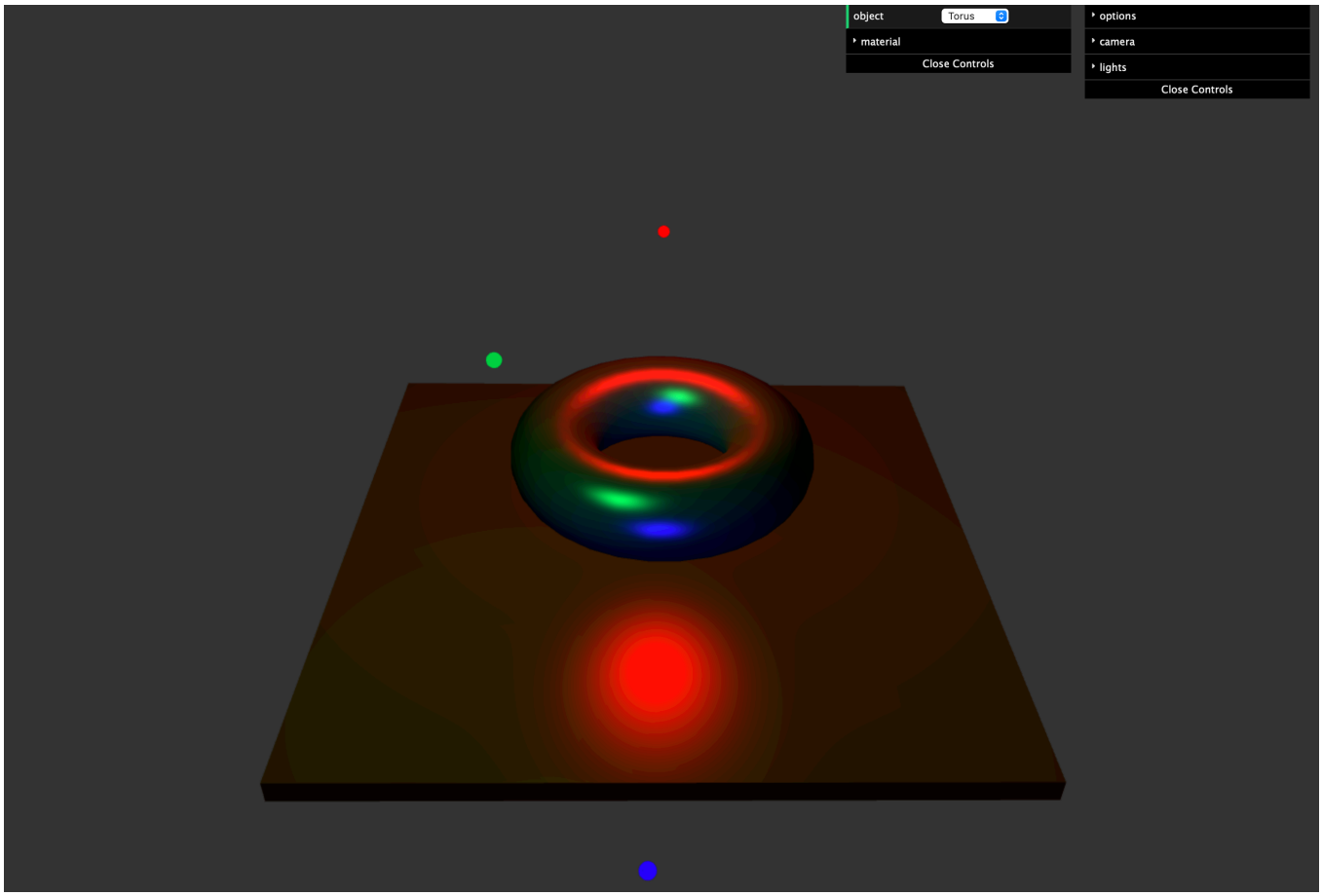
Projecto 03 - Light Studio



Changelog

- **2021/12/23 09h00** - Publicação da versão inicial do enunciado.

Introdução

Neste projecto vamos criar uma aplicação que permite visualizar objectos sob o efeito de, pelo menos, 1 fonte de luz, através duma câmara com uma projeção perspectiva.

A iluminação dos objetos será implementada recorrendo ao modelo de iluminação de Phong, avaliado no referencial da câmara. A iluminação deverá ser calculada ao nível de cada fragmento.

