



UNIVERSIDADE DO MINHO

CG - Fase 4 do Trabalho Prático  
Grupo nº39

Hugo Filipe de Sá Rocha  
(A96463)

Gabriel Alexandre Monteiro da Silva  
(A97363)

José Diogo Lopes Faria  
(A95255)

4 de junho de 2023

# Conteúdo

<b>1</b>	<b>Contextualização</b>	<b>3</b>
1.1	Alterações no Generator . . . . .	3
1.2	Alterações no Engine . . . . .	3
<b>2</b>	<b>Concepção da solução</b>	<b>4</b>
2.1	Coordenadas dos vetores normais . . . . .	4
2.1.1	Plano . . . . .	4
2.1.2	Caixa . . . . .	4
2.1.3	Cone . . . . .	5
2.1.4	Esfera . . . . .	5
2.1.5	Cilindro . . . . .	6
2.1.6	Teapot . . . . .	6
2.1.7	Normalização . . . . .	6
2.2	Engine . . . . .	7
2.2.1	Estrutura Luz . . . . .	7
2.2.2	Estrutura Cores . . . . .	7
2.2.3	Armazenamento das normais . . . . .	7
2.2.4	Outros . . . . .	7
2.3	Demonstração . . . . .	8
2.4	Sistema Solar . . . . .	11
<b>3</b>	<b>Conclusão</b>	<b>13</b>

# Capítulo 1

## Contextualização

### 1.1 Alterações no Generator

Nesta quarta e última fase do trabalho prático, na aplicação **generator**, adicionamos, a cada modelo, as coordenadas dos vetores normais a cada vértice para efeitos de iluminação.

### 1.2 Alterações no Engine

Quanto ao **Engine**, foi atualizado de forma a conseguir ativar vários tipos de luzes: **ponto de luz**, **direcional** e **spotlight** e também definir as componentes de cor para cada uma das propriedades da iluminação, **difusa**, **especular**, **emissiva** e **ambiente**.

## Capítulo 2

# Concepção da solução

### 2.1 Coordenadas dos vetores normais

#### 2.1.1 Plano

Todos os vértices do plano apresentam como normal  $(0,1,0)$ .

#### 2.1.2 Caixa

Visto que a caixa é composta por quatro planos: **trás**, **frente**, **cima**, **baixo**, **esquerda** e **direita**; bastou-nos aplicar as normais do plano a cada plano, isto é:

- O plano de trás apresenta como normais  $(0,0,-1)$ ;
- O plano da frente apresenta como normais  $(0,0,1)$ ;
- O plano de cima apresenta como normais  $(0,1,0)$ ;
- O plano de baixo apresenta como normais  $(0,-1,0)$ ;
- O plano da esquerda apresenta como normais  $(-1,0,0)$ ;
- O plano da direita apresenta como normais  $(1,0,0)$ ;

### 2.1.3 Cone

Os vértices que compõem a base do cone apresentam como normais o vetor  $(0,-1,0)$ . A superfície lateral do cone é composta por vértices com coordenadas:

- $px = raio\_stack\_atual * \sin(j * \alpha);$
- $py = i * stack\_height;$
- $pz = raio\_stack\_atual * \cos(j * \alpha);$

Onde  $i$  é a variável que percorre o número de **stacks** e  $j$  a variável que percorre o número de **slices**. O valor de  $\alpha$  é  $(2*\pi)/slices\_nr$ .

As normais para estes vértices são:

- $nx = \sin(j * \alpha);$
- $ny = 0;$
- $nz = \cos(j * \alpha);$

### 2.1.4 Esfera

A esfera é composta por vértices com coordenadas:

- $px = radius * \cos(\pi/2 - j * \beta) * \sin(i * \alpha);$
- $py = radius * \sin(\pi/2 - j * \beta);$
- $pz = radius * \cos(\pi/2 - j * \beta) * \cos(i * \alpha);$

Onde  $i$  é a variável que percorre o número de **slices** e  $j$  a variável que percorre o número de **stacks**. O valor de  $\alpha$  é  $(2*\pi)/slices\_nr$  e o de  $\beta$  é  $\pi/stacks\_nr$ .

