NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

Iluminação e sombreamento

CGI – 1º Semestre 2022/2023

Realizado por:
Guilherme Poças – 60236
João Oliveira – 61052

Tipos de dados usados

Para representar uma luz no lado da aplicação JavaScript foi usada a classe *lightClass*, com os seguintes atributos:

- Boolean on – Indica se a luz está, ou não, ligada.
- Int type – O tipo de luz, pontual, direcional ou spotlight (correspondem a valores de 0, 1 e 2).
- Vec3 ambiente – Intensidade da luz ambiente.
- Vec3 diffuse – Intensidade da luz difusa.
- Vec3 specular – Intensidade da luz especular.
- Vec4 position – Coordenadas da luz no mundo.
- Vec3 axis – Vetor direção da luz.
- Float aperture – Ângulo em graus de abertura do feixe de luz.
- Float cutoff – Parametro que atenua a intensidade do feixe de luz à medida que nos afastamos do seu centro.

No *fragment shader* é utilizada a *struct LightInfo* com os mesmos parâmetros.

Nova Fonte de Luz

Para adicionar uma nova fonte de luz será necessário apenas adicionar um novo objeto *lightClass* ao array *lights*. Caso o número de luzes ultrapasse o máximo (*MAX_LIGHTS*) definido no *fragment shader*, então também teremos de aumentar o seu valor.

Implementação do Spotlight

O spotlight foi implementado com o seguinte código no *fragment shader*:

```
if(uLights[i].type == SPOTLIGHT) {
    float angle = acos(dot(L ,normalize(-uLights[i].axis)));
    if(angle < uLights[i].aperture) {

        vec3 Iinc = ambientColor + diffuse + specular;
        float c = pow(cos(angle), uLights[i].cutoff);
        I += vec3(mix(0.0, Iinc.x, c),
         mix(0.0, Iinc.y, c),
         mix(0.0, Iinc.z, c));
    }
}
```

Onde a variável *angle* é o ângulo do vetor L com o vetor oposto do feixe de luz, e apenas se este ângulo for menor que o ângulo do feixe de luz é que a intensidade da luz é incrementada.

Implementação do desafio

```
let drag = false;
var old_x, old_y;
var dX = 0, dY = 0;
let mousedown = function (e) {
    drag = true;
    old_x = e.pageX;
    old_y = e.pageY;

    return false;
};

let mouseup = function (e) {
    drag = false;
};

let mousemove = function(e){
    if (!drag) return false;
    dX = (e.pageX - old_x)/5;
    dY = (e.pageY - old_y)/5;

    old_x = e.pageX, old_y = e.pageY;

    let r = rotate(-dY, cross([0, 1, 0], camera.eye))
    camera.eye = mult(r, vec4(camera.eye, 1))
    camera.eye = (mult(rotateY(-dX), vec4(camera.eye, 1)))
}
```

Para ser possível rodar a cena com o rato, cada vez que o utilizador carrega no ecrã, guardamos a posição do rato. Ao mover o rato, vemos em que direção ele o moveu, se foi para cima, ou seja, em Y, ou para o lado, em X.

Para calcular a rotação que é preciso fazer no eixo dos x e z, usamos a função "rotate" da biblioteca MV.js onde damos a posição em que o rato se mexeu, cima ou baixo e calculamos o produto externo entre o Y e a direção da camera, o "eye". Depois é só multiplicar essa matriz de rotação pelo "eye" atual.

Para calcular a rotação em y, é mais simples, por ser apenas um eixo. Usamos a função "rotateY", indicando se é uma rotação para a esquerda ou direita, e o nosso "eye".

Funcionalidades Extra

Foi implementada a seguinte funcionalidade extra:

- É possível mudar o tipo da cada luz entre pontual, direcional ou spotlight, e ao mudar para spotlight o GUI adiciona novas definições para alterar os parâmetros relevantes (axis, aperture e cutoff);