A cena é constituída por:

- um objecto primitivo, dos fornecidos na pasta `libs`, com uma transformação de modelação nula (matriz identidade).
- Um cubo, deformado num paralelepípedo com as dimensões de 3 x 0.1 x 3, transformado de modo a que a face superior fique em y=-0.5. O material deste cubo é fixo e as suas propriedades podem ser escolhidas livremente por si.
- 1 ou mais luzes controladas pelo utilizador usando uma interface implementada usando a biblioteca `dat.gui`.

É forçoso que **atualizem a vossa pasta `libs` com o conteúdo mais recente do repositório**, visto alguns objectos terem sofrido pequenas correções para que a iluminação funcione.

O utilizador poderá ainda:

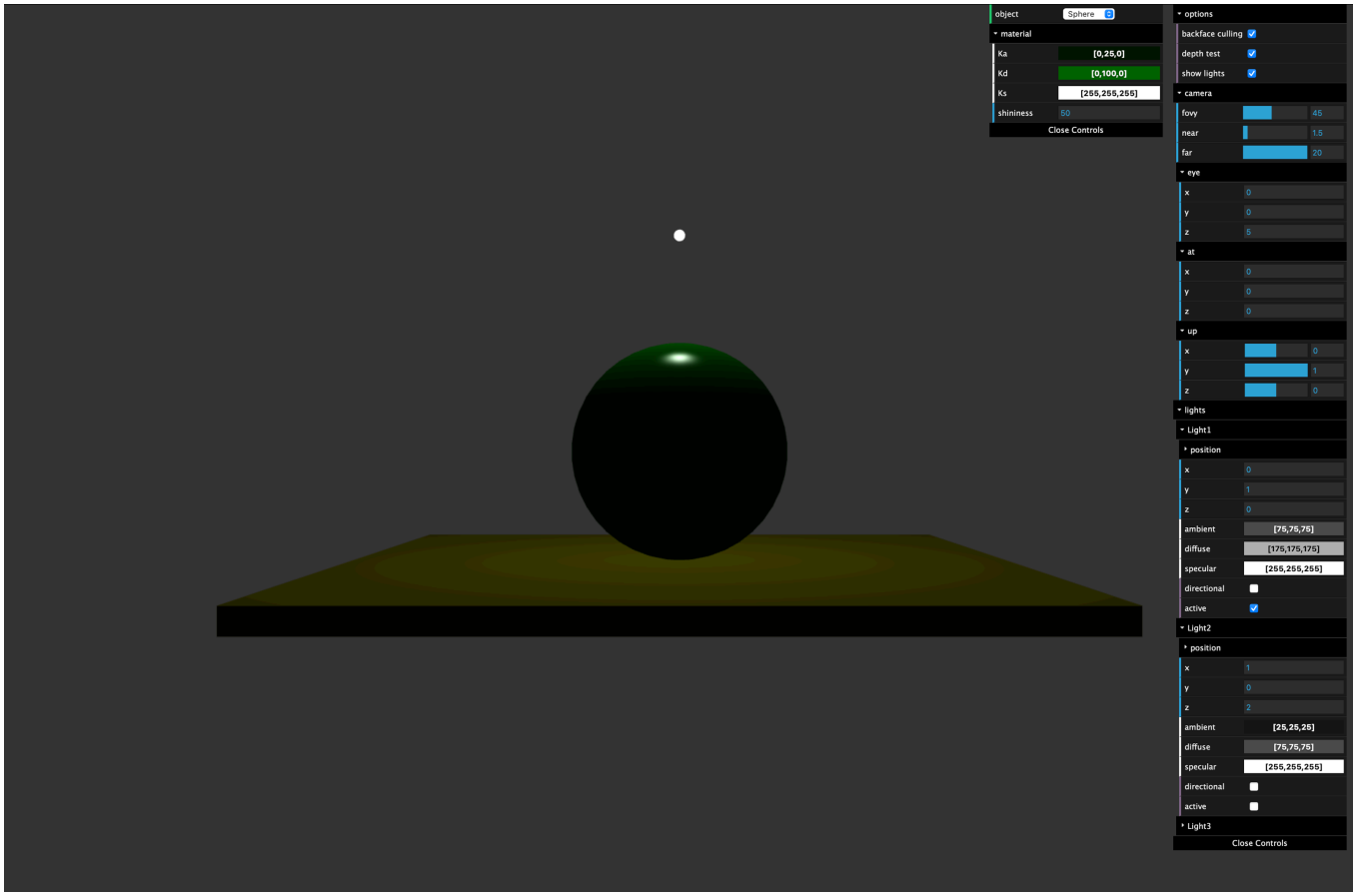
- ligar/desligar o método de remoção de faces ocultas conhecido pelo método do produto interno (ou *back face culling*);
- ligar/desligar o método de remoção de faces ocultas conhecido pelo nome de *z-buffer* (ou depth buffer);
- ligar/desligar a visualização das localizações dos pontos de luz (desenhados como pequenas esferas em wireframe e sem iluminação);
- escolher, em qualquer altura, qual o objeto que pretende visualizar (esfera, cilindro, torus, pirâmide ou cubo);
- alterar as características do material aplicado ao objeto (*Ka*, *Kd*, *Ks* e *shininess*);
- cada fonte de luz poderá ser ligada/desligada, parametrizada para ser pontual ou direcional. Se for pontual o utilizador pode controlar a sua posição na cena (em World Coordinates). Se for direcional, o utilizador pode definir a direção dessa mesma luz (na realidade definirá um vetor com a direção oposta, apontando da cena para o local de onde a luz provém);
- A assinatura espectral de cada fonte de luz também deverá poder ser modificada (intensidades r, g e b para cada um dos termos do modelo de

