



UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Computação Gráfica

Fase IV – Normais e Coordenadas de Textura

João Neves (a81366) Luís Manuel Pereira (a77667)
Rui Fernandes (a89138) Tiago Ribeiro (a76420)

Maio 2021

Resumo

O presente relatório descreve o trabalho prático realizado no âmbito da disciplina de *Computação Gráfica*, ao longo do segundo semestre do terceiro ano do Mestrado Integrado em Engenharia Informática da Universidade do Minho.

Esta fase consiste na adição de novas funcionalidades e modificações ao trabalho já realizado, tais como o uso de texturas e a aplicação de iluminação, por forma a obter representações mais realistas dos sistemas a representar.

Neste documento descrevemos sucintamente a aplicação desenvolvida discutimos as decisões tomadas durante a realização do trabalho prático.

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 2 | <i>Generator</i> | 2 |
| 2.1 | Ficheiros XML | 2 |
| 2.2 | Normais & Coordenadas de Textura | 2 |
| 2.2.1 | Plano | 3 |
| 2.2.2 | Paralelepípedo | 4 |
| 2.2.3 | Cone | 5 |
| 2.2.4 | Esfera | 7 |
| 2.2.5 | <i>Torus</i> | 7 |
| 2.2.6 | <i>Patches</i> de Bézier | 8 |
| 3 | <i>Engine</i> | 9 |
| 3.1 | Iluminação | 9 |
| 3.2 | <i>VBOs</i> | 10 |
| 4 | Análise de Resultados | 11 |
| 5 | Conclusão | 13 |

Lista de Figuras

| | | |
|----|---|----|
| 1 | Mapeamento de texturas | 3 |
| 2 | Vetores normais ao plano xOz | 3 |
| 3 | Coordenadas das texturas (à esquerda) e coordenadas dos vértices do plano (à direita) | 4 |
| 4 | Vetores normais ao paralelepípedo | 4 |
| 5 | Formato da textura de um paralelepípedo | 5 |
| 6 | Circunferência paralela à base de um cone | 5 |
| 7 | Vetores normais ao cone | 6 |
| 8 | Formato da textura de um cone | 6 |
| 9 | Vetores normais à esfera | 7 |
| 10 | Vetores normais ao <i>torus</i> | 7 |
| 11 | Modelo de Iluminação de Phong | 10 |
| 12 | Aplicação de Texturas e Iluminação | 10 |
| 13 | Renderização do modelo dinâmico do Sistema Solar | 11 |
| 14 | Renderização do modelo dinâmico do Sistema Solar | 12 |
| 15 | Renderização do modelo dinâmico do Sistema Solar | 12 |

1 Introdução

Nesta quarta e última fase do projeto proposto no âmbito da unidade curricular de Computação Gráfica, damos seguimento ao trabalho desenvolvido nas fases anteriores. Nesta fase pretende-se estender a aplicação de modo a que o gerador de primitivas gere, para além das coordenadas de cada vértice, as normais e coordenadas de textura.

Relativamente ao motor gráfico, este deve ser suportar as funcionalidades de iluminação e texturização, assim como possibilitar a pormenorização de cores e texturas no ficheiro XML.

Dito isto, nos capítulos que se seguem será explicado, detalhadamente, todo o processo para obter os resultados pretendidos.

2 Generator

Até à fase anterior, o *generator* gerava as coordenadas dos pontos constituintes de cada um dos modelos. No entanto, de forma a poder prosseguir à aplicação de texturas, este deve ser alterado de modo a calcular também as coordenadas normais e de textura de cada um dos ponto.

2.1 Ficheiros XML

De maneira a ser possível implementar as novas funcionalidades, foi necessária a adição de novas *tags* e atributos ao ficheiro XML da fase anterior.

Uma vez que todos os elementos da cena passam a possuir textura, foi adicionado dentro de cada modelo, junto ao atributo *file* que indica o ficheiro .3d utilizado, o atributo *texture* que referencia imagem representativa da textura.

```
<models>
    <model file="sphere.3d" texture="../textures/2k_jupiter.jpg" />
</models>
```

Para além disso, foi também adicionada a indicação de todas as luzes a ser consideradas:

```
<lights>
    <light type="POINT" posX="0" posY="0" posZ="0" diffR="1" diffG="1"
          diffB="1" />
</lights>
```

Por fim, adicionou-se, também, as características dos materiais associados a cada uma das primitivas:

```
<models>
    <model file="sphere.3d" diffR="0.1" diffG="0.2" diffB="0.3" />
</models>
```

Tendo estes aspectos em consideração, é possível notar que o ficheiro XML evoluiu bastante ao longo das fases, passando de um simples ficheiro com primitivas geométricas, a um ficheiro complexo onde existem transformações geométricas, curvas, texturas e luzes.

