

Computação Gráfica: Fase Nº4

Universidade do Minho

Mestrado Integrado em Engenharia Informática

Grupo 14

Flávio Martins

A65277



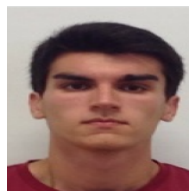
Gonçalo Costeira

A79799



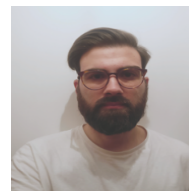
José Ramos

A73855



Rafael Silva

A74264



Contents

1	Introdução	4
1.1	Resumo	4
2	Contextualização	5
2.1	Vetor Normal	5
2.2	Coordenadas de Textura	6
3	Gerador	7
3.1	Calculo da Normal	7
4	Motor	8
4.1	XML	8
4.2	Parsing	8
4.3	Rendering	8
5	Sistema Solar	9
6	Conclusão	11

List of Figures

1	Vetor Normal	5
2	Aplicação de textura	6
3	Sistema Solar	9
4	Vista Superior do Sistema Solar	9
5	Exemplo de Iluminação de planetas	10
6	Apresentação do Planeta Jupiter	10

1 Introdução

1.1 Resumo

Tratando se esta da da ultima fase do projeto pratico, procuramos desenvolver e finalizar o trabalhos desenvolvido previamente nas fases antecedentes. Esta fase traz consigo duas novidades , e para as implementarmos recorreremos a diversas alterações quer no gerador, que nesta fase, passara a ser capaz de partir de uma dada imagem e obter as normais e os pontos de textura para os vários vértices das primitivas geométricas anteriormente criadas. Quer no o motor, que terá novas funcionalidades. Os ficheiros XML passarão a possuir informação relativa a iluminação do cenário tal que teve der alterado o parser deste ficheiros e tambem o modo como o modo de processamento da informação recebida com o intuito de gerar todo o cenário pretendido. Por fim de modo a acomidar o motor a preparação dos VBOs e das texturas durante o processo de leitura do ficheiro de configuração. Deste modo fomos capazes de gerar um modelo do Sistema Solar ainda mais realista em relação ao alcançado na fase anterior.

2 Contextualização

2.1 Vetor Normal

Um vetor normal a um plano é um qualquer vetor que apresente direção ortogonal em relação a qualquer reta pertencente ao plano em causa.

Através deste e de um dado ponto podemos definir um outra reta.

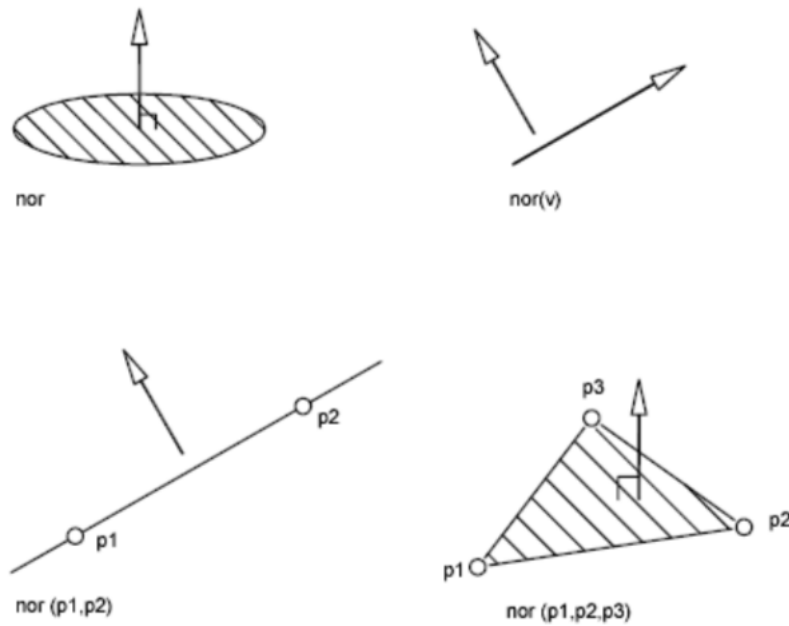


Figure 1: Vetor Normal

2.2 Coordenadas de Textura

Uma textura trata-se de uma imagem com componente RGB e alpha bem como no seu mapeamento gráfico que consiste na aplicação de uma imagem 2D sobre a superfície de um objeto 3D. De modo a aplicar-nos a textura devemos criar uma relação entre os vértices desta, e os do polígono sobre o qual desejamos mapear a textura.

A título de exemplo escolhemos uma imagem quadrada a qual identificamos os vértices de A,B,C,D e um cubo com os vértices de uma face A1,B1,C1,D1, aos quais mapeamos correspondentemente os pontos A a A1, B a B1 e assim sucessivamente.

