Universidade Nova de Lisboa

Faculdade de Ciências e Tecnologia

**Projeto 3:**

Uma imagem com chávena, chávena de café

Descrição gerada automaticamente**Modelo de iluminação de Phong com sombreamento de Phong**

Realizado por:

Guilherme Fernandes nº 60045

Joana Wang nº 60225

**Índice**

1. Objetivo........................................................................................ 3

2. Representação das luzes............................................................. 3

2.1. Javascript.............................................................................. 3

2.2. Fragment shader................................................................... 4

3. Implementação de >3 luzes.......................................................... 4

4. Implementação do spotlight.......................................................... 4

5. Implementação do desafio............................................................ 5

**1. Objetivo**

Este projeto consiste na criação de uma aplicação onde permite visualizar objetos sob iluminação de várias fontes de luzes, através de uma câmara numa projeção perspetiva.

**2.Representação das luzes**

Temos três tipos de luz: pontual, direcional e spotlight. Cada um destes pode ser ligado ou desligado através da interface, assim como podem ser manipuladas os seus valores das variáveis.

**2.1. Javascript**

A representação das três luzes no javascript é através de um vetor, um em cada posição.

Para cada luz são guardados vários parâmetros:

- on: indica se a respetiva luz está ou não ligada;

- position: posição da luz nas coordenadas da câmara;

- lightAmb: intensidade (r, g, b) da fonte de luz do modelo de iluminação ambiente;

- lightDif: intensidade (r, g, b) da fonte de luz do modelo de iluminação difusa;

- lightSpec: intensidade (r, g, b) da fonte de luz do modelo de iluminação especular;

Os restantes parâmetros são apenas considerados para o tipo de luz spotlight, sendo ignorado pelos restantes tipos de luz:

- axis: direção central para onde o spotlight aponta;

- apertura: ângulo de abertura (em graus) que define a região cónica formado pelo spotlight;

- cutoff: decaimento da atenuação da intensidade da luz à medida que se afasta da zona central.

**2.2. Fragment shader**

O dado das luzes é apenas passado do javascript para o fragment shader. Está representado num tipo de dados struct, tal como sugerido. Todas os parâmetros referidos acima no javascript são passados para esse struct, excepto o ‘on’, permitindo efetuar os cálculos baseado no modelo de iluminação de Phong.

**3. Implementação de >3 luzes**

Tal como referido, o nosso programa está implementado com 3 luzes. Caso for um número arbitrário de luzes, definido na interface, teria de efetuar várias alterações no código:

- Da interface: acrescentar 3 blocos de interface do tipo dropdown/slider para indicar o número de luzes para cada tipo.

- Da javascript: de acordo com valor indicado na interface para cada tipo, criar um vetor com o número de luzes.

- Do shader: não necessita de alterações pois já implementa a iluminação com uNLights passados pelo javascript.

**4. Implementação do spotlight**

Passos para a implementação de luz do tipo spotlight:

1- Calcular a luz ambiente;

2- Determinar spotlightDir, i.e., a direção para onde o spotlight aponta.

3- Determinar fragmentDirToLight, i.e., a direção da luz (baseado no conceito de luz pontual) apontando da superfície à fonte de luz;

4- Calcular o ângulo **α** através de arccos do produto interno entre vetores unitários spotlightDir e fragmentDirToLight;

5- Comparar o ângulo **α** com o ângulo aperture:

- Caso for **menor**: retornar o produto do máximo entre luz ambiente e decaimento (calculado com , sendo n o valor de cutoff) com a iluminação pontual nesse ponto.

- Caso for **maior**: retornar apenas a luz ambiente.

**5. Implementação do desafio**

Passos para a implementação do controlo da posição e orientação da câmara com o rato efetuando rotações no referencial da câmara:

1. Calcular a distância que está a câmara da superfície (obtida calculando o tamanho do vetor com origem na câmara e que termina no ponto para onde a câmara está a apontar (r))
2. Calcular o angulo inicial(φ) que o vetor faz com o plano xz
3. Calcular o angulo inicial(θ) que o vetor faz com o plano zy
4. Guarda-se o deslocamento do rato enquanto o left mouse click está pressionado
5. Normaliza-se o valor do deslocamento em x em relação á largura da janela e o valor do deslocamento em y em relação á altura da janela
6. Subtrai-se a φ o valor normalizado do deslocamento em x e soma-se a θ o valor normalizado do deslocamento em y
7. Calcula-se a coordenada x da câmara, somando o produto de r pelo seno de θ e pelo cosseno de φ ao vértice para onde a câmara está a apontar
8. Calcula-se a coordenada y da câmara, somando o produto de r pelo seno de φ ao vértice para onde a câmara está a apontar
9. Calcula-se a coordenada z da câmara, somando o produto de r pelo cosseno de θ e pelo cosseno de φ ao vértice para onde a câmara está a apontar