

Computação Gráfica

4ª Fase – Normais e Coordenadas de Textura

André Geraldes 67673

Patrícia Barros 67665

Sandra Ferreira 67709



Conteúdo

Conteúdo	2
Índice de Figuras	3
Introdução	4
Desenvolvimento.....	5
Iluminação e Normais.....	5
Texturas.....	7
<i>Motor 3D</i>	8
Análise de Resultados.....	9
Conclusão	12



Índice de Figuras

Figura 1 Exemplo inicio ficheiro esfera.3d	5
Figura 2 Normais	5
Figura 3 Mapeamento de imagens.....	7
Figura 4 Esquema de textura para esfera	7
Figura 5 Alterações classe Primitiva	8
Figura 6 Visualização do Sistema Solar sem órbitas.....	9
Figura 7 Visualização lateral de um excerto do Sistema Solar	9
Figura 8 Sistema Solar visto de cima	10
Figura 9 Visualização lateral do Sistema Solar	11



Introdução

Este trabalho foi desenvolvido no âmbito da Unidade Curricular de Computação Gráfica, pertencente ao plano de estudos do 3º ano da licenciatura em Engenharia Informática.

Este projeto será constituído por 4 fases distintas com o objetivo final de criar um *motor 3D*. Nesta quarta fase foi-nos proposto fazer algumas alterações no trabalho desenvolvido anteriormente, nomeadamente a implementação de iluminação e texturização no motor 3D e também alterações no Gerador de forma a gerar as normais e coordenadas de textura para cada vértice, para além das coordenadas dos pontos.

Desenvolvimento

Iluminação e Normais

Para gerar a iluminação no sistema solar é necessário definir os pontos das normais, só desta forma podemos obter a iluminação que está dependente do ângulo resultante entre a normal em cada ponto e a direção da luz.

Para a implementação desta funcionalidade foi necessário alterar o nosso Gerador para que este seja capaz de calcular os valores das normais e armazená-las no ficheiro .3d. Começamos por alterar a maneira como o ficheiro .3d está organizado, nesta fase as primeiras 3 linhas do ficheiro contém informações sobre a primitiva:

1	2400
2	20
3	20

Figura 1 Exemplo início ficheiro esfera.3d

A primeira linha indica o número de pontos da primitiva, que é também o número de pontos de normais e de textura. As linhas seguintes dizem respeito ao número de camadas horizontais e verticais respectivamente.

Optamos por utilizar normais em cada ponto em vez de normais por plano (triângulo), como indica a figura seguinte, o que permitiu uma iluminação mais realista e suave.

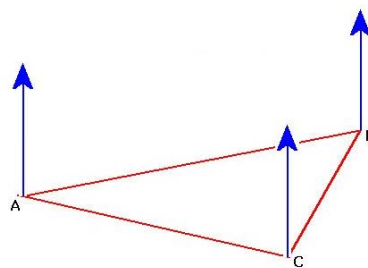


Figura 2 Normais

Os pontos das normais são calculados dividindo o valor de x, y, z em cada ponto pelo valor do raio como mostra o seguinte exemplo:

```
p = Ponto::Ponto(x1/raio,y1/raio,z1/raio);  
normais.push_back(p);
```



No nosso sistema solar optamos por colocar apenas um ponto de luz que se encontra dentro do sol, ou seja, centrado na origem.

```
<luzes>  
  <luz tipo="POINT" posX="0.0" posY="0.0" posZ="0.0" />  
</luzes>
```

Permitido assim iluminar todo o sistema solar e ficar o mais realista possível em termos de iluminação e sombras dos planetas.

