

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA
LICENCIATURA EM ENGENHARIA INFORMÁTICA

Computação Gráfica - Trabalho Prático - Fase 4
Ano Letivo 2021/2022
Grupo 6

Gonçalo Braz (a93178) Simão Cunha (a93262)
Tiago Silva (a93277) Gonçalo Pereira (a93168)

5 de junho de 2022

Resumo

Este relatório inicia-se com uma breve introdução do trabalho a efetuar nesta fase. Segue as alterações efetuadas às duas aplicações previamente desenvolvidas: o *generator* e o *engine*. Para o gerador são indicadas as alterações responsáveis pela geração das coordenadas para as texturas e as normais para cada vértice e para o motor gráfico são indicadas as alterações realizadas para suportar texturas e iluminação. Por fim, é apresentado duas cenas do Sistema Solar semelhantes à fase anterior mas aplicando as texturas e a iluminação, terminando este relatório com uma conclusão sobre o trabalho desenvolvido.

Conteúdo

1	Introdução	3
2	Aquitetura da solução	3
2.1	Plano	3
2.1.1	Cálculo das normais	3
2.1.2	Textura	4
2.2	Cubo	4
2.2.1	Cálculo das normais	4
2.2.2	Textura	5
2.3	Esfera	6
2.3.1	Cálculo das normais	6
2.4	Textura	8
2.5	Cone	9
2.5.1	Cálculo das normais	9
2.5.2	Textura	11
2.6	Superfície de Bézier	12
2.6.1	Cálculo das normais	12
2.6.2	Textura	14
3	Demo scenes	15
4	Conclusão	16

1 Introdução

Este relatório surge no âmbito da quarta e última fase do trabalho prático da UC de Computação Gráfica. Os objetivos desta fase foram o cálculo das coordenadas para a textura e as normais para cada vértice das diversas primitivas; implementação das funcionalidades de iluminação e de texturas nos objetos das cenas dos ficheiros XML. Além disso, foi-nos requerida que apresentássemos a cena que temos desenvolvido ao longo do trabalho com as alterações implementadas nesta fase: o sistema solar.

2 Aquitetura da solução

2.1 Plano

2.1.1 Cálculo das normais

A normal, sendo um vetor normalizado perpendicular ao plano, só podia ser $(0,1,0)$ ou $(0,-1,0)$. Como nós queremos que a parte de cima do plano seja a visível e a atingida pela luz, então a normal escolhida foi o vetor $(0,1,0)$.

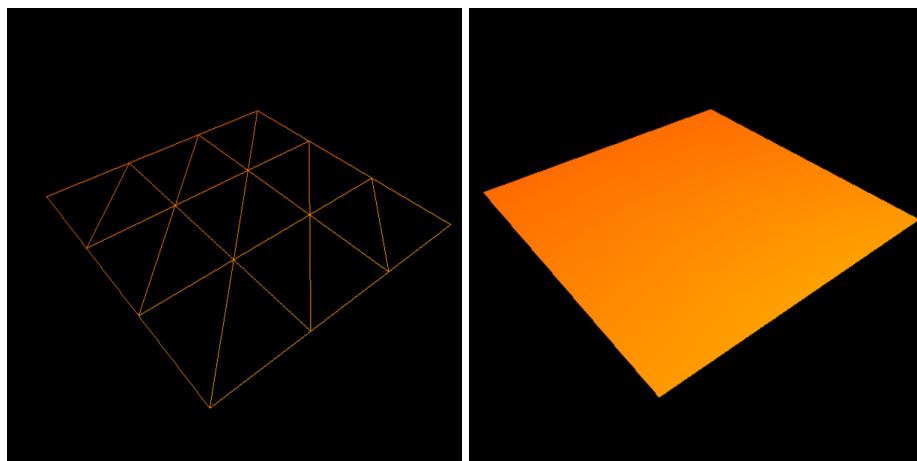


Figura 1: Exemplos das cores no plano

```
<color>
  <diffuse R="200" G="200" B="000" />
  <ambient R="50" G="50" B="00" />
  <specular R="0" G="0" B="0" />
  <emissive R="255" G="0" B="0" />
  <shininess value="128" />
</color>
```

Figura 2: Código XML com as cores utilizadas nos exemplos acima

2.1.2 Textura

Tendo em conta que o plano já é um quadrado que está centrado na origem, enquadrá-lo foi relativamente simples. Ao x e z o valor de metade do tamanho do plano e de seguida, para ter a certeza que as coordenadas ficam entre os valores 0 e 1, divide-se por esse tamanho.