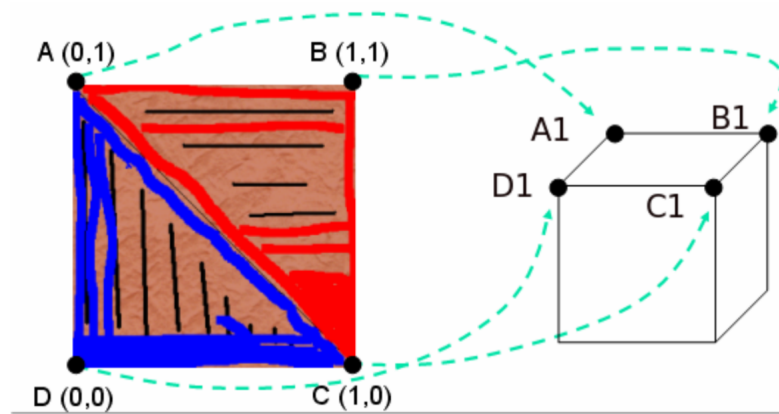


Figure 2: Aplicação de textura

3 Gerador

De modo a adaptarmos-nos a inserção de texturas no processo renderização tivemos de alterar o formato dos ficheiros 3D.

- O numero inteiro 'n' da primeira linha passou a ser $n = \text{numero de triângulos} * 3$ de modo a podermos adaptar a inserção de texturas
- Por cada linha, passamos a ter 8 valores por linha em detrimento de 3;
 - 3 valores para os vértices;
 - 3 valores para a normal;
 - 2 valores para a textura;

3.1 Calculo da Normal

Para calcular o vetor normal a um triângulo dado , em relação ao calculo do desenho do cone e de patch procedemos a implementar uma função “normal” que recebe três variáveis (double) , os vértices do triângulo em causa, devolvendo um array de três valores. Para as outras figuras o calculo é imediato tal que não precisamos de desenvolver um método específico.

- Plane: como se trata e uma superfície plana o calculo dos vetores normais e imediato, sendo assim a textura e mapeada uma vez para cada lado.
- Box: para cada face calculamos o vetor normal e mapeamos como se fosse um Plane.
- Sphere: Como esta e desenhada através de coordenadas esféricas vetores normais correspondem aos seu próprios vertices. tal que para os normalizar, n apenas não os multiplicamos pelo valor do raio , radius. Para as coordenadas de textura, de modo a mapear os vertices num intervalo de zero a um temos de dividir por π e 2π .
- Cone: Este utiliza a função normal para o cálculo dos seus vetores normais enquanto calculamos as coordenadas de textura como o mesmo metido usado para o caso da Sphere.
- Patch: Tal como o Cone, este tampem utiliza a função normal para o cálculo dos seus vetores normais sendo que efetuamos o calculo das coordenadas de textura uma vez para cada patch.

4 Motor

4.1 XML

No ficheiro XML adicionamos as seguintes tags/atributos:

4.2 Parsing

Relativamente trabalho apresentado previamente na terceira fizemos algumas adições e alterações, nomeadamente:

- A classe Light, representante de um ponto de luz a ser adicionado, tal que, a classe Figura passa a ter ainda uma lista de pontos de luzes;
- Como o numero de VBO's necessário por figura aumentou também tivemos de aumentar o tamanho do buffer na classe Modelo sendo necessários 3 respetivamente para : vértice, vetores normais e texturas sendo o carregamento das figuras efetuado na fase de parsing. Na classe Model inserimos os atributos necessários à iluminação e textura de um objeto.

4.3 Rendering

Com as alterações desta fase tivemos também de ativar novas funcionalidades :

```
int main( int argc, char **argv)
{
    ...
    glEnable(GL_LIGHT0);
    glEnable(GL_LIGHTING);
    glEnable(GL_TEXTURE_2D);
    ...
}
```

Depois, em renderScene devemos "ligar" todas as luzes que foram carregadas para a apresentação do produto final sendo que dentro de cada Group de figuras primeiro realizamos as transformações necessárias a cada cena e depois ativamos as funções de luz e textura.

5 Sistema Solar

Apresentamos agora exemplos gráficos do trabalho desenvolvido ao longo do projeto, salientando as novas adições desta fase, a iluminação e a utilização de texturas.

