

**Universidade do Minho**

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

**Unidade Curricular de**

**Computação Gráfica**

Ano Lectivo de 2016/2017

|  |
| --- |
|  |
| Relatório do Trabalho Prático |
|  |
| 2016/2017 |

**Grupo 35**

Pedro Cunha A73958, José Silva A74601, Luís Fernandes A74748, João Coelho A74859

21 de maio de 2017

Índice

[Introdução 2](#_Toc478599577)

[Fase 4 3](#_Toc478599578)

[Conclusão 4](#_Toc478599579)

### Introdução

A quarta e última fase do trabalho prático assinala a conclusão do sistema solar. Pretende-se, agora, iluminar e texturizar as figuras constituintes da cena, aproximando da realidade o sistema solar representado.

Quanto ao parser, pretende-se que o XML possa conter a definição das componentes difusa, especular, emissiva e ambiente da cor, assim como a localização e o tipo das fontes luminosas. A estas tarefas propostas pelo enunciado, decidiu-se acrescentar às capacidades do parser a leitura de um campo “label”, previsto como sendo a legenda da figura na qual se clica.

Aliado a estas tarefas, a melhoria do código C++ e o desenvolvimento, ou aperfeiçoamento, de outras cenas, já apresentadas em fases anteriores, fazem também parte da carga de trabalhos prevista para esta última fase do projeto.

### Fase 4

1. Descrição do processo de leitura

Tal como introduzido anteriormente, além das funcionalidades já incluídas na fase anterior, o parser é agora capaz de ler:

1. **as diferentes componentes da cor**,

sendo que para isso …

1. **as posições das fontes luminosas**,

às quais está associado o tipo de luz …

1. **a legenda atribuída à figura desenhada,**

cuja tag atribuída foi “label”, tendo sido colocada junto da textura/cor e do ficheiro, isto é, na tag “model”, uma vez que é também uma propriedade caracterizadora da figura a ser desenhada.

1. Descrição do processo de normalização e texturização

A definição das normais e coordenadas de textura esteve, naturalmente, dependente da figura em causa. Assim, abordar-se-ão separadamente as figuras que receberam estas novas coordenadas. Antes, mencionar que foram criados mais dois buffers, um para cada tipo de coordenadas. FALAR SOBRE OS BUFFERS!.

1. Toro

* As normais, vetores perpendiculares à superfície, do toro foram simples de definir, dada a sua semelhança com as coordenadas dos vértices. A diferença reside na omissão, nalgumas coordenadas de certos vértices, do produto pelo raio da secção tubular e da substituição do valor do raio exterior por 1, uma vez que o que realmente importa é a direção do vetor e, assim, aproximámo-lo do menor valor.
* A definição das coordenadas de textura passou por fazer variar as componentes em x e em y sobre a forma de múltiplos de 1/anéis e 1/lados, respetivamente, onde os anéis são o número de secções radiais do toro e os lados são o número de lados de cada secção radial. Esta variação acompanha o crescimento dos dois ângulos que auxiliam a construção do toro. O ângulo que acompanha o xé responsável pelo “anel” que se constrói eo que acompanha o yoscila de valor consoante o número de lados já desenhados.

1. Cilindro

* De novo, as coordenadas dos vetores normais não diferem muito das dos vértices. Retirando os raios que surgiam a multiplicar nas coordenadas x e z dos vértices, obteve-se a direção, segundo estes eixos, do vetor. Quanto ao y, para as bases do cilindro o valor é 1, para todos os vértices, e para as laterais é 0, porque o vetor perpendicular à base de um cilindro é vertical, mas no caso das laterais encontra-se na horizontal.
* Já para as coordenadas de textura, decidiu-se que as bases do cilindro seriam independentes das laterais, ou seja, a textura é independentemente atribuída a cada umas das partes. Para as laterais, o processo foi em tudo igual ao do toro, mas neste caso as componentes em x e em y variam sobre a forma de múltiplos de 1/sides e 1/stacks, respetivamente. Para as bases, imaginou-se um círculo dentro de um quadrado (textura quadrada) e procurou-se definir as coordenadas de textura de modo a que seja essa região circular a textura da base. Para isso, à coordenada x do vetor normal somou-se 1, multiplicando-se depois a soma por 0.5, processo que se repete para obter a segunda coordenada de textura, desta feita através do z do vetor normal. Assim, garante-se que o valor do x e do y está compreendido entre 0 e 1.

1. Rubi

* Dado que o rubi, na construção da sua parte superior, utiliza a função que origina o cilindro, as normais e coordenadas de textura estão, desde logo, parcialmente atribuídas. Consequentemente, teremos coordenadas de textura independentes para a parte superior do rubi (cilindro) e para a parte inferior (cone invertido).
* Quer para as normais da parte inferior, quer para as coordenadas de textura, o processo foi equivalente ao usado para as partes laterais do cilindro.

1. Esfera
2. Plano
3. Caixa
4. Bezier

### Conclusão

Concluído este projeto, importa destacar o quanto importante foi para a aprendizagem do grupo. Desde o adquirir de rotinas numa linguagem até então não estudada, o C++, até à aplicação e prática de conhecimentos adquiridos em aulas teóricas.

Particularizando a esta fase do projeto, destaca-se o cumprimento de praticamente todos os objetivos estabelecidos para esta fase. Em falta fica apenas aquela que foi a meta adicional estabelecida pelo grupo, ou seja, a legenda das figuras. Apesar de funcional, um problema associado às texturas leva a que nem sempre, clicando no planeta, a legenda seja exibida. Perante isto, e sabendo que a legendagem tem um impacto adicional de duas condições if por frame, optou-se por enviar em comentário o código alusivo a esta aptidão adicional do programa, para que possa ser exposta ao professor na apresentação.