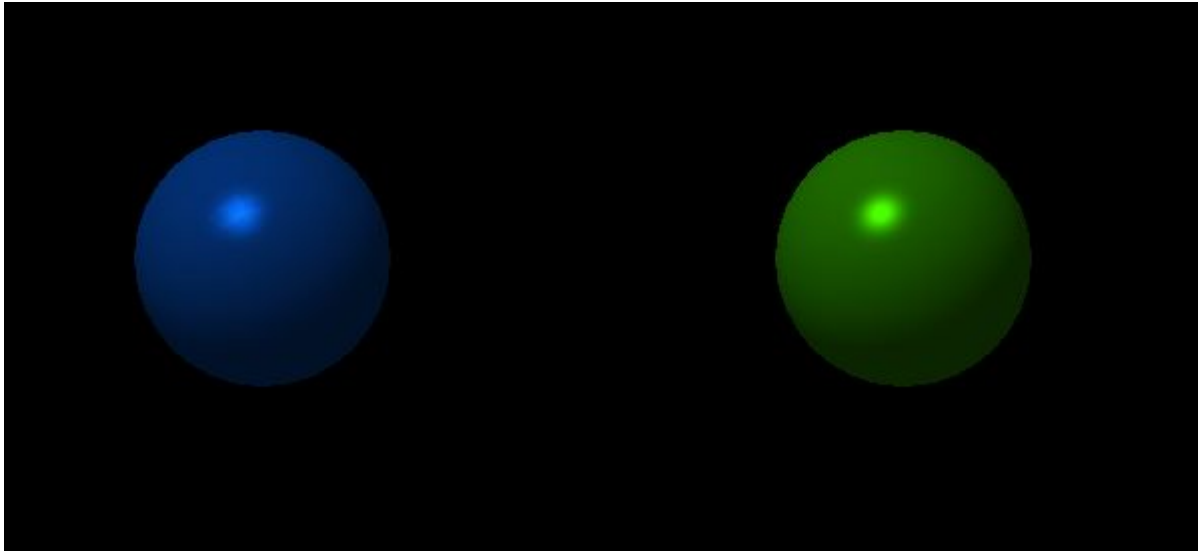


Fig 1. Método de Gouraud (Azul) e Phong (Verde)

## Phong e Gouraud

A diferença principal entre estes 2 métodos é que um no método de gouraud faz-se a interpolação da cor. Por outro lado, no método de Phong é interpolada a normal. Desta forma para o método de Gouraud fazemos o cálculo da cor nos vértices (usando o *Vertex Shader*) e no *Fragment Shader* apenas fazemos a interpolação da cor.

No método de Phong passa-se a normal a partir do *Vertex Shader* para o *Fragment Shader* usando uma variável do tipo *varying*. No *Fragment Shader* faz-se o cálculo da cor tendo em conta essa normal.

## Cálculo da cor

A diferença principal entre os 2 métodos é onde se faz o cálculo da cor, e por isso, o cálculo da cor é análogo (tendo em conta que no método de Phong a normal é passada a partir do *Vertex Shader*).

Assim, e de forma a facilitar a compreensão, foi seguido o slide 110 da T6 sobre o cálculo da luz.

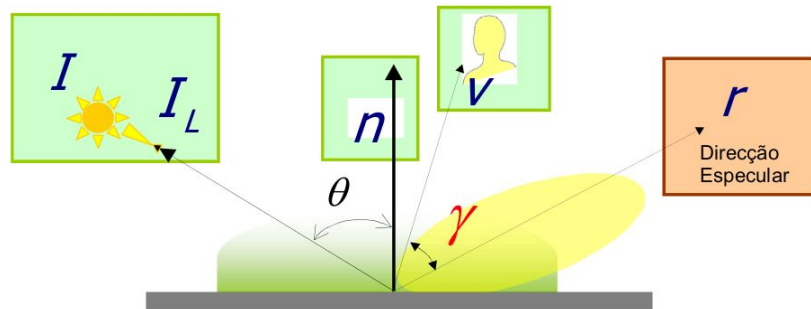


Fig.2 As várias componentes da Luz

```

N = normalize(gl_Normal);
I = normalize(I);

vpos = vec3(gl_Vertex) / gl_Vertex.w; // Getting the position of the vector in 3 space

//Ambient
ambient = mat[0] * 0.4;

//Difuse
diffuse = mat[1] * max(dot(I, N), 0.0);

//Specular
V = normalize(vpos - userPos); // Getting a vector from user to point
R = reflect(I,N);
specular = mat[2] * pow(max(dot(R, V), 0.0),mat[3]);

//Calculating the color
gl_FrontColor = (ambient + diffuse + specular)* matP * lightP;

```

Fig 3. Cálculo da Luminosidade e cor no modelo de Gouraud (GouraudV.glsl)

Onde as variáveis no gráfico representam o mesmo do que a figura acima.

As propriedades **mat** representam os coeficientes ambiente, difuso e especular, bem como a shininess do objeto. A propriedade **lightP** representa a cor da luz, neste caso, para ambos os modelos deixada a branco (1,1,1); A título de exemplo, foi ainda assumido que as todas as componentes do objeto, ambiente, especular e difusa, tinham a mesma cor, denota pela variável **matP**.

Por fim, **vpos** e **userPos** representam as posições do vetor e do utilizador. A posição do utilizador é passada através de uma variável varying.

## Movimento da Luz

Para movimento a luz a variável **I** é composta através de uma variável do tipo *uniform* que é definida no OpenGL(main.cpp) como o vetor que define a luz direcional. Desta forma para movimentar a luz, basta alterar essa variável usando o teclado, a partir do OpenGL.

## Perspectiva

Por fim, é ainda possível mudar a perspectiva e mexer a mesma a partir do main.cpp, usando o teclado.