2.2 Normais & Coordenadas de Textura

As normais de cada um dos vértices são gerados para que, a partir de uma fonte de luz, seja possível gerar uma sombra.

De modo a obter as normais para os diferentes vértices de cada figura, foi necessário efetuar um estudo de cada triângulo que constitui a mesma. É, assim, necessário obter um vetor normal para cada vértice, *i.e.* um vetor perpendicular à superfície nesse ponto.

Por outro lado, para conseguir inserir texturas nos objetos, surge a necessidade de definir os pontos textura para cada uma das figuras.

Desta forma, é necessário o estudo de um plano 2D para a figuras geométricas tridimensionais, usando como revestimento das mesmas. O mapeamento é feito da seguinte forma:

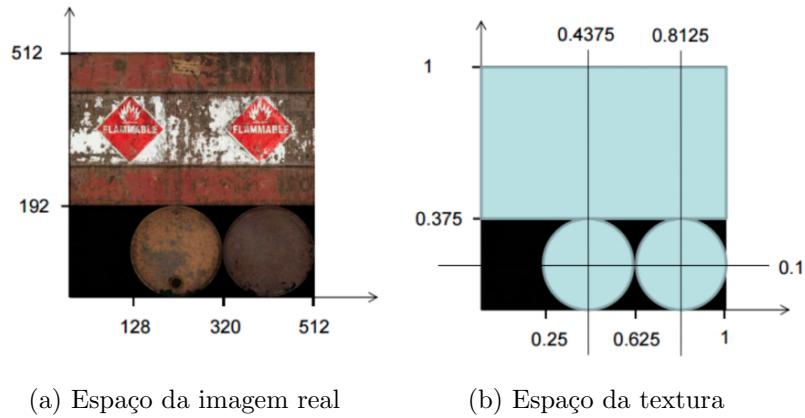


Figura 1: Mapeamento de texturas

Tal como se pode verificar na figura anterior, estabelece-se que o eixo cartesiano da textura tem como limite máximo 1 e um limite mínimo de zero. Importa notar que estes limites correspondem aos limites da imagem real, no entanto, mapeados para valores no intervalo $[0, 1]$, usando o canto inferior esquerdo de todas as imagens como origem.

2.2.1 Plano

Tendo em conta a orientação do plano, isto é, se este se encontra de frente para o ecrã ou virado na mesma direção (ordem dos pontos), as normais serão um vetor constituído pela orientação e valores nulos nos restantes eixos.

Por exemplo, para um plano no eixo xOz , com $y = 0$ as normais dos pontos serão $(0, 1, 0)$, tal como representado na Figura 2.

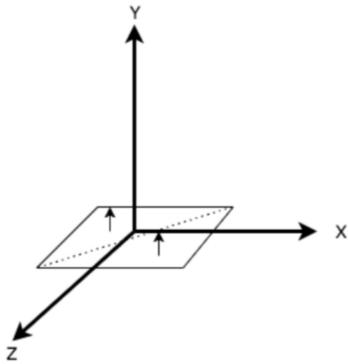


Figura 2: Vetores normais ao plano xOz

O processo de obtenção das coordenadas de textura é simples no caso desta figura geométrica. O formato é o mesmo que o da imagem 2D, sendo apenas necessário fazer a correspondência direta de cada vértice do plano com os vértices da imagem 2D, tal como ilustrado de seguida.

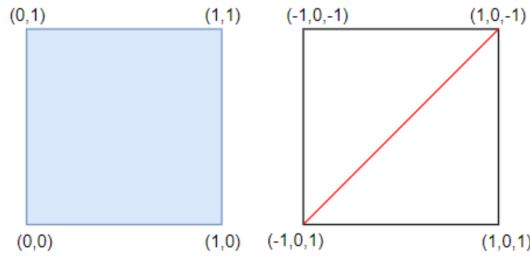


Figura 3: Coordenadas das texturas (à esquerda) e coordenadas dos vértices do plano (à direita)