Texturas

Para a implementação das texturas é necessário o mapeamento das texturas de forma a poder “cobrir” os objetos 3D com a imagem da textura, simulando assim o aspecto real do sol e planetas. As texturas são também coordenadas que são calculadas pelo Gerador, e que concluem o nosso ficheiro .3d, o calculo destes pontos é feito de maneira a mapear cada ponto da figura a um ponto da imagem que contém a textura da seguinte forma:

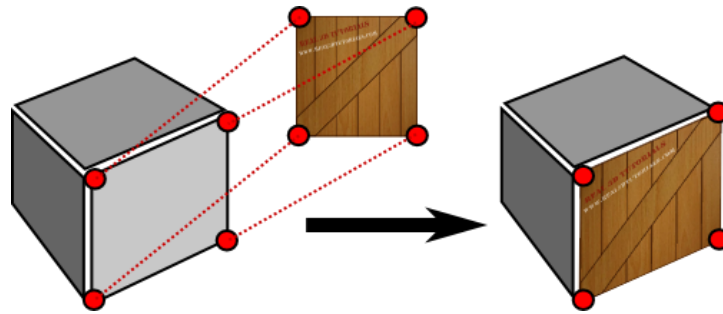


Figura 3 Mapeamento de imagens

O mapeamento da esfera é idêntico, visto que esta é desenhada por camadas. O esquema utilizado por nós foi idêntico ao da figura seguinte:

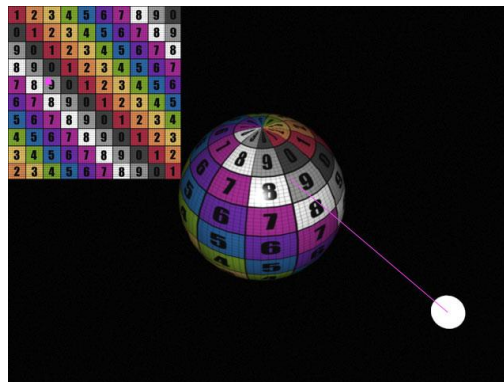


Figura 4 Esquema de textura para esfera

As coordenadas das texturas são geradas ao mesmo tempo que os pontos da primitiva, pois estão dependentes deles. Os pontos das texturas apenas usam duas coordenadas, x e y e ambas variam entre 0 e 1, a imagem é vista como um referencial com origem no canto inferior esquerdo e o mapeamento é iniciado no canto superior esquerdo e finalizado no canto inferior direito. Exemplo da criação do primeiro ponto:

```
p = Ponto::Ponto(1 - (camadasV - j)*espv, 1-i*esph, 0);
texturas.push_back(p);
```



Motor 3D

Foram feitas várias alterações ao Motor 3D, nomeadamente na função de parse do XML, função de leitura de ficheiros .3d, adição de variáveis para a luz e texturas, entre outras. As primeiras alterações foram na classe primitiva, na qual adicionamos mais campos.

```
class Primitiva{

    string nome;
    string textura;
    vector<Primitiva> filhos;
    vector<Ponto> pontos;
    vector<Ponto> normais;
    vector<Ponto> texturas;
    Transformacao transformacao;
    //VBOs
    GLuint buffers[3];
    float *v; //Pontos
    float *n; //Normais
    float *tex; //Texturas
    int nvertex;
    int nnormais;
    int ntexturas;
    //Textura:
    unsigned int t, tw, th;
    unsigned int texID;
    unsigned char *texData;
```

Figura 5 Alterações classe Primitiva

Adicionamos uma string textura que serve para guardar o nome da imagem de textura, duas variáveis do tipo vector<Ponto> para guardas os pontos das normais e das texturas, duas variáveis do tipo float * que são usadas nos VBOs das normais e texturas, e algumas unsigned int que são utilizadas nas texturas. Alteramos também a função que prepara as VBOs para suportar agora as normais e texturas, adicionamos uma função para criar a textura e alteramos a função de desenho para agora também incluir as normais e texturas.

Após isto fizemos pequenas alterações nas funções de leitura de ficheiro, na leitura do XML foi adicionada a leitura da luz e na leitura do ficheiro .3d foi alterada para fazer a leitura das 3 primeiras linhas e depois ler 3 vezes o número de pontos e guardar os pontos das primitiva, das normais e das texturas.