As normais para estes vértices são:

- $nx = \cos(\pi/2 - j * \beta) * \sin(i * \alpha);$
- $ny = \sin(\pi/2 - j * \beta);$
- $nz = \cos(\pi/2 - j * \beta) * \cos(i * \alpha);$

### 2.1.5 Cilindro

Os vértices da base do cilindro possuem como normais o vetor  $(0,-1,0)$ . Os vértices da circunferência do topo do cilindro têm como normais  $(0,1,0)$ . A superfície lateral do cilindro é composta por vértices com as seguinte coordenadas:

- $px = radius * \sin(i * alpha);$
- $py = ratio * j;$
- $pz = radius * \cos(i * alpha);$

Onde  $i$  é a variável que percorre o número de **slices** e  $j$  a variável que percorre o número de **stacks**. O valor de **alpha** é  $(2*pi)/slices\_nr$  e o valor de **ratio** é  $height/stacks\_nr$ . As normais para estes vértices são:

- $nx = \sin(i * alpha);$
- $ny = 0;$
- $nz = \cos(i * alpha);$

### 2.1.6 Teapot

No cálculo das normais do **teapot** acabamos por ter algumas dificuldades visto os vértices do mesmo serem calculados através de operações sobre matrizes. No entanto, para um vértice do teapot genérico  $(px,py,pz)$  assumimos que a sua normal correspondente é o próprio vetor  $(px,py,pz)$  mas normalizado.

### 2.1.7 Normalização

Para garantir que as normais calculadas apresentam como norma **1**, foram todas normalizadas, isto é, dividimos cada componente da normal por  $\sqrt{nx*nx + ny*ny + nz*nz}$ .

## 2.2 Engine

Nesta aplicação foi necessário adaptar a função de **parsing** do ficheiro XML de forma a ler corretamente os dados necessários para efeitos de iluminação, isto é, quantas e quais luzes devem ser ativadas e, para cada modelo, receber as cores dos materiais para cada propriedade de iluminação. Além disso, é responsável por ler as normais de cada modelo e armazená-las em **VBOS**.

### 2.2.1 Estrutura Luz

Foi criada uma estrutura **Luz** que armazena o número da luz, um vetor **pos** que deve ser usado nos pontos de luz, um vetor **dir** que deve ser usado nas luzes direcionais, um float **cutoff** para ser usado na luz spotlight e uma flag que indica qual o tipo de luz a aplicar (ponto de luz, direcional e spotlight). Ainda nesta estrutura foram definidos dois métodos: o primeiro, **set\_indice**, coloca o número de luz respetivo somando uma variável global que é iniciada a 0 com **GL\_LIGHT0**. Após isto, o valor da variável global é incrementado para que o número da próxima luz a ser recebida seja correto. O outro método, aplica a luz através das várias variantes da função **glLightfv** dependendo do tipo de luz.

### 2.2.2 Estrutura Cores

Criámos também uma outra estrutura **Cores** que armazena um vetor **RGB** que representa a cor, um float **shininess** e uma flag que indica a propriedade de iluminação que deve ser aplicada. Assim como na estrutura Luz, foi definido um método que aplica a respetiva propriedade da iluminação através da função **glMaterialfv**.

### 2.2.3 Armazenamento das normais

As normais de cada modelo foram armazenadas em **VBOS**. Para isso, criámos um mapa que a cada nome do modelo associa um vetor de Ponto correspondente às normais desse modelo. Após isso, é necessário ativar o buffer para cada modelo e preenchê-lo com o vetor das normais.

### 2.2.4 Outros

Todas as luzes foram guardadas num vetor de Luz, onde este é percorrido na função **main** para todas as luzes serem ativadas através da função **glEnable**. Para cada modelo, criámos um vetor de Cores de forma a saber as cores a aplicar em cada modelo na função **renderScene**.