$$\text{halfsize} = \frac{\text{size}}{2} \quad (1)$$

,

$$(tx, ty) = \left(\frac{x + \text{halfsize}}{\text{size}}, \frac{z + \text{halfsize}}{\text{size}} \right) \quad (2)$$

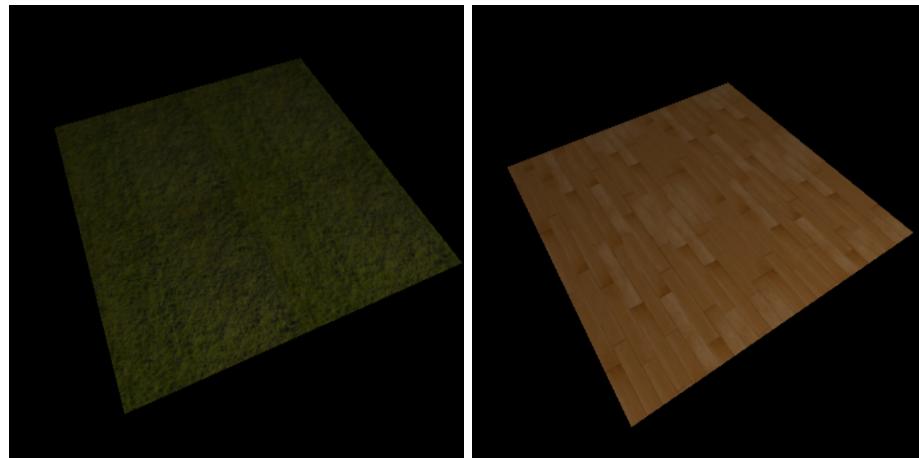


Figura 3: Exemplos de texturas no plano

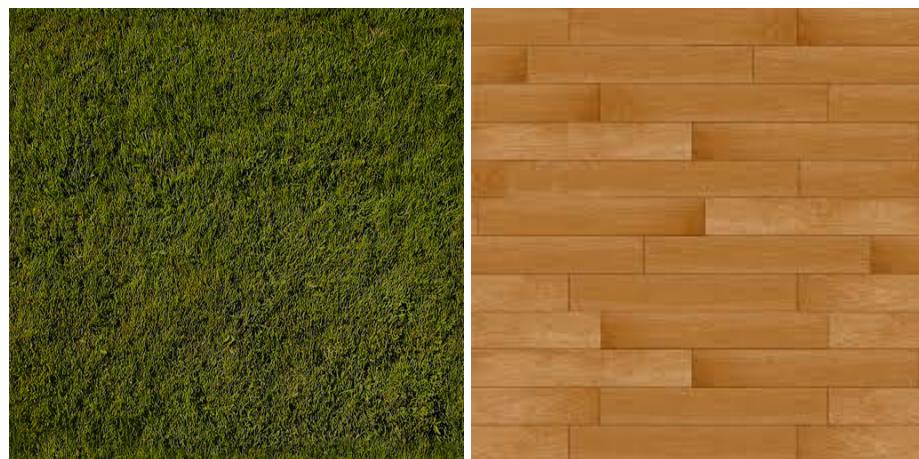


Figura 4: Texturas usadas nos exemplos acima

2.2 Cubo

2.2.1 Cálculo das normais

Para o cálculo das normais do cubo, utilizamos o mesmo raciocínio utilizado anteriormente no plano, já que o cubo é a junção de 6 planos.

Planos	Coordenadas da normal
Oxz+	(0,1,0)
Oxz-	(0,-1,0)
Oxy+	(0,0,1)
Oxy-	(0,0,-1)
Ozy+	(1,0,0)
Ozy-	(-1,0,0)