Por último alteramos a renderScene para criar o ponto de luz na origem, criar a iluminação em cada planeta e mostrar a sua textura.

Análise de Resultados

Passamos então a mostrar os resultados que obtivemos com o nosso Motor3D para o ficheiro XML que criamos com o Sistema Solar. Este ficheiro foi melhorado relativamente ao realizado na terceira fase, e incluí o Sol, oito planetas, a nossa lua e as luas de Saturno. Optamos também por remover o cometa visto que tivemos dificuldades em visualiza-lo após algumas alterações.

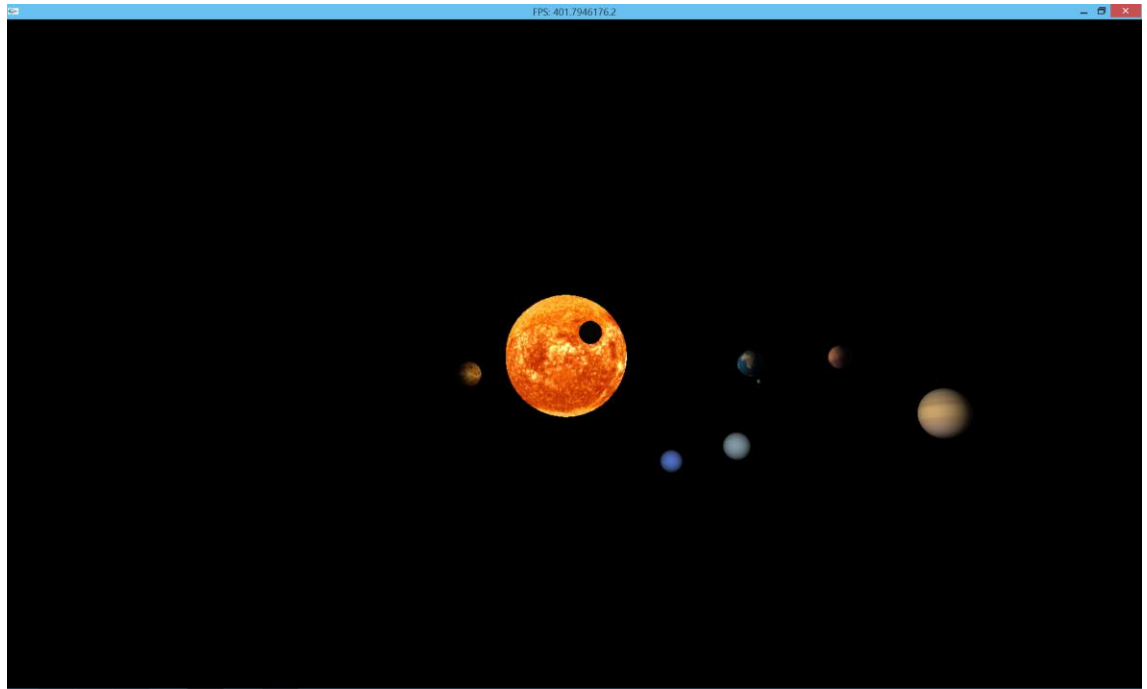


Figura 6 Visualização do Sistema Solar sem órbitas

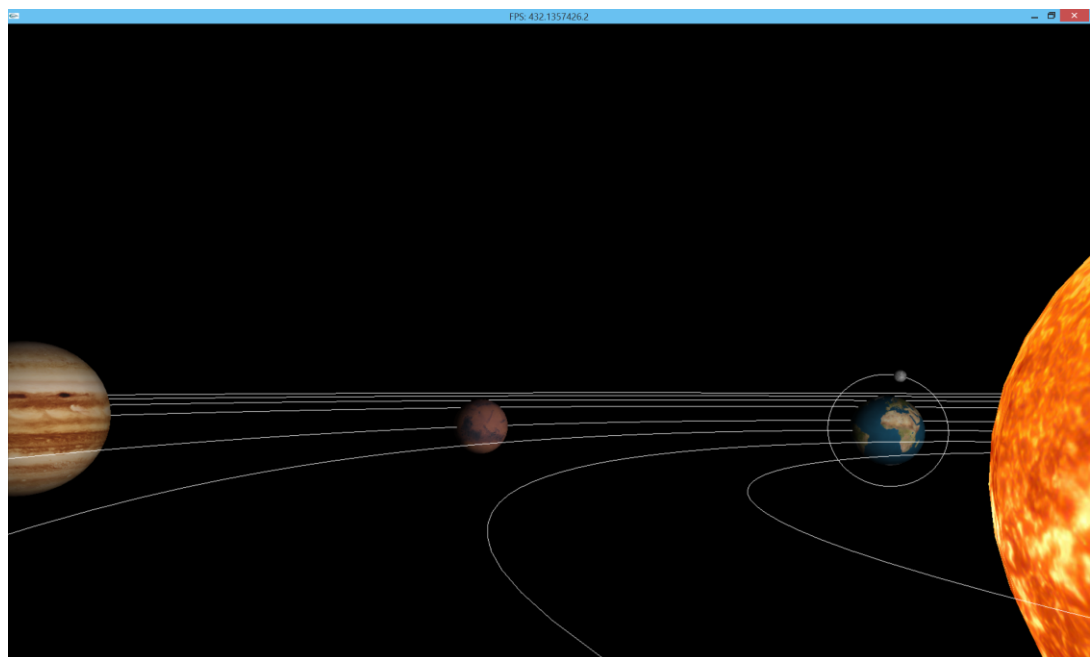


Figura 7 Visualização lateral de um excerto do Sistema Solar

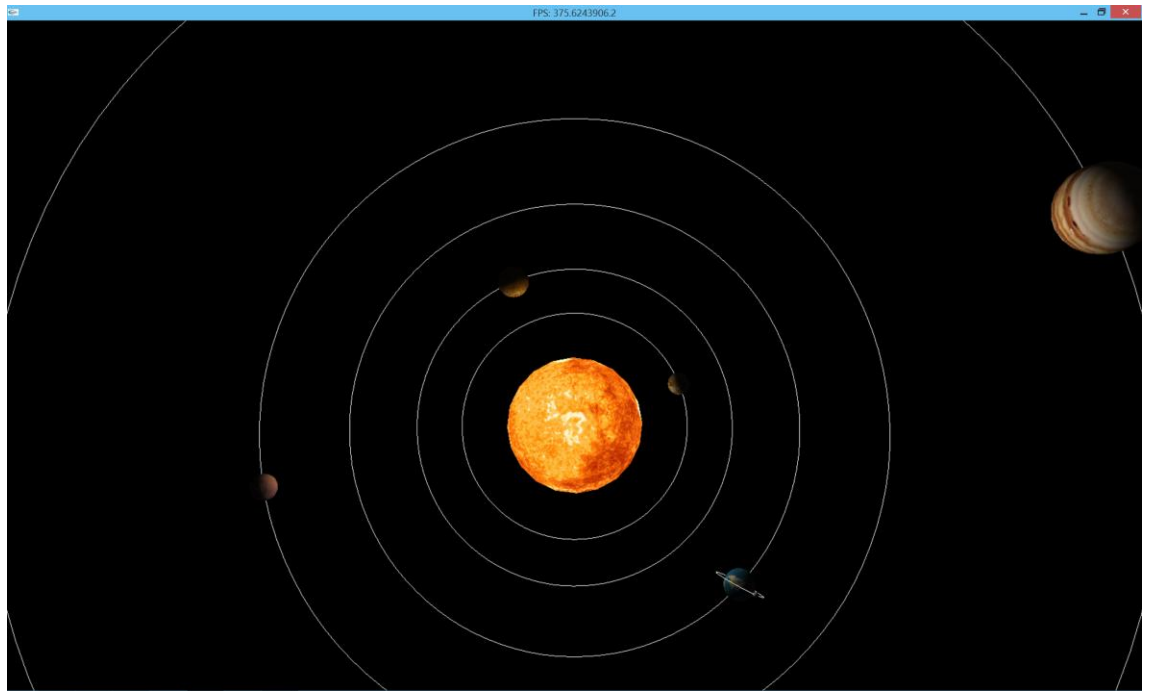


Figura 8 Sistema Solar visto de cima

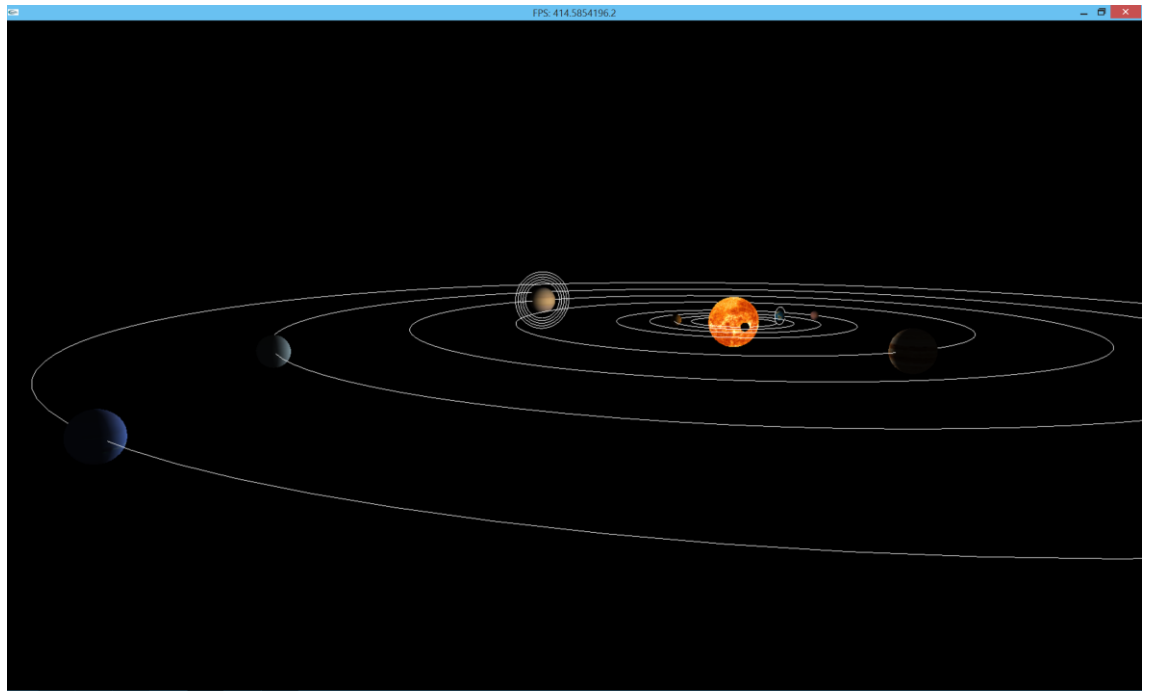


Figura 9 Visualização lateral do Sistema Solar



Conclusão

No fim desta quarta fase, temos um sistema solar dinâmico completo, com um elevado grau realista, que era algo que ansiávamos visualizar desde o início do trabalho. Esta fase permitiu-nos sintetizar o conhecimento das aulas sobre iluminação, sombras e texturas.

As maiores dificuldades que tivemos nesta fase foram na parte do Gerador, os cálculos das coordenadas das texturas implicaram algum tempo e esforço para ficar direito. Outro problema que tivemos foi o teapot, que com a adição da iluminação não foi possível ser visualizado.

Resumindo estamos muito satisfeitos com o trabalho que desenvolvemos e achamos que conseguimos alcançar os objectivos deste trabalho prático.