## 2.3 Demonstração

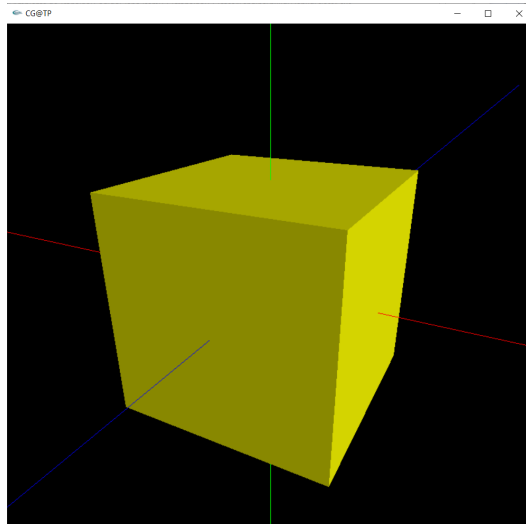


Figura 2.1: Teste 4\_1

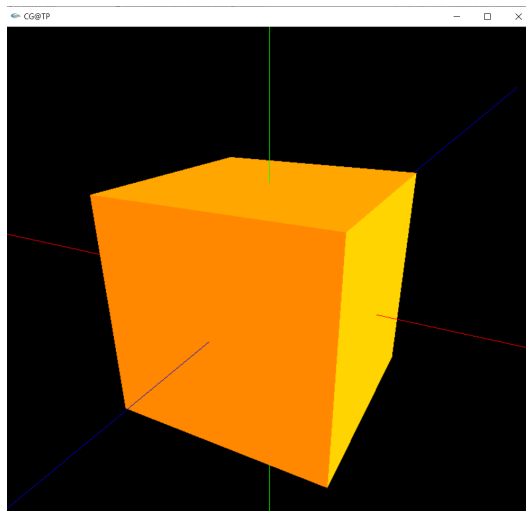


Figura 2.2: Teste 4\_2



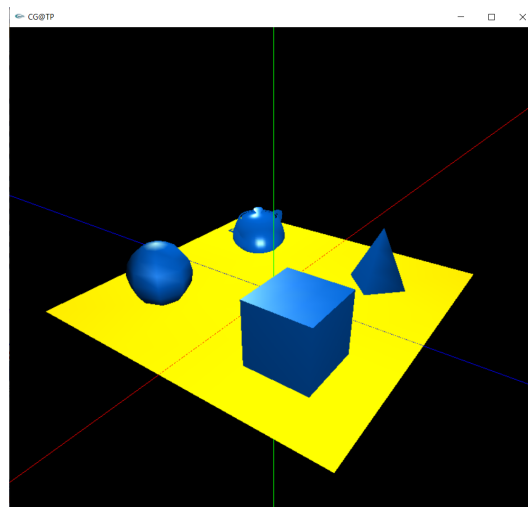


Figura 2.3: Teste 4\_3

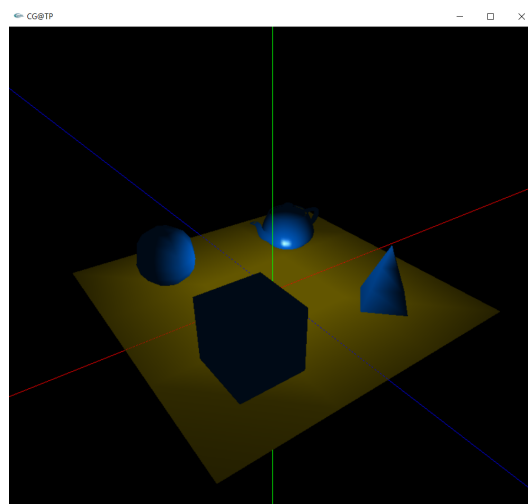


Figura 2.4: Teste 4\_4

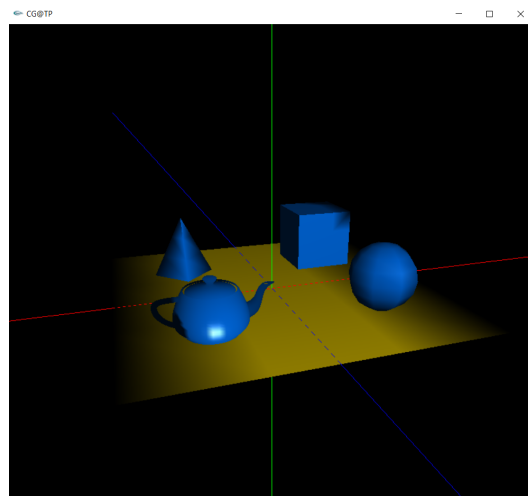


Figura 2.5: Teste 4\_5

## 2.4 Sistema Solar

As alterações referentes ao sistema solar consistem na adição de um ponto de luz no ponto de coordenadas  $(0,0,0)$ , que emite luz em todas as direções. Foram também adicionadas as iluminações dos planetas. Para cada planeta tentamos aproximar o máximo à sua cor característica daí que:

- Sol - Laranja tendo em conta a representação comum do sol visto do espaço.
- Mercúrio - Cor vermelha escura para representar a sua superfície rochosa.
- Vénus - Amarelo claro para refletir a atmosfera densa e o brilho do planeta.
- Terra - Azul, frequentemente usado para representar o nosso planeta visto do espaço.
- Marte - Vermelho de modo a representar a coloração característica deste planeta.
- Júpiter - Tom laranja para simbolizar as manchas atmosféricas.
- Saturno - Amarelo dourado representativos da atmosfera de Saturno.
- Urano - Azul claro e brilhante dado pela atmosfera de metano deste planeta.
- Netuno - Azul escuro característico da atmosfera gelada deste planeta.

