



LUCAS SIMÕES DE ALMEIDA

MATRÍCULA - 1712101

TAREFA 3.1

MAPEAMENTO DE RUGOSIDADE

Relatório da Disciplina Computação
INF1761, Segundo Semestre do
ano de 2021.

Professor: Waldemar Celes

RIO DE JANEIRO

2021

Nesta tarefa, foi pedido a aplicação de um mapeamento de rugosidade em uma esfera, simulando a aparência de uma bola de golfe, esta tarefa partindo da conclusão do projeto 2 foi mais tranquila que o esperado, pois maior parte do código da main permaneceu intacto.

O problema surge realmente nas criações de novos shaders, afinal para mapear a rugosidade é necessário calcular novas maneiras de iluminação.

```
1 #version 460 core
2
3 layout(location = 0) in vec4 pos;
4 layout(location = 1) in vec3 normal;
5 layout(location = 2) in vec2 texcoord;
6 layout(location = 3) in vec3 tangent;
7
8
9 uniform vec4 leye; //light pos in eyespace
10 uniform mat4 mvp;
11 uniform mat4 inv;
12
13
14 out data {
15     vec3 ltan;
16     vec3 vtan;
17     vec2 texcoord;
18 }v;
19
20
21 void main(void)
22 {
23     vec3 t = normalize(tangent);
24     vec3 n = normalize(normal);
25     vec3 b = normalize(cross(normal,tangent));
26     mat3 base = transpose(mat3(t,b,n));
27
28     vec4 eye = inv * vec4(0.0f,0.0f,0.0f,1.0f);
29     v.vtan = base * normalize(vec3(eye-pos));
30     vec4 lobj = inv * leye;
31     v.ltan = base * normalize(vec3(lobj-pos));
32     v.texcoord = texcoord;
33     gl_Position = mvp * pos;
34 }
35
```

Analisando as aulas e os slides, consegui desenvolver os shaders sem problemas, mesmo assim, a minha esfera não recebia o seu mapeamento.

Eu percebi eventualmente que o problema se encontrava na Vao da esfera, pois não havia passado a tangente dela para o novo shader, então modifiquei a função da esfera para informar a tangente ao shader.

```
static GLuint createSphere(int nx, int ny)
{
    int csize = 3 * (nx + 1) * (ny + 1);
    int tsize = 2 * (nx + 1) * (ny + 1);
    float* texcoord;
    float* tangent;
    float* coord = sphereCoords(nx, ny, &texcoord,&tangent);
    int isize = 6 * nx * ny;
    unsigned int* index = sphereIncidence(nx, ny);
    GLuint vao = CreateVAO();
    CreateBuffer(GL_ARRAY_BUFFER, csize * sizeof(float), (GLvoid*)coord);
    VertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 0, 0);
    VertexAttribPointer(1, 3, GL_FLOAT, GL_FALSE, 0, 0);
    CreateBuffer(GL_ARRAY_BUFFER, tsize * sizeof(float), (GLvoid*)texcoord);
    VertexAttribPointer(2, 2, GL_FLOAT, GL_FALSE, 0, 0);
    CreateBuffer(GL_ARRAY_BUFFER, csize * sizeof(float), (GLvoid*)tangent);
    VertexAttribPointer(3, 3, GL_FLOAT, GL_FALSE, 0, 0);
    CreateBuffer(GL_ELEMENT_ARRAY_BUFFER, isize * sizeof(unsigned int), (GLvoid*)index);
    free(index);
    free(coord);
    free(texcoord);
    free(tangent);
    return vao;
}
```

Resultado final:

Imagem 1 -

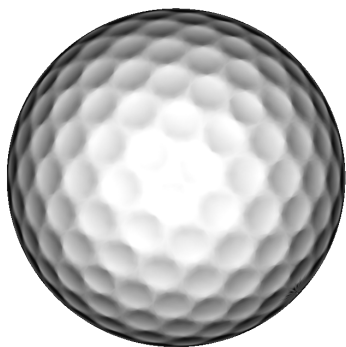
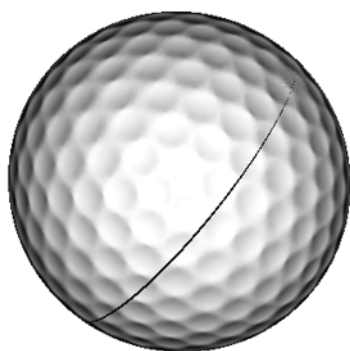


Imagem 2 -



Esse traço que aparece na imagem provavelmente se dá devido algum erro de cálculo na esfera, ou a utilização de poucos pontos para seu desenho.