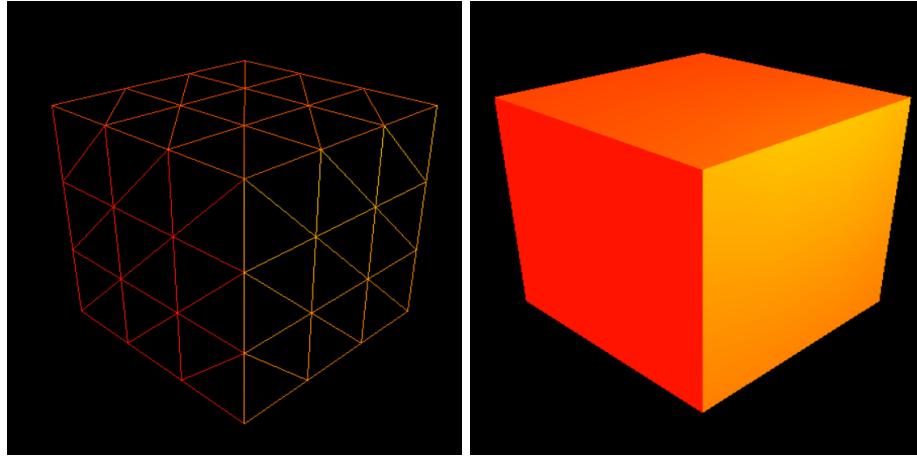


Figura 5: Exemplos das cores no cubo

Foram utilizadas as mesmas cores do código utilizado nos exemplos do plano. (Figura 2)

2.2.2 Textura

Assim como nas normais, nas texturas foi utilizado o mesmo raciocínio utilizado anteriormente no plano, pelas mesmas razões.

$$\textit{halfsize} = \frac{\textit{size}}{2} \quad (3)$$

,

Planos	Coordenadas da textura
Oxz	$\left(\frac{x+\textit{halfsize}}{\textit{size}}, \frac{z+\textit{halfsize}}{\textit{size}} \right)$
Oxy	$\left(\frac{x+\textit{halfsize}}{\textit{size}}, \frac{y+\textit{halfsize}}{\textit{size}} \right)$
Ozy	$\left(\frac{z+\textit{halfsize}}{\textit{size}}, \frac{y+\textit{halfsize}}{\textit{size}} \right)$

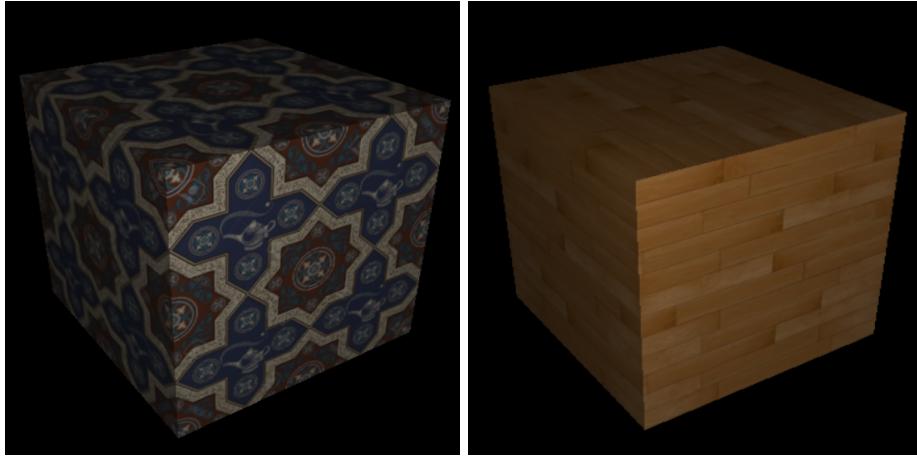


Figura 6: Exemplos de texturas no cubo



Figura 7: Texturas utilizadas

2.3 Esfera

2.3.1 Cálculo das normais

A normal (nx, ny, nz), em cada ponto da esfera, é dado pelo vetor normalizado calculado através desse mesmo ponto menos o centro da esfera.

