

UNIVERSIDADE DO MINHO
DEPARTAMENTO DE INFORMÁTICA



TRABALHO PRÁTICO
CIÊNCIAS DA COMPUTAÇÃO - LCC

Computação Gráfica

GRUPO 9



Gonçalo Rodrigues A91641



Hugo Sousa A91654

Junho de 2023

Conteúdo

1	Introdução	2
1.1	Contextualização	2
1.1.1	Estrutura do Relatório	2
2	Análise e Especificação	3
2.1	Descrição e Enunciado	3
2.1.1	Generator	3
2.1.2	Engine	3
2.2	Requisitos	4
2.2.1	Generator	4
2.2.2	Engine	4
3	Concepção e Desenho da Resolução	5
3.1	Iluminação e texturas	5
4	Codificação	6
4.1	Código e ferramentas utilizadas	6
4.2	Alterações ao generator	6
4.3	Alterações à engine	6
5	Testes	8
5.1	Sistema Solar	8
6	Conclusão	11

Capítulo 1

Introdução

1.1 Contextualização

Com seguimento no que temos desenvolvido nas fases anteriores, chegou o momento de fazer com que o generator seja capaz de produzir as normais dos modelos gerados e coordenadas de textura. Num segundo momento foram pedidas alterações à engine, onde deveríamos aplicar a iluminação e texturas definidas na config.xml.

1.1.1 Estrutura do Relatório

Neste relatório será descrita a interpretação efetuada do enunciado do trabalho prático e a abordagem usada para resolver os problemas propostos.

Capítulo 2

Análise e Especificação

2.1 Descrição e Enunciado

2.1.1 Generator

Tendo em conta o enunciado desta fase, chega a altura de calcular as normais dos modelos utilizados na demo final, ou seja, é necessário que, ao criar um certo modelo, colocar a informação referente a coordenadas de textura, assim o ficheiro que contém os triângulos a ser desenhados contém, agora, as coordenadas de textura e ainda às normais.

2.1.2 Engine

No caso da engine foi necessário adaptar a forma como o ficheiro xml é lido e como guardamos a nova informação, pois existem agora 3 novos elementos a ser lidos, as textures, colors e luzes. Dessa forma a demo final deve ser capaz de representar as texturas e a iluminação aplicada.

2.2 Requisitos

2.2.1 Generator

No generator não foi aplicada nenhuma função nova, apenas calculamos as normais e as coordenadas de textura guardando essa informação no mesmo ficheiro que posteriormente será lida e aplicada.

2.2.2 Engine

A engine é capaz de reconhecer os novos tipos, texture, color e light. Todas as funcionalidades anteriores foram mantidas, mas neste momento deve-se aplicar texturas aos objetos passados na config e aplicar iluminação na demo final.

Capítulo 3

Concepção e Desenho da Resolução

3.1 Iluminação e texturas

- `<texture file = "image.jpg" / >`
- `<color >`
 - `<diffuse R = "200" G= "200" B = "200" / >`
 - `<ambient R = "50" G= "50" B = "50" / >`
 - `<specular R = "0" G= "0" B = "0" / >`
 - `<emissive R = "0" G= "0" B = "0" / >`
 - `<shininess value = "0" / >`
- `< /color >`
- `<lights >`
 - `<light type="point" posX="0" posY="10" posZ="0" / >`
 - `<light type="directional" dirX="1" dirY="1" dirZ="1" / >`
 - `<light type="spotlight" posX="0" posY="10" posZ="0" dirX="1" dirY="1" / >`
- `< /lights >`

Capítulo 4

Codificação

4.1 Código e ferramentas utilizadas

Para a execução deste trabalho foi utilizada a linguagem C++ tanto na construção do *generator* como da *engine*. Para permitir a leitura da configuração do motor foi usada a biblioteca *tinyXML2*. Foi ainda usada a biblioteca **DevIL** para trabalhar com as imagens referentes das texturas.

4.2 Alterações ao generator

Tal como referimos em cima, a única alteração feita no generator foi a adição do resultado do cálculo das normas, que são calculadas fazendo o produto cruzado entre dois vetores que descrevem o plano, e as coordenadas das texturas de acordo com a esfera.

4.3 Alterações à engine

Nesta última fase foram feitas várias alterações na engine. De início foi melhorado o parser da config para poder reconhecer os vários tipos de luzes e os seus parâmetros. Temos três tipos de luzes disponíveis para utilização, um ponto de luz que recebe como parâmetro as suas coordenadas no plano XYZ, uma luz direcional que recebe a sua direção e por fim uma “spotlight” que recebe como parâmetros as suas coordenadas, a sua direção e o seu ângulo. De seguida foi também adicionada a capacidade de reconhecer ficheiros de textura, de modo a adicionar texturas na nossa cena. Para este efeito foi necessário mais uma vez melhorar o parser da config XML.