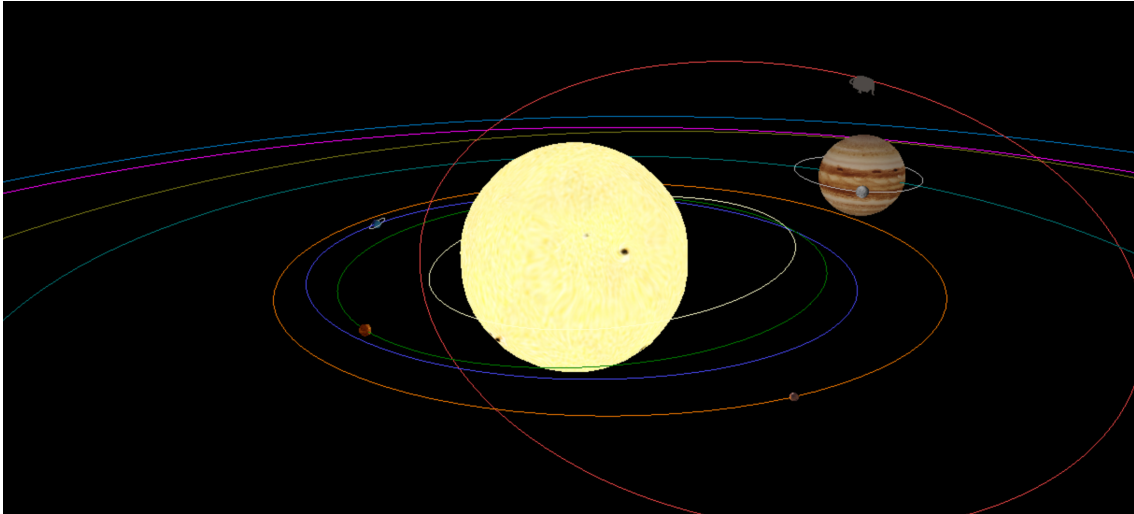


Figure 3: Sistema Solar

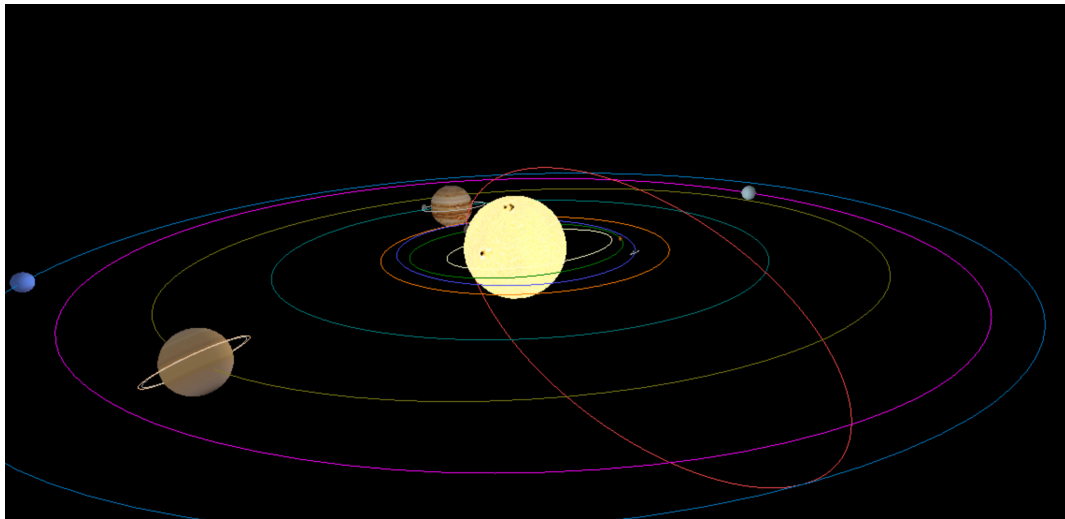


Figure 4: Vista Superior do Sistema Solar

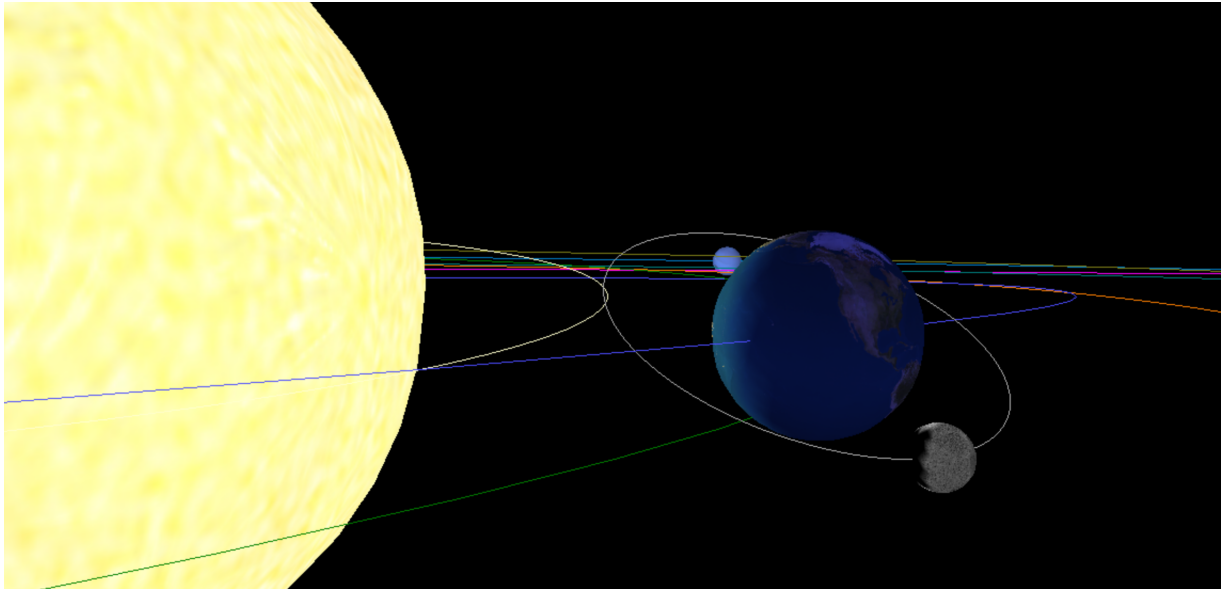


Figure 5: Exemplo de Iluminação de planetas

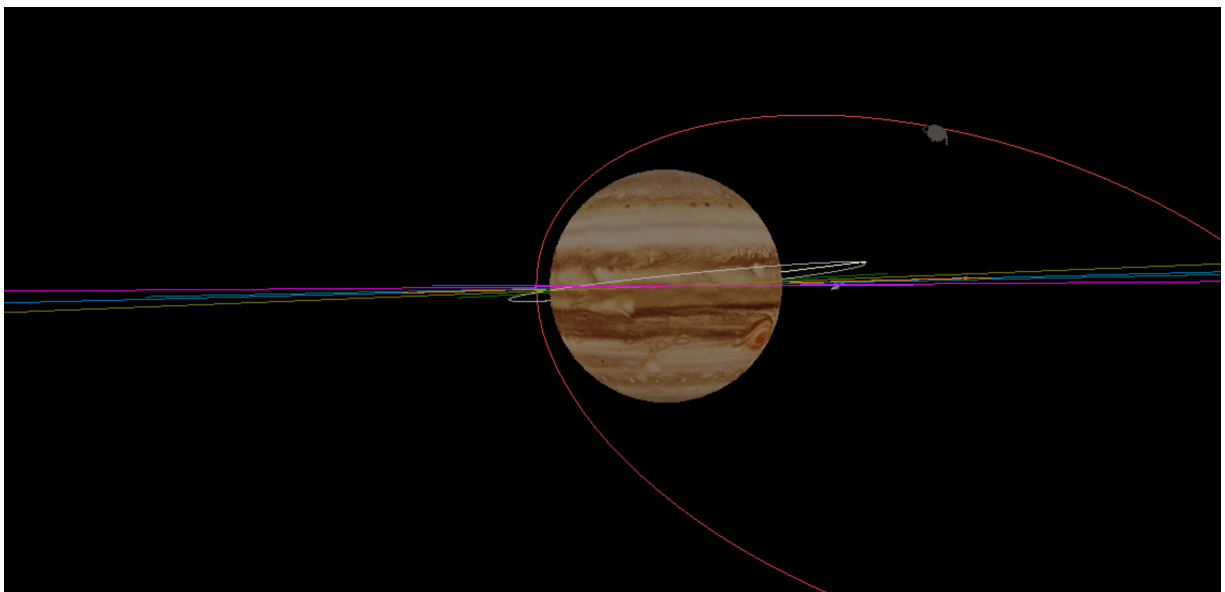


Figure 6: Apresentação do Planeta Jupiter

6 Conclusão

Terminando a quarta e ultima fase do projeto desenvolvida no âmbito da Unidade Curricular de Computação Gráfica , o grupo conseguiu sente-se agora capaz de apresentar e implementar um modelo de renderização que fomos estruturando mesmo ate experimentando novas formas de representar cada vez mais fiel e eficazmente as renderizações desejadas. Em natureza da tarefa proposta sentimos que o método de aplicação fase a fase do conhecimento permitiu-nos consolidar o conhecimento lecionado , bem como desenvolver sentido critico e retrospectivo a eficiência de cada uma das funcionalidades e recursos que nos foi lecionado e apresentado para o projeto. Deste modo , o grupo sente-se realizado, contente com a sua prestação e resultado final obtido .