$$(nx, ny, nz) = \text{normalize}(P(x, y, z) - C(cx, cy, cz)) \quad (4)$$

Uma vez que o nosso C será sempre (0,0,0), fica que,

$$(nx, ny, nz) = \text{normalize}(P(x, y, z)) \quad (5)$$

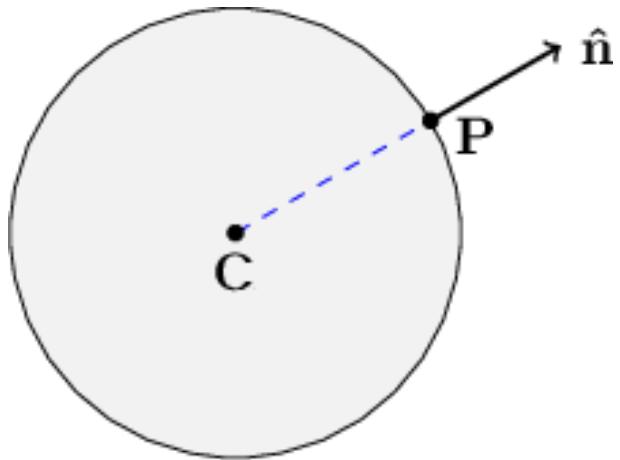


Figura 8: Vetor normal num ponto da esfera

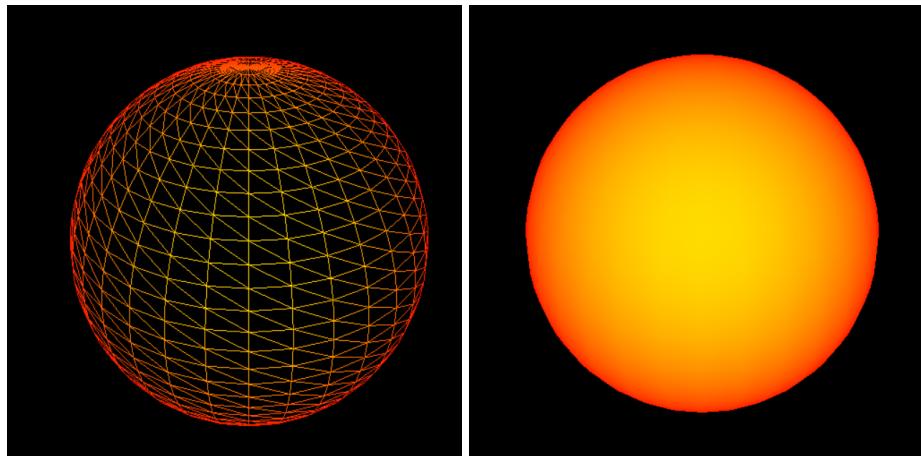


Figura 9: Exemplos das cores na esfera

Foram utilizadas as mesmas cores do código utilizado nos exemplos do plano. (Figura 2)

2.4 Textura

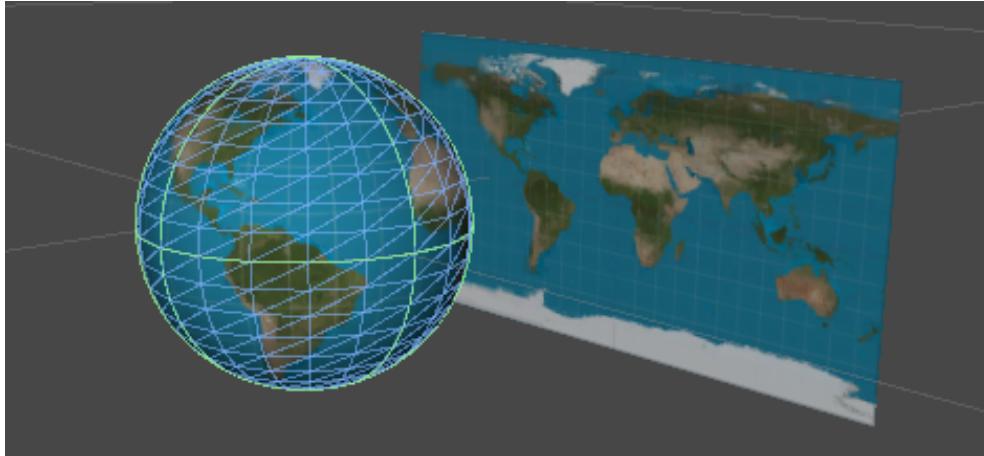


Figura 10: Exemplificação da transformação de uma textura para a superfície da esfera

Como exemplificado na imagem acima, o cálculo da textura (tx, ty) em cada ponto da esfera depende da *slice* e da *stack* em que esse ponto se encontra:

$$tx = \frac{\text{slice_atual}}{\text{slices}} \quad (6)$$

$$ty = \frac{\text{stack_atual}}{\text{stacks}} \quad (7)$$

Com isto, as texturas estavam a ficar ao contrário verticalmente, então trocamos a equação de tx , de modo a que, com a iteração das *slices*, fosse construído de forma correta.

$$tx = \frac{\text{slices} - \text{slice_atual}}{\text{slices}} \quad (8)$$



Figura 11: Exemplos de texturas na esfera



Figura 12: Texturas utilizadas

2.5 Cone

2.5.1 Cálculo das normais

Na parte de baixo do cone, a normal é paralela ao eixo dos y, sendo a direção de cima para baixo. Logo, a normal é $(0, -1, 0)$.

Em todos os outros pontos, na superfície lateral, seguimos a lógica do seguinte artigo 3.