2.2.2 Paralelepípedo

Seguindo um raciocínio análogo, o cálculo dos vetores normais associados a um paralelepípedo torna-se trivial. Desta forma, tal como ilustrado na Figura 4, dependendo da face em questão, os vetores normais serão os seguintes:

- Face Frontal: $(0, 0, 1)$
- Face Esquerda: $(-1, 0, 0)$
- Face Traseira: $(0, 0, -1)$
- Topo: $(0, 1, 0)$
- Base: $(0, -1, 0)$
- Face Direita: $(1, 0, 0)$

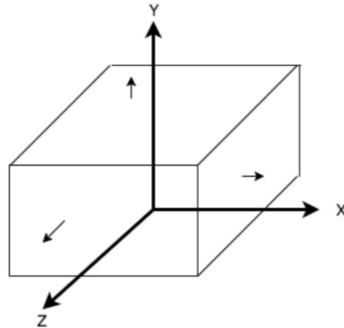


Figura 4: Vetores normais ao paralelepípedo

Para a obtenção das coordenadas de textura do paralelepípedo, estabeleceu-se uma imagem 2D como regra possuindo sempre o seguinte formato:

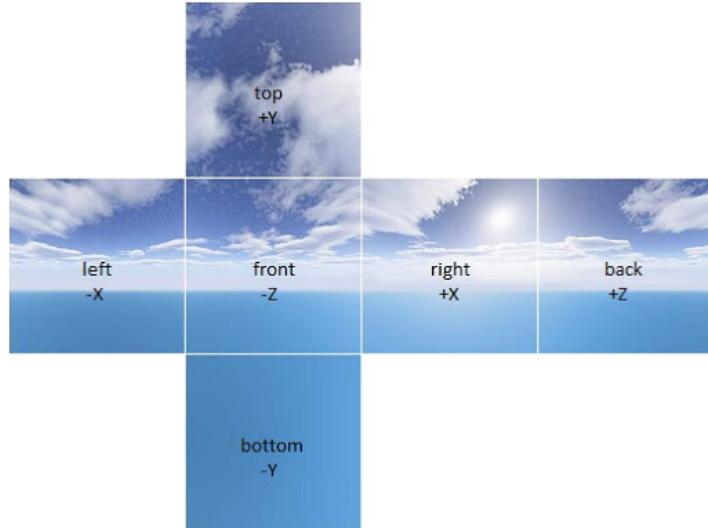


Figura 5: Formato da textura de um paralelepípedo

Desta forma, para calcular as coordenadas de textura, obtém-se a posição da imagem a que corresponde a face em questão e itera-se no mesmo sentido e, que o paralelepípedo é desenhado, atribuindo textura a cada vértice o ponto de textura correspondente.

2.2.3 Cone

O cone possui dois tipos de normais, as da sua base e as da sua superfície lateral. Sendo assim, as normais referentes à base do cone são dadas pelo vetor $(0, -1, 0)$.

No que diz respeito à superfície do cone, a determinação dos vetores normais é realizada em dois passos. Primeiro, determinam-se os vetores normais para uma circunferência paralela à base, como a representada em seguida:

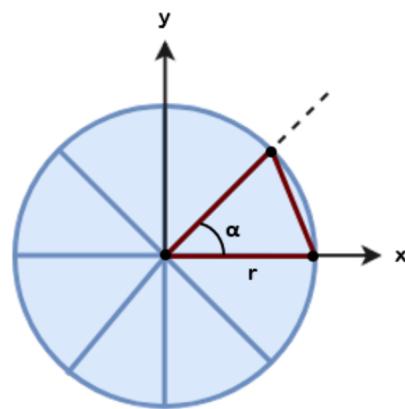


Figura 6: Circunferência paralela à base de um cone

De seguida, representando por h a altura do cone e por r o raio da base, determina-se a inclinação do clone, θ , da seguinte forma:

$$\tan(\theta) = \frac{h}{r} \Leftrightarrow \theta = \arctan\left(\frac{r}{h}\right)$$

Por fim, tal como ilustrado na Figura 7, um vértice na superfície lateral do cone terá como vetor normal o seguinte vetor:

$$\vec{N} = (\cos(\alpha) \times \cos(\theta), \sin(\theta), \sin(\alpha) \times \cos(\theta))$$