Quanto ao cometa, este apresenta uma cor roxa.

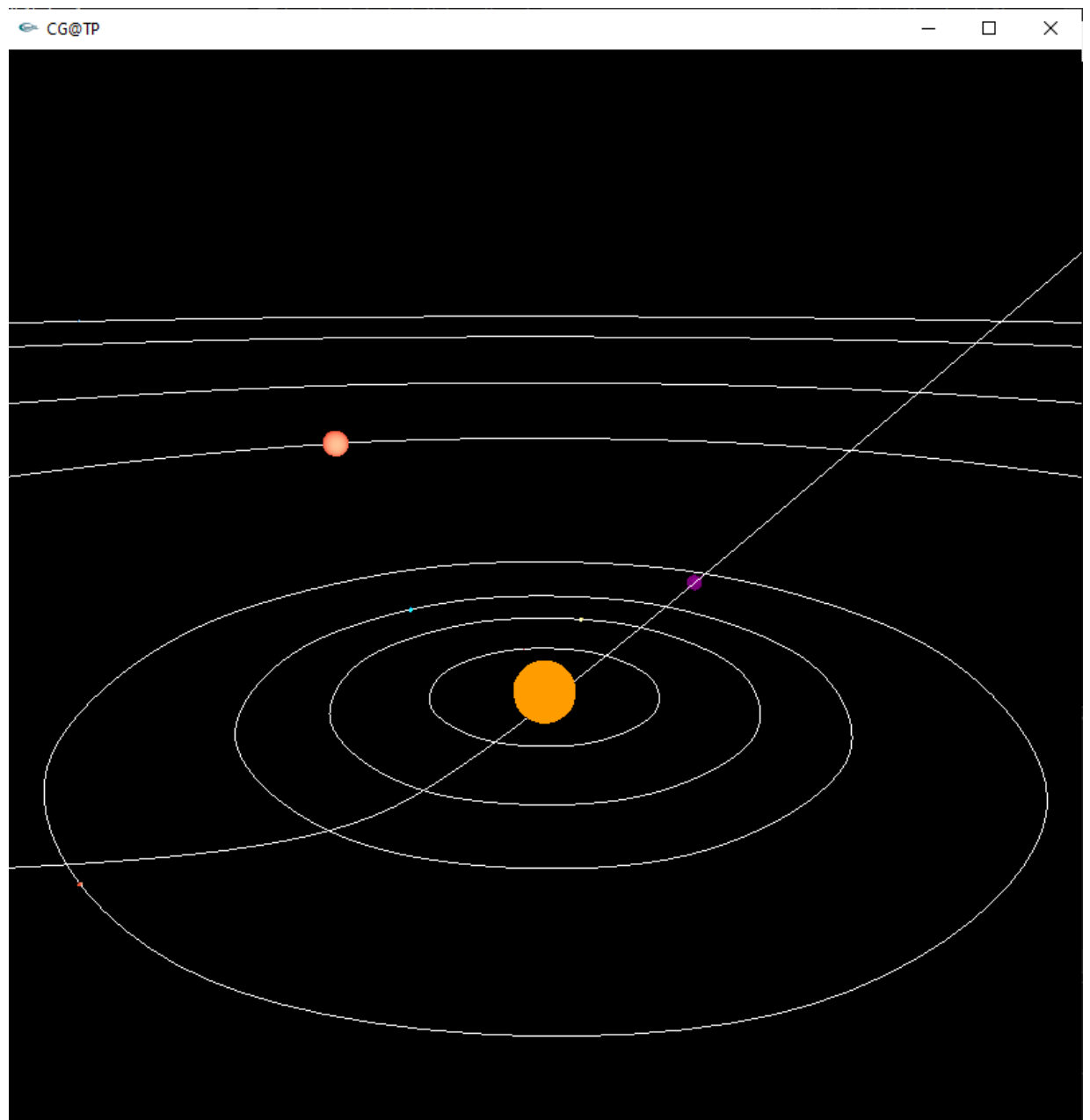


Figura 2.6: Sistema Solar

## Capítulo 3

# Conclusão

Ao longo do desenvolvimento desta fase do projeto a principal dificuldade esteve na construção das normais para cada modelo, nomeadamente, no **teapot**. O novo formato XML também nos fez criar mais duas estruturas para efeitos de iluminação o que demorou bastante tempo a ser implementado, impossibilitando-nos de trabalhar nas texturas. No entanto, dentro daquilo que foi feito, acreditámos que esta última fase do trabalho está razoável.