
FASE 4

Normais e Coordenadas de Texturas



A95414
Artur Luís



A95835
Bianca Vale



A95454
Lara Ferreira



A95111
Luís Ferreira

Índice

1	Introdução	1
2	Alterações à estrutura do projeto	2
2.1	Generator	2
2.1.1	<i>generator.cpp</i>	2
2.2	Engine	2
2.2.1	<i>engine.cpp</i>	2
2.2.2	<i>luz.cpp</i> e <i>luz.h</i>	2
3	Alterações ao XML	3
4	Texturas e Iluminação	4
5	Conclusão	7

1 Introdução

Nesta última fase do trabalho temos como objetivo gerar coordenadas de textura e as normais para cada vértice. Assim, vamos precisar de implementar a luz e a textura no nosso *Engine* e aplicar as coordenadas de textura e normais a partir das *model files*.

No final desta fase vamos ter um modelo do Sistema Solar com iluminação e texturas.

2 Alterações à estrutura do projeto

2.1 Generator

Nesta fase alteramos o *Generator* para ser capaz de gerar as normais e coordenadas de textura para os vértices.

2.1.1 *generator.cpp*

Neste ficheiro, alteramos as funções de geração de formas de modo a que sempre que se adiciona um ponto, é também adicionada a normal e uma coordenada de textura.

Além disso, antes de escrever para o ficheiro, é normalizada a normal através da função *normalizar_normal*.

2.2 Engine

2.2.1 *engine.cpp*

Este ficheiro foi alterado para guardar as normais e as coordenadas de textura de modelos criados com VBOs.

O ficheiro é agora capaz de carregar texturas, garantindo que não existem texturas repetidas. A textura é convertida para o formato RGBA.

Parsing

2.2.2 *luz.cpp* e *luz.h*

Estes ficheiros foram adicionados para processar os vários tipos de luz presentes no cenário.

São declarados as 3 classes que representam 3 tipos de luz diferentes (*Luz_Ponto*, *Luz_Direcional* e *Luz_Ambiente*).

No caso da *Luz_Ponto*, precisamos somente de guardar a posição onde se encontra.

Já para a *Luz_Direcional* é necessário guardar a sua direção.

Para a *Luz_Ambiente* é necessário guardar a posição, direção e o seu ângulo de abertura.

A função *aplicar* é quem aplica as componentes ambiente, difusa e especular. Dependendo do tipo de luz aplica então as características necessárias.

3 Alterações ao XML

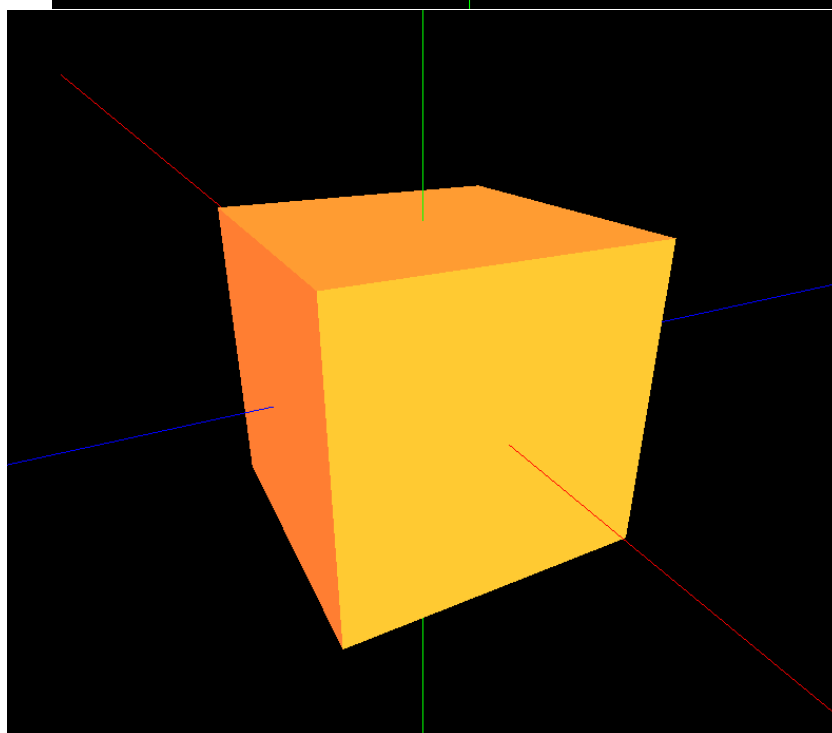
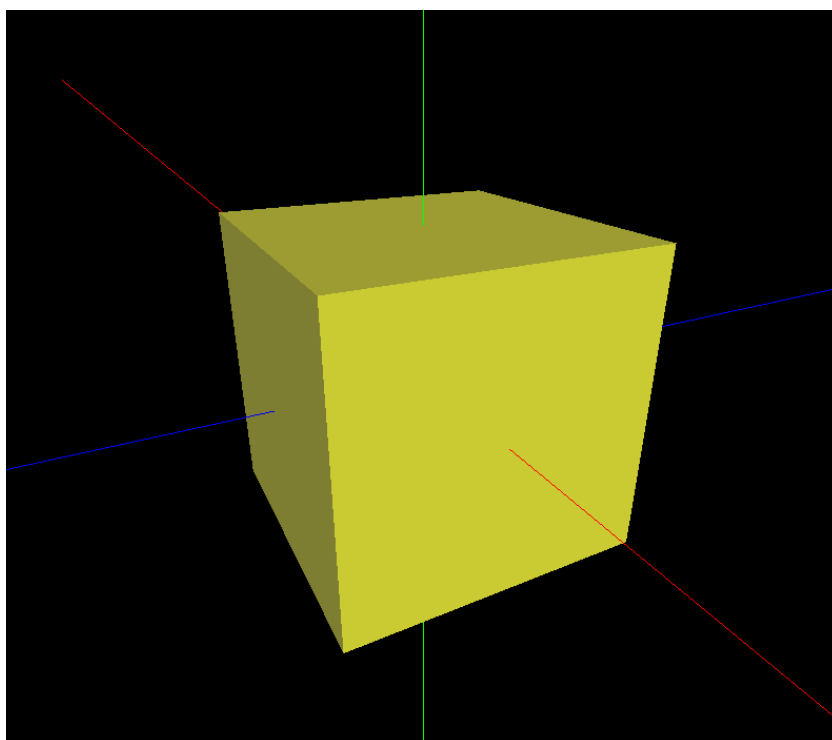
Para suportar as novas funcionalidades foi necessário alterar novamente a estrutura do ficheiro XML.