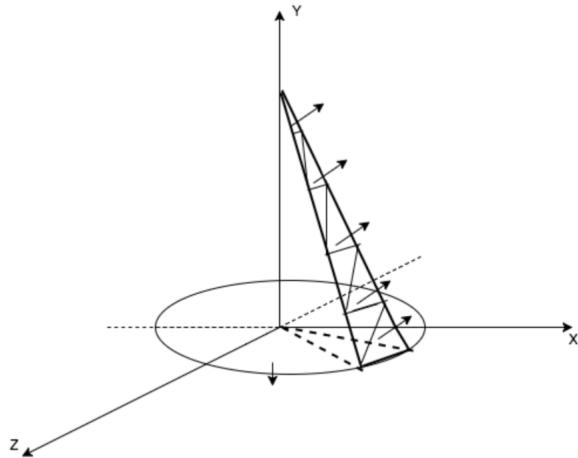


Figura 7: Vetores normais ao cone

De forma a aplicar texturas, começou-se por definir o formato que estas terão, representado em seguida:

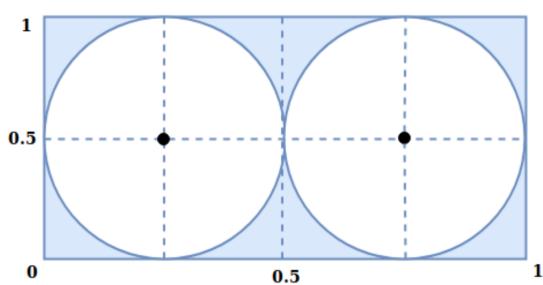


Figura 8: Formato da textura de um cone

Deste modo, definiu-se a circunferência da esquerda como sendo a representação da base do cone, enquanto a da direita representa a sua superfície lateral.

2.2.4 Esfera

O cálculo dos vetores normais à esfera é realizado tendo em conta a origem da mesma e um ponto arbitrário que esteja presente nela. Uma vez que a própria representação da esfera já se baseia neste pressuposto, conclui-se que o valor das normais da figura poderá corresponder às coordenadas destes pontos. Ou seja, o vetor das normais poderá ser representado como $\vec{N} = \vec{OP} = P - O = P$.

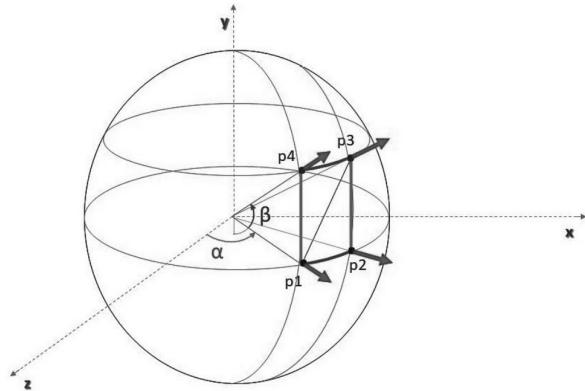


Figura 9: Vetores normais à esfera

Para associar um vértice a um ponto de textura, apenas é necessário conhecer o número de *stacks* e *slices*. Assim, conhecendo o valor da variação associada a cada um destes parâmetros e a ordem dos pontos na criação da esfera, é possível mapear facilmente a textura.

Assim, para cada vértice (i, j) , a sua textura será:

$$\left(\frac{j}{slices}, \frac{i}{stacks} \right)$$

2.2.5 Torus

Por forma a obter os vetores normais ao *torus*, aplicou-se o mesmo raciocínio que utilizado na esfera, em que o próprio ponto corresponde à sua normalização. Desta forma, as expressões ficam iguais às da esfera.

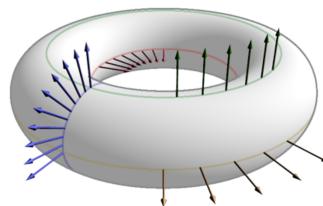


Figura 10: Vetores normais ao *torus*

Quanto às coordenadas das texturas, cada anel do *torus* corresponde a uma textura, fazendo com que chegando ao fim de todas as iterações tenha sido preenchido cada um destes anéis, estando, portanto, o *torus* preenchido na totalidade.

2.2.6 Patches de Bézier

Para calcular as normais de um dado ponto (u, v) de uma superfície de Bezier, recorre-se às seguintes equações:

$$\frac{\partial B(u, v)}{\partial u} = [3u^2 \quad 2u \quad 1 \quad 0] MPM^T V^T$$

$$\frac{\partial B(u, v)}{\partial u} = UMPM^T \begin{bmatrix} 3v^2 \\ 2v \\ 1 \\ 0 \end{bmatrix}$$

Tendo as derivadas parciais, o vetor normal a qualquer ponto na superfície é o resultado normalizado do produto externo dos vetores tangentes ao ponto (sendo eles o resultado das duas equações acima). Relativamente às texturas, a um ponto (u, v) corresponde a textura (u, v) , uma vez que u e v variam no intervalo $[0, 1]$, tal como a textura.