iluminação - ambiente, difusa e especular)

- manipular a posição e orientação da câmara, de forma semelhante à pedida nos exercícios da aula prática 10, bastando contudo utilizar os sliders da interface e não sendo necessário qualquer mecanismo baseado em gestos executados com rato sobre o canvas.

Nota: Caso suporte mais do que uma fonte de luz, a sua interface deverá ser criada de tal forma que, acrescentar uma fonte de luz nova no seu programa traduzir-se-á apenas na adição de mais um objeto (com as propriedades da fonte de luz) a um vetor que guarda todas as luzes da cena.

A imagem seguinte mostra uma possível interface para o programa. Note que a interface apresentada não escala bem com o número de fontes de luz.



Redimensionamento da janela

O utilizador pode mudar as dimensões da janela do browser, fazendo com que a relação de aspeto se altere. A relação de aspeto da câmara será exatamente a mesma

que a da janela do browser. Por essa razão não há qualquer necessidade expor na interface o parâmetro *aspect* da câmara perspetiva.

Limites do programa e especificações técnicas

O número máximo de luzes suportadas pelo seu programa deverá ser uma constante que deverá aparecer exatamente 2 vezes no seu código, concretamente, no ficheiro javascript da aplicação e no *fragment shader* do programa GLSL responsável pela iluminação.

Sugere-se a inclusão das seguintes declarações no *fragment shader*:

```
const int MAX_LIGHTS = 8;

struct LightInfo {
    vec3 pos;
    vec3 Ia;
    vec3 Id;
    vec3 Is;
    bool isDirectional;
    bool isActive;
};

struct MaterialInfo {
    vec3 Ka;
    vec3 Kd;
    vec3 Ks;
    float shininess;
};

uniform int uNLights; // Effective number of lights used

uniform LightInfo uLight[MAX_LIGHTS]; // The array of lights present in
the scene
uniform MaterialInfo uMaterial;  // The material of the object being
drawn
```

A novidade aqui é o agrupamento de várias variáveis de tipo **uniform** em estruturas. Do lado do javascript, poderemos passar valores para estas estruturas **uniform** campo, a

campo, usando os seus nomes. Por exemplo, podemos escrever:

```
const uShininess = gl.getUniformLocation(program, "uMaterial.shininess");
const uKaOfLight0 = gl.getUniformLocation(program, "uLight[0].Ia");
```

Quem optar por implementar apenas uma fonte de luz, deverá substituir o array `uLight` do exemplo acima por uma variável única. Contudo, a implementação de várias luzes será valorizada, bem como a facilidade em as acrescentar, mudando o menos possível o código da aplicação.

Movimento das fontes de luz

O movimento das fontes de luz no video desta página é apenas decorativo, não sendo necessário implementar tal funcionalidade.A figura mostra a alteração necessária na geração dos pontos da grelha:

Avaliação

- A definir brevemente

Entrega, prazos e penalizações

A entrega do trabalho deverá ser efetuada no Moodle até ao final do prazo limite (domingo, 19 de dezembro de 2021, 23:59). O Moodle aceitará trabalhos entregues fora de prazo com a penalização de 2 valores por cada dia de atraso.

This article was updated on dezembro 10, 2021

Fernando Birra

← Previous
Sessão 10 - Projeção Perspetiva