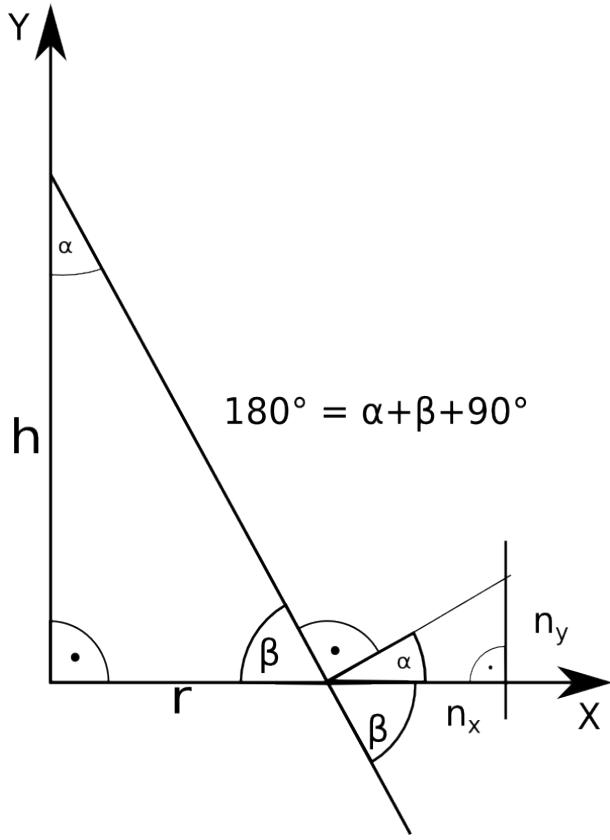


Figura 13: Cálculo da normal na superfície lateral do cone

Pela figura, podemos identificar que a normal na superfície, um vetor perpendicular à superfície, pode ser descoberto como sendo a hipotenusa de um triângulo, semelhante ao original (corte 2D do cone).

Seguindo esta figura, conseguimos identificar os passos necessários para o cálculo da componente y da normal num ponto da superfície, sendo que as componentes x e z serão iguais aos valores no ponto em questão, claro está, antes da normalização da normal.

Assim, temos que, sabendo a altura e o raio do cone, conseguimos descobrir o valor de α :

$$\alpha = \tan^{-1}\left(\frac{h}{r}\right) \quad (9)$$

Seguindo a figura, sabemos então que:

$$n_y = \sin(\alpha) \quad (10)$$

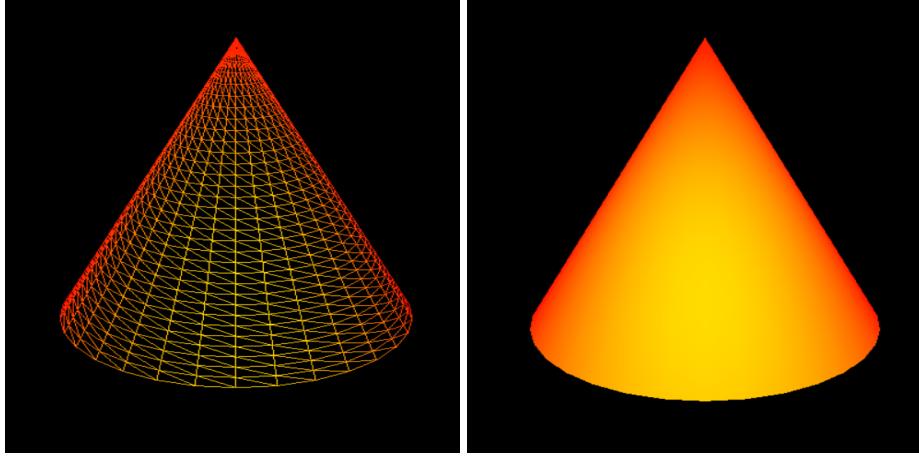


Figura 14: Exemplos das cores no cone

Foram utilizadas as mesmas cores do código utilizado nos exemplos do plano. (Figura 2)

2.5.2 Textura

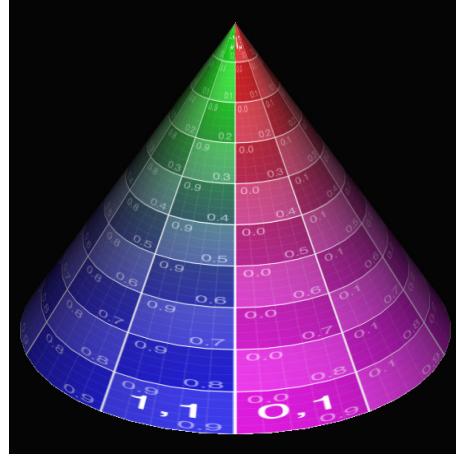


Figura 15: Exemplificação da transformação de uma textura para a superfície lateral do cone 2

Como visto na imagem acima, a textura (tx, ty) da superfície do cone, utiliza uma ideia semelhante utilizada na esfera:

$$tx = \frac{\text{slice_atual}}{\text{slices}} \quad (11)$$

$$ty = \frac{\text{slice_atual}}{\text{stacks}} \quad (12)$$

Na base, deixamos o ty fixo a 0 e o tx a depender da $slice$ atual como na superfície.