3 *Engine*

O *engine* é responsável pelo armazenamento da informação dos modelos a representar, assim como as respetivas transformações geométricas de cada um desses modelos, que no seu conjunto, irão formar o modelo do Sistema Solar que se pretende representar.

Esta aplicação também sofreu alterações na sua implementação, resultantes da necessidade de incorporar novas características dos modelos – *iluminação*. Findando o processo de leitura do ficheiro, passamos para a fase seguinte, a qual é responsável pela renderização da informação previamente obtida.

3.1 Iluminação

A iluminação dos modelos gerados no projeto é corretamente obtida através do cálculo das várias normais, sendo que é através destas que se determina a intensidade da luz.

Posição

Para a criação de diferentes cenários e de modo a permitir uma simulação mais realista foram usados tipos diferentes de luz:

- **POINT** – Trata-se de uma luz que é posicionada num ponto e emite feixes de luz em todas as direções à sua volta;
- **DIRECTIONAL** – Trata-se de uma luz que vem de um ponto no infinito e tem uma direção apenas de maneira a atingir o cenário de uma só direção.

Cor

De forma a ser possível ter uma percepção complexa e real da luz e da sua reflexão por parte dos objetos, a cor deve possuir um total de 4 parâmetros:

- **GL_AMBIENT** – Corresponde à cor de um objeto quando este se encontra à sombra, ou seja, não existe nenhum feixe de luz que o atinja;
- **GL_DIFFUSE** – Este componente representa a cor do objeto, *i.e.*, a cor que possui quando atingido com uma luz branca;
- **GL_EMISSION** – Cor da luz emitida por um objeto;
- **GL_SPECULAR** – Cor da reflexão especular característica a uma superfície brilhante.

Na Figura 11, encontra-se um representado um exemplo de como alguns dos diferentes parâmetros mencionados afetam a visualização do objeto.

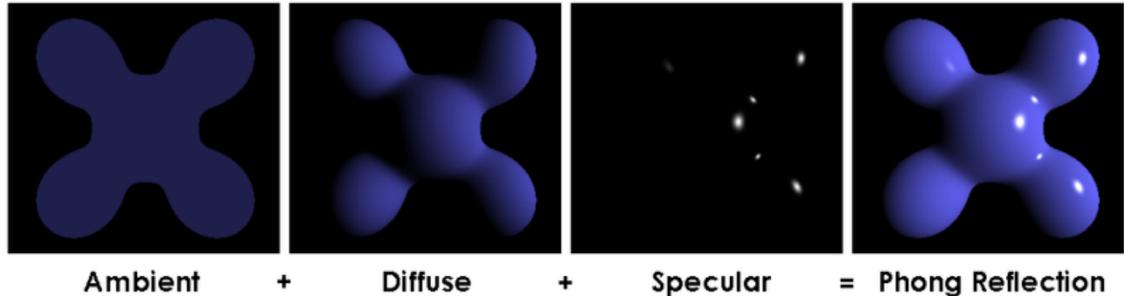
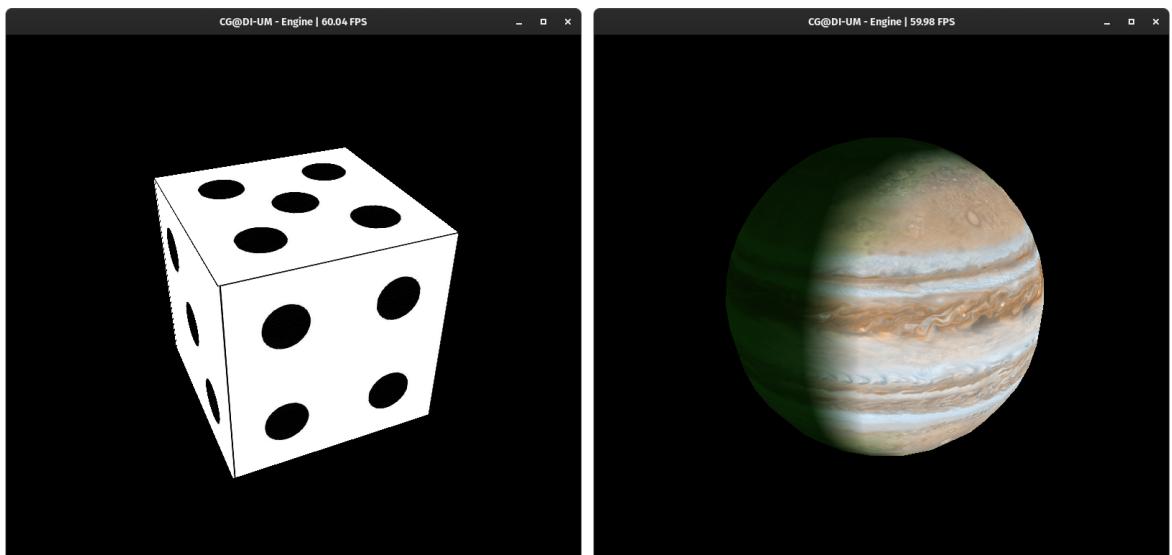


