



Centro de Informática – João Pessoa
Disciplina: Computação Gráfica
Klismann de Oliveira Barros - 20200085284
Thaís Gabrielly Marques - 20180135293

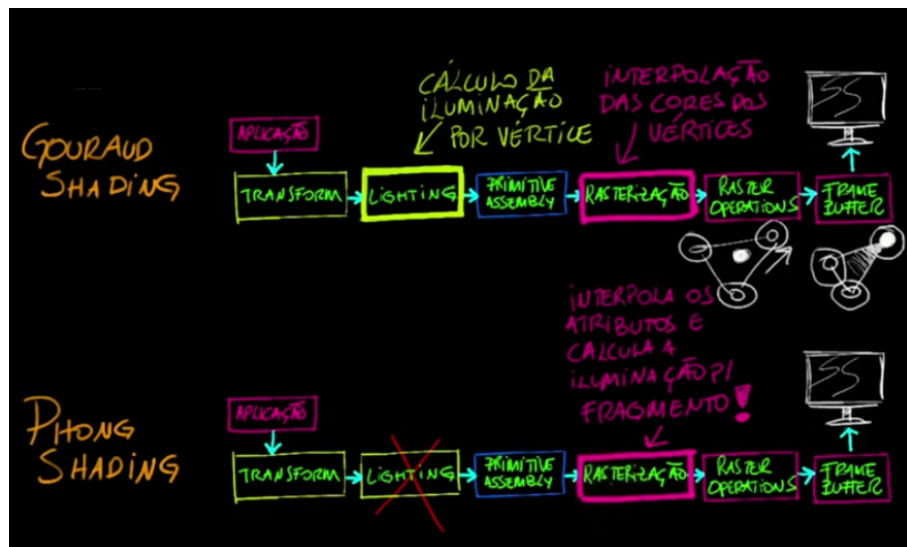
ATIVIDADE PRÁTICA 3 - IMPLEMENTAÇÃO DE MODELOS DE ILUMINAÇÃO

Descrição:

Na atividade foi pedido a implementação do modelo local de iluminação de Phong, em duas formas, primeiro utilizando interpolação de Gouraud e segundo utilizando a interpolação de Phong. O resultado esperado era que o torus renderizada em vermelho assumisse a cor azul e as interpolações pedidas em cada exercício.

Estratégias adotadas:

Seguindo como determinado, teríamos que alterar o vertex e o fragment shaders do template fornecido. Então começamos revendo as aulas fornecidas pelo professor, para revisar os conceitos e sanar algumas dúvidas na estrutura dos problemas fornecidos.



Fonte: Aula 18 - Modelo de Iluminação de Phong

Entendendo que, geralmente na interpolação de Gouraud o vertex shader é responsável por transformar os vértices, do espaço objeto para o espaço de recorte e

também fica responsável pela avaliação do modelo de iluminação para calcular a cor final do vértice. Já que o fragment shader fica incumbido de atribuir ao fragmento o valor da cor interpolada.

No entanto, na interpolação de Phong, temos diferenças, o vertex shader apenas fica incumbido de transformar os vértices, do espaço objeto para o espaço de recorte. E é no fragment shader que a avaliação da iluminação acontece.

Exercício 1: Implementação do Modelo de Phong usando Interpolação Gouraud

Com o template tínhamos o ponto de partida para começar as implementações, primeiro teríamos que usar o Gouraud para avaliar o modelo de Phong, dentro do vertex shader. O professor disponibilizou as principais variáveis para facilitar o nosso trabalho.