Figura 16: Exemplos de texturas no cubo



Figura 17: Texturas utilizadas

2.6 Superfície de Bézier

2.6.1 Cálculo das normais

De modo a ser possível ter qualquer um vetor normal em cada ponto da superfície de Bézier, é necessário ter 2 vetores tangentes à superfície perpendulares entre si, como é possível observar na figura abaixo.

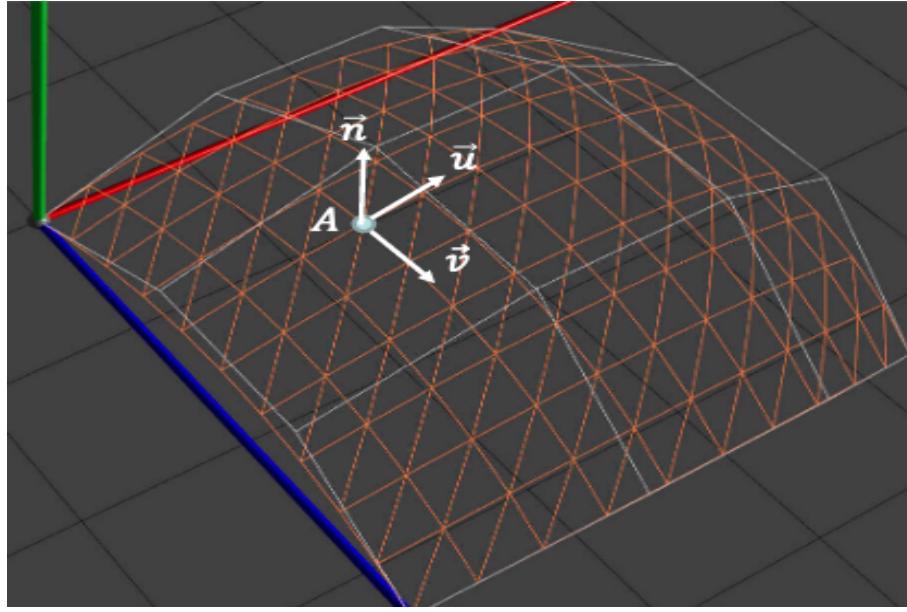


Figura 18: Cálculo da normal numa superfície de Bézier

As fórmulas que permitem obter as tangentes são:

$$\vec{u} = [3u^2 \quad 2u \quad u \quad 1] M \begin{bmatrix} P_{00} & P_{01} & P_{02} & p_{03} \\ p_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T V^T \quad (13)$$

$$\vec{v} = U M \begin{bmatrix} P_{00} & P_{01} & P_{02} & P_{03} \\ P_{10} & P_{11} & P_{12} & P_{13} \\ P_{20} & P_{21} & P_{22} & P_{23} \\ P_{30} & P_{31} & P_{32} & P_{33} \end{bmatrix} M^T V^T \begin{bmatrix} 3v^2 \\ 2v \\ v \\ 1 \end{bmatrix} \quad (14)$$

Onde,

$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}; U = [u^3 \quad u^2 \quad u \quad 1]; v = \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix} \quad (15)$$

Tendo as tangentes calculadas o vetor normal num dado ponto é definida pelo resultado normalizado do produto vetorial da **tangente2** pela **tangente1** (obedecer regra da mão direita), sendo que os vetores tangentes devem estar também normalizados.

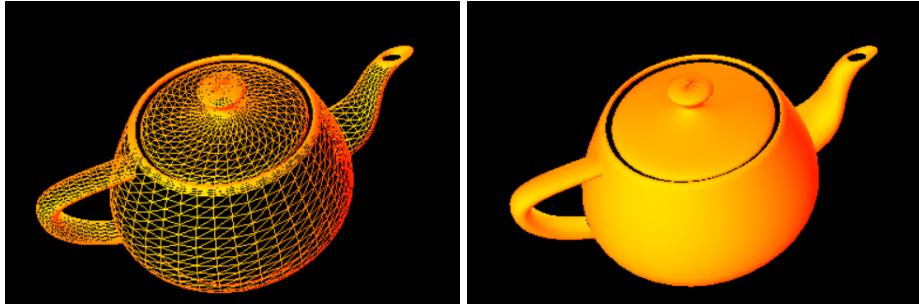


Figura 19: Exemplos das cores no *teapot* (calculado com *Bézier*)

