Shaders - Exercícios

Filipi Nascimento Silva

1.Silhueta:

Faça um **fragment shader** que receba:

$\mathbf{N}_e$ (*v_normal)* - vetor **normal** no espaço da câmera.

$\mathbf{V}_e$ (*v_eye*) - vetor com a **direção** da **câmera**.

ε (err) - Uma **constante** do tipo float.

Se ($\mathbf{V}_e$. $\mathbf{N}_e$) < **ε** - pinta o fragmento de **preto**.

Senão pinta o fragmento de **branco**.

```
#version 150
uniform float err;

in vec3 v_normal;
in vec3 v_eye;

out vec4 fragColor;

void main(){
	vec3 normal = normalize(v_normal);
	vec3 eye = normalize(v_eye);

	if(dot(normal,eye) < err){
		fragColor = vec4(0.0,0.0,0.0,1.0);
	}else{
		fragColor = vec4(1.0,1.0,1.0,1.0);
	}
}
```

2.Billboard:

Faça um **vertex shader** que sempre apresente o **modelo 3D voltado** para a **camera** (**billboard**).

3.Renormalizar a normal:

Por que devemos **renormalizar** a **normal** no fragment shader para ter a **luz** (per pixel) corretamente calculada?

4.Displacement vs Bumping Mapping:

Em qual tipo de shader a técnica de **Displacement mapping** deve ser implementada? E quanto à técnica de **bumping mapping**? Discuta sobre a **performance** vs **qualidade** dessas técnicas no **contexto** de **shaders**.