A estratégia adotada nesse modelo, foi fazer as operações necessárias dentro dos vetores *ambient*, *diffuse* e *specular*, e depois, como pedido no template, delimitamos guardar em **I** o final gerado pelo modelo de iluminação.

```
// Gouraud Shading (interpolação por vértice).

vec3 ambient = k_a*Ip_ambient_color;

vec3 diffuse = Ip_diffuse_color * k_d * max(vec3(0.0,0.0,0.0),
dot(N_cam_spc, L_cam_spc));

vec3 specular = Ip_diffuse_color * k_s * pow(max(vec3(0.0,0.0,0.0),
dot(R_cam_spc, normalize(-L_cam_spc))), vec3(3.0,3.0,3.0));

I = vec4(ambient + diffuse + specular, 1.0);
```

Exercício 2: Implementação do Modelo de Phong usando Interpolação Phong

Neste exercício teríamos que avaliar por cada fragmento, movendo os cálculos de iluminação do vertex para o fragment shader. Deixando no vertex apenas a transformação do espaço objeto para o espaço de recorte.

```
//-----
//-----
// Vertex Shader
```

```

//-----
material.vertexShader = `
// 'uniforms' contendo informações sobre a fonte de luz pontual.

uniform vec3 Ip_position;
uniform vec3 Ip_ambient_color;
uniform vec3 Ip_diffuse_color;
uniform vec3 k_a;
uniform vec3 k_d;
uniform vec3 k_s;
varying vec3 N_cam_spc;
varying vec3 R_cam_spc;
varying vec3 L_cam_spc;

// Programa principal do Vertex Shader.

void main() {

    vec4 Ip_pos_cam_spc = modelViewMatrix * vec4(Ip_position, 1.0);

    vec4 P_cam_spc = modelViewMatrix * vec4(position, 1.0);

    N_cam_spc = normalize(normalMatrix * normal);

    L_cam_spc = normalize(Ip_pos_cam_spc.xyz - P_cam_spc.xyz);

    R_cam_spc = reflect(L_cam_spc, N_cam_spc);

    gl_Position = projectionMatrix * P_cam_spc;
}
`;

//-----
// Fragment Shader
//-----
material.fragmentShader = `

```

```

    varying vec3 N_cam_spc;
    varying vec3 R_cam_spc;
    varying vec3 L_cam_spc;

    uniform vec3 k_a;
    uniform vec3 k_d;
    uniform vec3 k_s;

    uniform vec3 Ip_position;
    uniform vec3 Ip_ambient_color;
    uniform vec3 Ip_diffuse_color;

    // 'uniforms' contendo informações sobre as reflectâncias do
objeto.

    // Programa principal do Fragment Shader.

void main() {

    vec3 ambient = k_a*Ip_ambient_color;

    vec3 diffuse = Ip_diffuse_color * k_d *      max(vec3(0.0,0.0,0.0),
dot(N_cam_spc, L_cam_spc));

    vec3 specular = Ip_diffuse_color * k_s * pow(max(vec3(0.0,0.0,0.0),
dot(R_cam_spc, normalize(-L_cam_spc))), vec3(3.0,3.0,3.0));

    // 'gl_FragColor' : variável de sistema que conterà a cor final
do fragmento calculada pelo Fragment Shader.

    gl_FragColor = vec4((ambient + diffuse + specular), 1.0);
}

;

//-----
-----

```

Algumas falhas durante o processo:

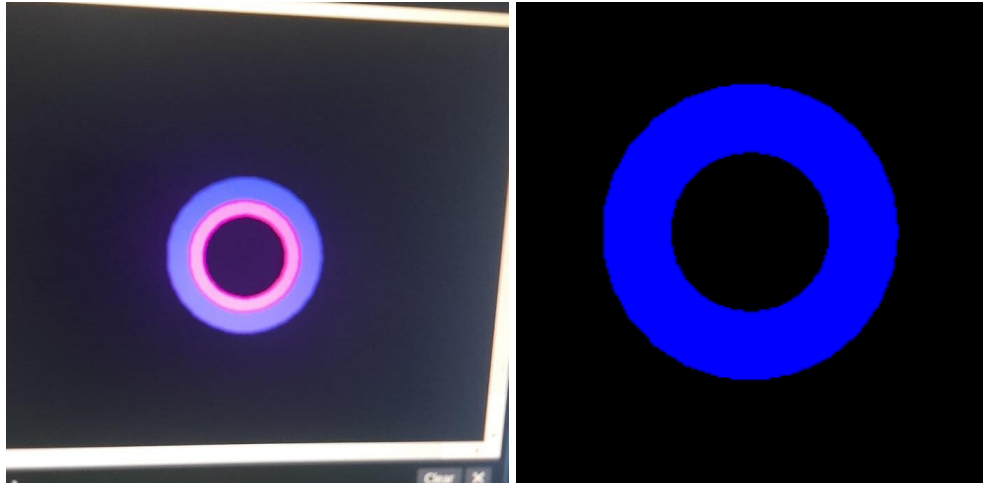


Imagem 02 - falhas na avaliação utilizando interpolação Gouraud

Resultados gerados:

Exercício 1:

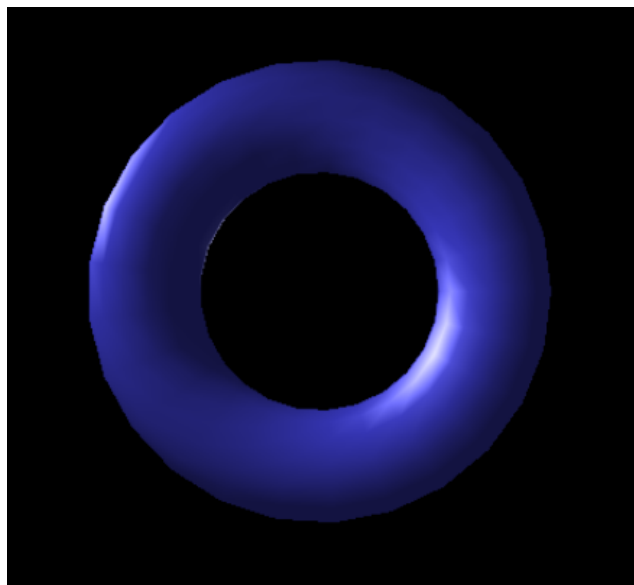


Imagem 03 - Resultado final gerado pela Implementação do Modelo de Phong usando Interpolação Gouraud.

Exercício 2:



Imagem 04 - implementação do Modelo de Phong usando Interpolação Phong

Dificuldades e possíveis melhorias:

Tivemos um conjunto de dificuldades durante as implementações, no primeiro exercício o Gouraud Shading apresentou falhas como: código não gerando nada e depois que resolvemos esse primeiro bug, ele ficou apresentando a imagem relatada nas falhas, até que conseguíssemos ajustar. No segundo exercício tivemos dificuldades na interpolação por fragmento, que ficou um pouco similar ao primeiro exercício.

Nossas melhorias futuras seriam: deixar o código mais legível na parte que implementamos, para melhorar a leitura. Tentar modificar para ficar mais nítido a diferença entre eles (Nesse video é possível acompanhar a diferença: https://drive.google.com/file/d/1PpFgMDW7VAHmb4SLyL_qInNbgqj4-A9m/view?usp=sharing) e implementar os modelos em outras formas, como a clássica panela e o coelho.

Referências bibliográficas:

- Notas e aulas do professor.
- <https://codepen.io/gabriellymarques02/pen/oNZYGqJ> (19/05/21-Tutorial shaders)
- <https://cglearn.codelight.eu/pub/computer-graphics/shading-and-lighting>
- <http://www.cs.toronto.edu/~jacobson/phong-demo/> (exemplos com outras formas)

Link repositório:

- <https://codepen.io/gabriellymarques02/pen/dyvVdRo?editors=0011> (Exercício 1: Implementação do Modelo de Phong usando Interpolação Gouraud)
- <https://codepen.io/gabriellymarques02/pen/mdWBjWX> (Exercício 2: Implementação do Modelo de Phong usando Interpolação Phong)