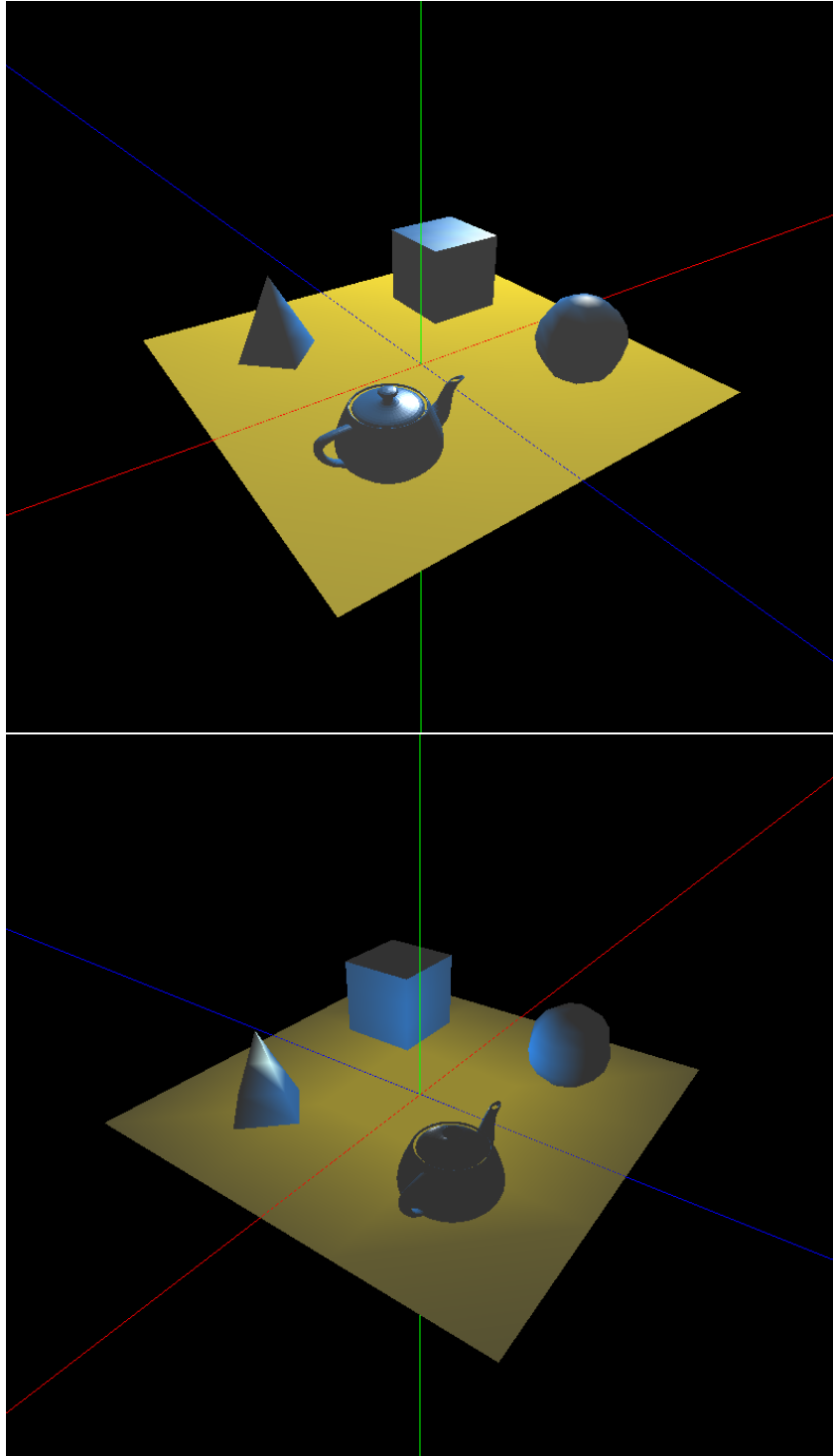
Como agora necessitamos de obter a iluminação, surgiu o elemento *lights*, que possui em si vários elementos *light*.