Foram utilizadas as mesmas cores do código utilizado nos exemplos do plano. (Figura 2)

2.6.2 Textura

Para a textura utilizamos:

$$(tx, ty) = \left(\frac{u}{tesselation}, \frac{v}{tesselation} \right) \quad (16)$$

sendo u e v valores que variam entre 0 e $tesselation$. Os valores para a textura variam entre entre 0 e 1 tal como no espaço de textura.



Figura 20: Exemplos de texturas no *teapot*

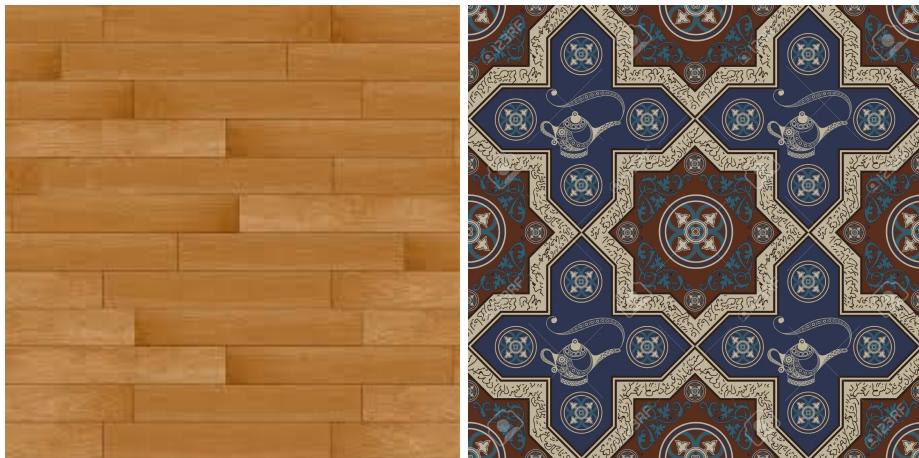


Figura 21: Exemplos de texturas no *teapot*

3 Demo scenes

Nesta fase do trabalho prático, foi-nos requerida a finalização da cena que temos desenvolvido ao longo das diferentes fases do trabalho prático: o sistema solar. Uma vez que na fase anterior apresentamos duas cenas semelhantes (uma com apenas o desenho da órbita do cometa *teapot* e outra com o desenho de todas as órbitas de todos os astros do Sistema Solar - excluindo o Sol), decidimos manter a mesma ideia e apresentamos duas cenas semelhantes.

Aqui, foi colocada todas as devidas texturas nos planetas e respetivas luas e tratamos da iluminação de forma a tornar a cena minimamente realista.

O primeiro passo a tomar foi aplicar todas as texturas dos astros (retiramos deste website - 1). Uma vez que vamos aplicar texturas, sabemos que não vai ser necessário colorir os objetos da cena.

De seguida, partimos para o campo da iluminação. Uma vez que os planetas têm de estar iluminados apenas na face voltada para o sol, colocamos uma luz no centro do mesmo do tipo *point*. Além disso, este astro tem de ser luminoso, pelo que aumentamos os campos *RGB* da luz emissiva.

Com isto, formamos as seguintes cenas:

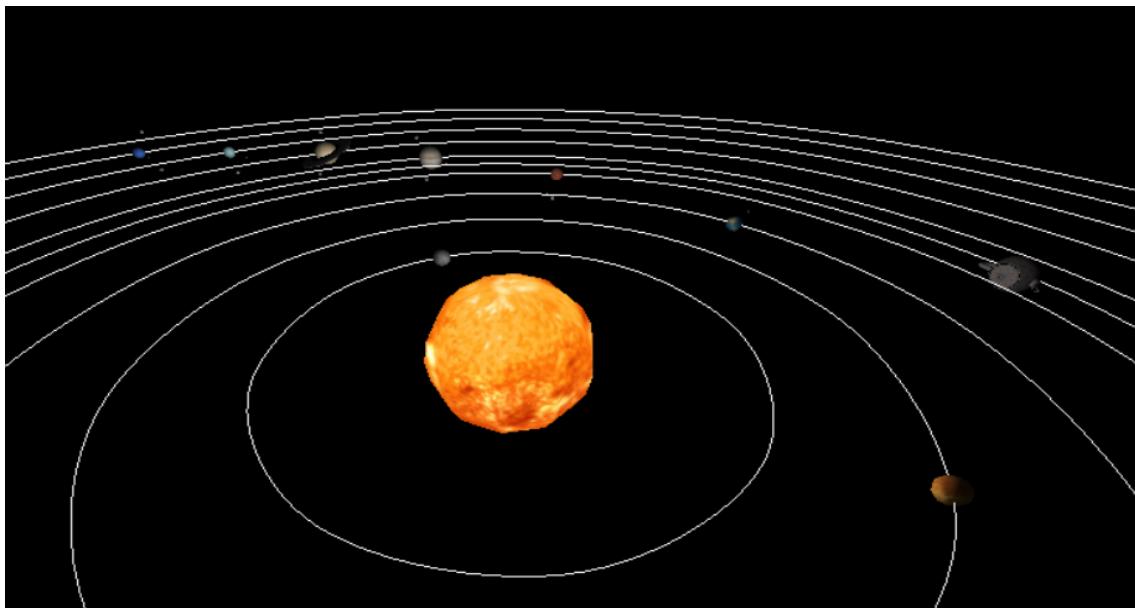


Figura 22: Sistema solar - apenas com órbita do cometa

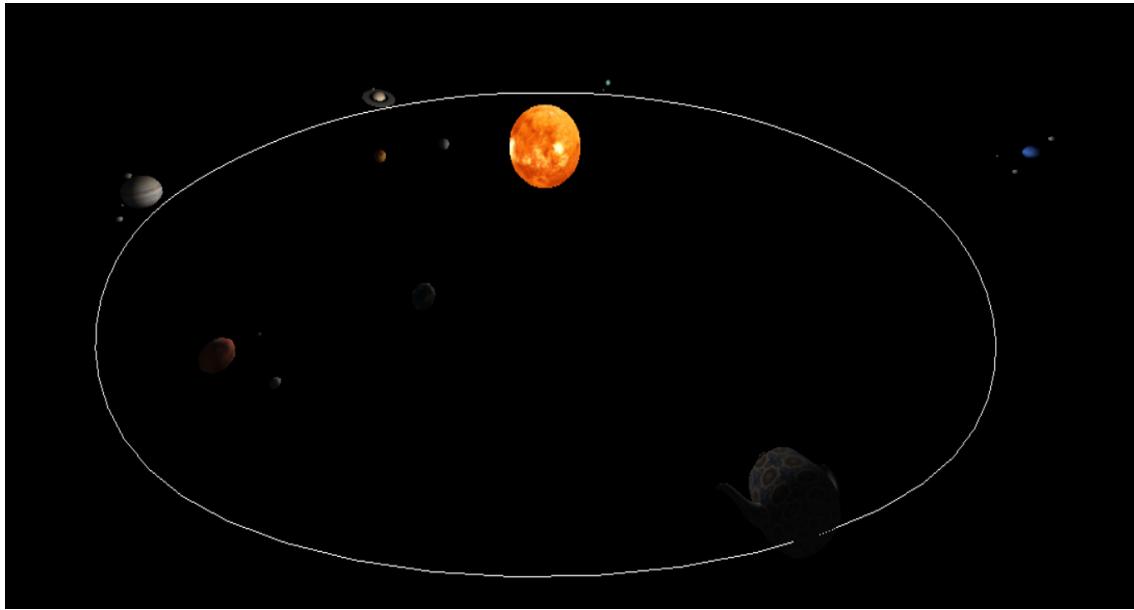


Figura 23: Sistema solar - órbita de todos os planetas

4 Conclusão

Nesta fase do trabalho prático, dedicamo-nos à implementação de texturas e iluminação. Isto implicou alterações do *generator* para cada primitiva de modo a que fossem geradas as coordenadas de textura e as normais para a iluminação. Além disso, efetuámos mudanças no *engine* para suportar os ficheiros XML que incluem as texturas e a iluminação. Tivemos alguns problemas em relação às cores em objetos que sofriam alterações de scaling, mas foi facilmente resolvido com uma pequena pesquisa (4).

Em suma, o presente relatório traduz-se na última fase do projeto que tem sido desenvolvido ao longo do semestre, que resulta na aplicação de todos os conhecimentos de Computação Gráfica lecionados na unidade curricular.

Referências

- [1] Repositório de texturas
- [2] Fórum com explicação de renderização das texturas de um cone
- [3] Artigo no Stack Overflow sobre o cálculo das normais do cone
- [4] Artigo no Stack Overflow sobre a mudança de cor aquando *scaling*