Por fim e de modo a poderem ser aplicadas tanto as luzes como as texturas foram adicionados os métodos necessário para tal. No caso das luzes, estas têm de ser criadas e ligadas, então no momento em que a informação é lida esta é guardada

numa estrutura que armazena o seu id e as suas informações. No caso das texturas, estas são primeiro carregadas para o programa, utilizando a biblioteca **DevIL** e de seguida será armazenado um id que identifica a mesma.

É também de salientar que as estruturas de dados foram simplificadas nesta fase pois não era necessário existir uma árvore para guardar cada grupo (grupo que armazena a informação das transformações). Para substituir esta árvore foi então adicionado um vetor que guarda os models, onde estes por si são uma estrutura identificada pelo nome do ficheiro associado ao objeto, que armazena um grupo com as transformações, as suas cores e um id de textura.

Por fim para permitir efeitos de luz e a aplicação das texturas foram adicionados dois novos vbos um para guardar os vértices das texturas e outro para as normas.

Capítulo 5

Testes

5.1 Sistema Solar

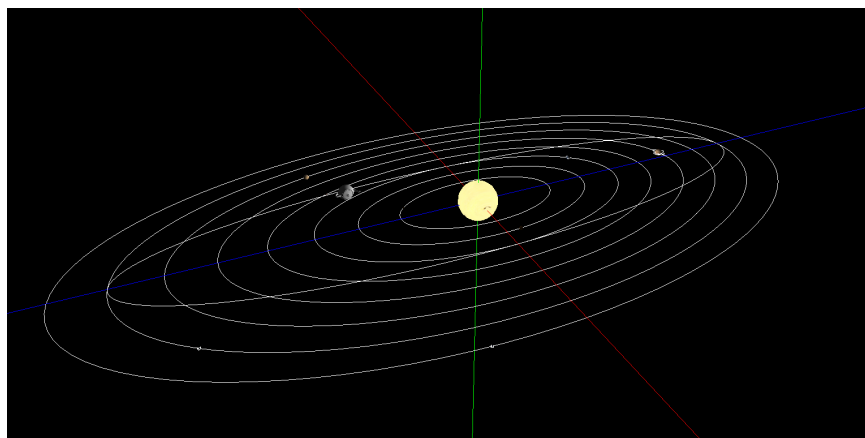


Figura 5.1: Representação do Sistema do Solar.

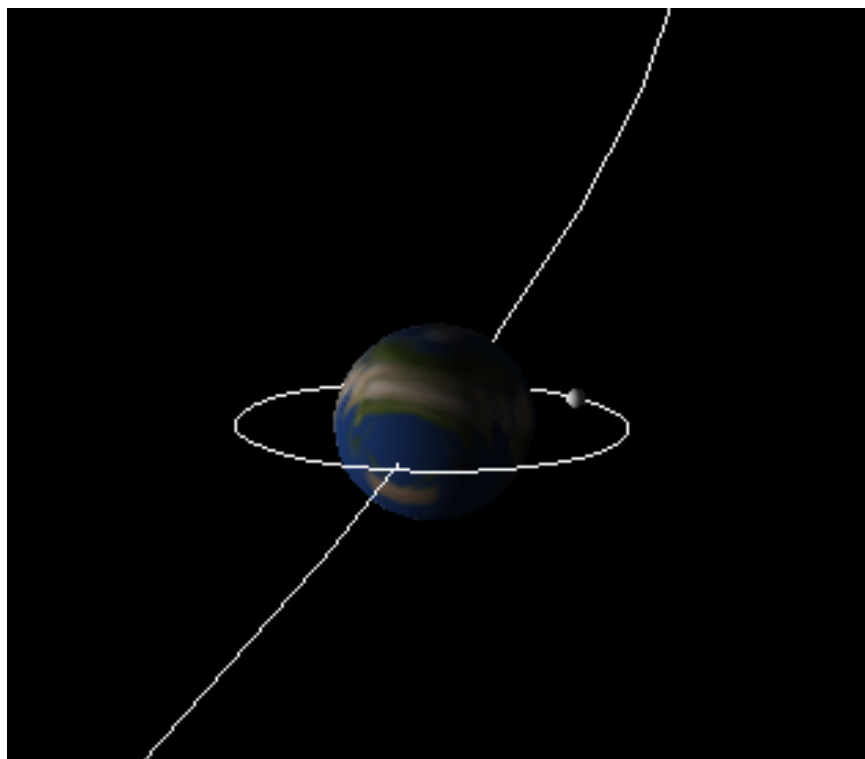


Figura 5.2: Representação da Terra.