Figura 11: Modelo de Iluminação de Phong

Tendo em conta os aspectos supramencionados, apresentam-se, de seguida, alguns exemplos da aplicação de texturas e iluminação:



(a) Textura aplicada a um paralelepípedo (b) Textura e iluminação aplicada a uma esfera

Figura 12: Aplicação de Texturas e Iluminação

3.2 VBOs

Tal como na fase anterior continuamos a fazer uso de VBOs (*Vertex Buffer Object*) para passar diretamente para a placa gráfica um *array* com as coordenadas dos vértices dos objetos a serem gerados. Com a implementação de texturas e iluminação, tornou-se, então, necessário criar mais 2 *VBOs*, um com as normais e outra com a textura de cada um dos pontos.

4 Análise de Resultados

O resultado final do modelo Sistema Solar desenvolvido ao longo do projeto correspondeu às expectativas do grupo. Todos os planetas foram representados a uma escala próxima da real, sendo sujeitas a ligeiros ajustes para permitir uma maior clareza, os seus tempos de rotação e translação têm em consideração as diferenças reais entre corpos celestes, e alguns satélites naturais foram representados, de modo a tornar o cenário o mais agradável possível.

Por fim, importa salientar que, para além do apresentado nestas imagens, o *engine* é capaz de processar qualquer uma das primitivas geométricas definidas nas fases anteriores, aplicando-lhes texturas e iluminação.

Apresentam-se, então, alguns dos resultados obtidos:



Figura 13: Renderização do modelo dinâmico do Sistema Solar

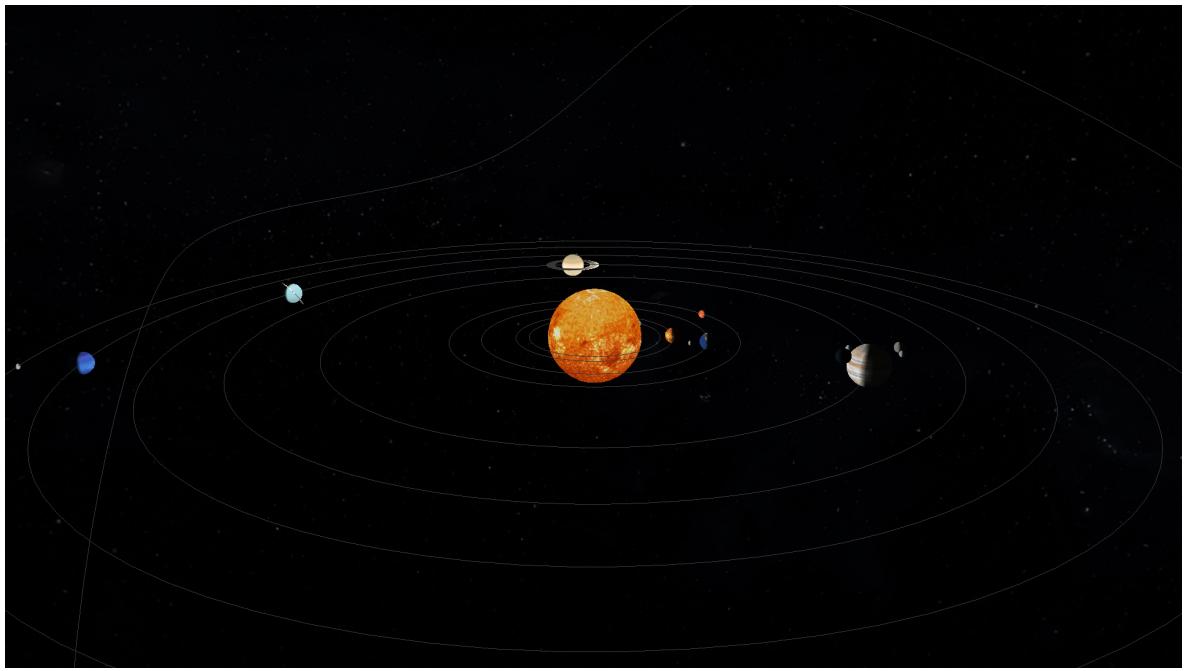


Figura 14: Renderização do modelo dinâmico do Sistema Solar

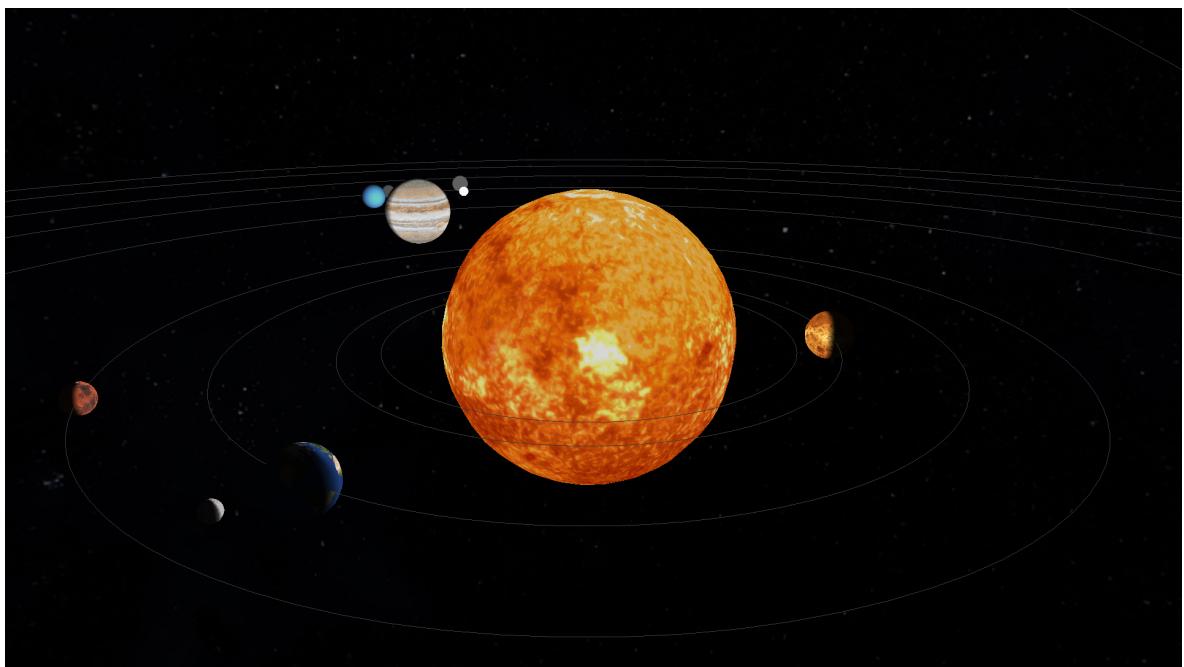


Figura 15: Renderização do modelo dinâmico do Sistema Solar

5 Conclusão

A realização desta quarta e última fase do trabalho prático permitiu assimilar conhecimento relativo aos processos de iluminação e aplicação de texturas em Computação Gráfica.

No entanto, o grupo deparou-se com algumas dificuldades, especialmente no que diz respeito ao mapeamento de uma textura a uma determinada figura. Tal deveu-se sobretudo ao facto de este mapeamento poder ser feito de várias formas. Apesar disto, após algumas tentativas e erros o grupo conseguiu superar o obstáculo com sucesso.

Em termos gerais, considera-se que o trabalho prático foi concluído com sucesso, permitindo desenvolver conhecimentos que relacionam a geometria à programação, assim como retratar cenários animados e com iluminação. Além disso, importa salientar que o trabalho desenvolvido permite gerar qualquer tipo de cenário através de um ficheiro XML, desde que este siga uma certa gramática.