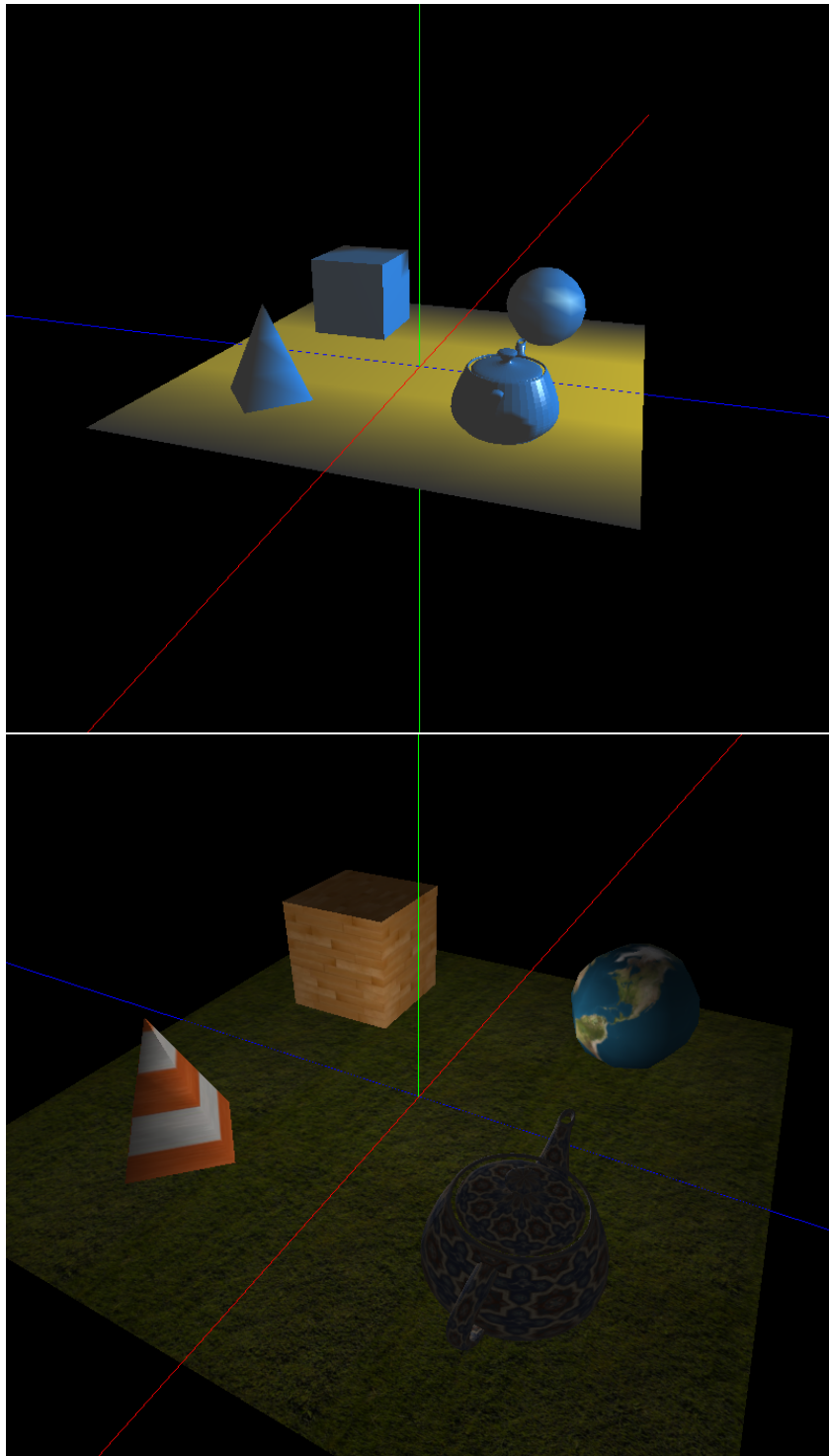
Para cada elemento *light* existe um atributo *type* que indica qual o tipo de luz. As componentes ambiente, difusa e especular podem ser especificadas.

No elemento *models* encontramos o atributo *file* que nos indica o caminho para o ficheiro de onde se carregam as texturas.

4 Texturas e Iluminação







5 Conclusão

Concluída a última fase do projeto, achamos que conseguimos cumprir com grande parte dos desafios propostos, embora não os tenhamos superado a todos.

Foi sentido o aumento da complexidade do trabalho a cada fase, mas também a evolução do nosso projeto.

No geral, o projeto foi bastante desafiante, mas conseguimos obter um resultado final bastante interessante.