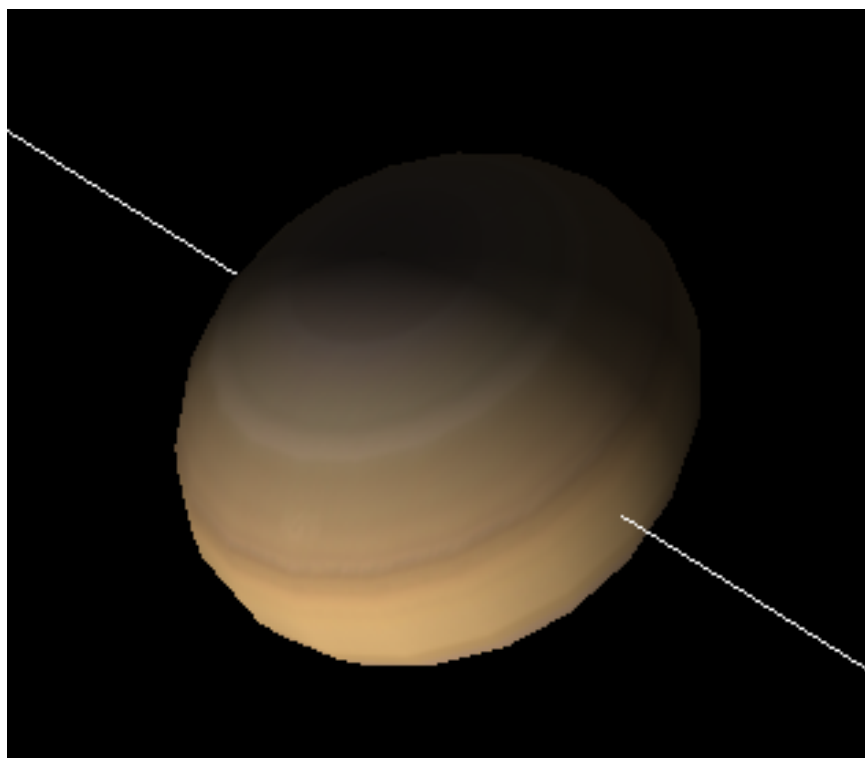


Figura 5.3: Representação de Saturno.

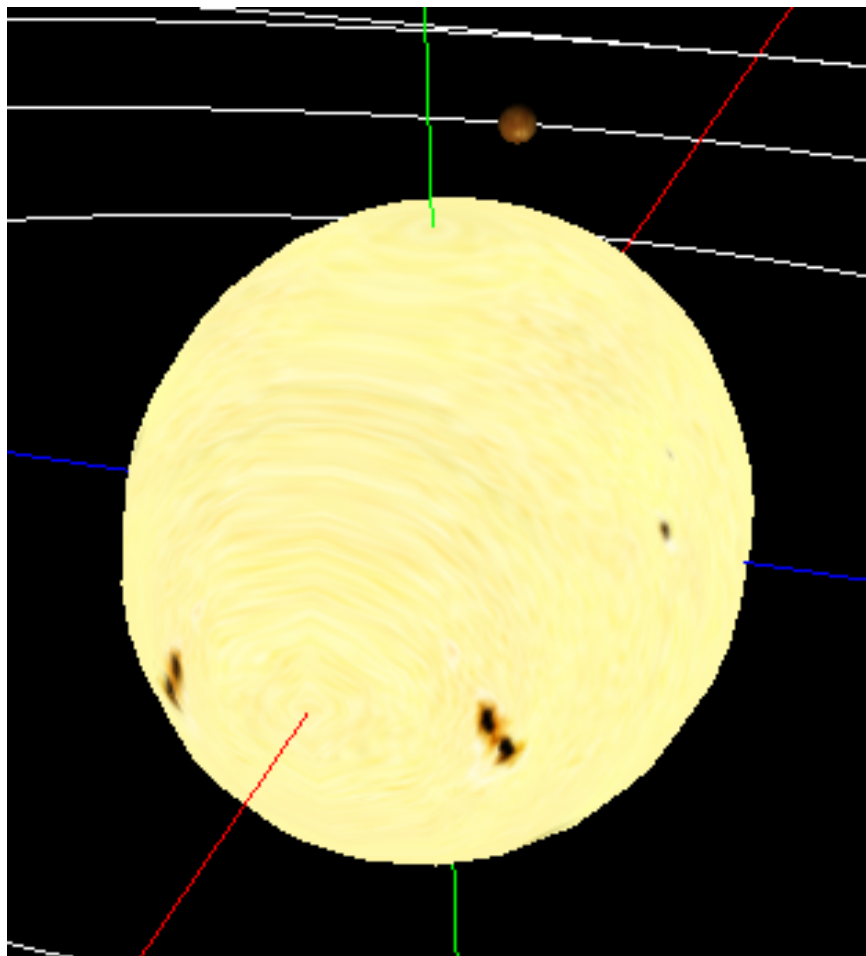


Figura 5.4: Representação do Sol.

Capítulo 6

Conclusão

Em suma, foi nesta fase que sentimos mais dificuldades devido ao que foi pedido no enunciado, pois é a parte da cadeira que menos nos sentimos à vontade, também devido à fase do semestre em que nos encontramos e sermos apenas dois elementos no grupo. Apesar disso, cumprimos os objetivos propostos nas datas propostas. Ficou por implementar o align que foi pedido na terceira fase, não percebes como deveria funcionar e por isso não funciona apesar de estar presente na config.