



Universidade do Minho
Escola de Engenharia

Computação Gráfica

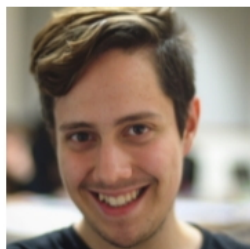
Fase 2 - Transformações geométricas

Grupo 52

29 de Março de 2020
MiEI - 3º Ano - 2º Semestre



Beatriz Rocha A84003



Filipe Guimarães A85308



Gonçalo Ferreira A84073



José Mendes A75481

Conteúdo

1	Introdução	3
2	Ficheiro xml	4
2.1	Representação do sistema solar	4
2.2	Ordem das Transformações nos Planetas	4
2.3	Representação das Luas	4
3	Engine	6
3.1	Processar o ficheiro xml	6
3.2	Transformações	6
3.2.1	Grupos	7
4	Teste do Sistema Solar	8
5	Conclusão e Análise de Resultados	10

Lista de Figuras

2.1	Desfazer da rotação aplicada ao planeta	5
3.1	Exemplo de grupo xml a processar	6
3.2	Estrutura <i>transformações_3d</i>	6
4.1	Sistema solar de perto	8
4.2	Sistema solar	9

Capítulo 1

Introdução

O objectivo desta fase passa primeiro por conseguir representar um sistema solar estático num ficheiro XML utilizando modelos de primitivas construídas da primeira fase e um conjunto de transformações geométricas. A segunda parte traduz-se na criação de uma estrutura que represente cada transformação geométrica possível e na aplicação de uma ou série de transformações a um conjunto de objectos.

Capítulo 2

Ficheiro xml

2.1 Representação do sistema solar

O nosso modelo de sistema solar é constituído pelo sol, os planetas e as luas. Considerando a estrutura de XML que nos foi dada optamos por colocar o sol no grupo mais geral, os planetas em grupos dentro do grupo do sol, e as luas em grupos dentro dos grupos dos planetas. Definimos assim porque faz sentido representar os planetas em relação à posição do sol e as luas em relação à posição dos planetas.

2.2 Ordem das Transformações nos Planetas

As transformações são aplicadas aos modelos do grupo em que se encontram e subgrupos. Neste sistema faz sentido aplicar transformações, escalas e rotações pelo que tivemos de respeitar a ordem das transformações a aplicar. Considerando a teoria da ordem em que devemos aplicar as transformações aos planetas são: translate, depois scale e por último rotate.

2.3 Representação das Luas

Como quisemos aplicar uma translação à lua então tivemos que ter atenção à rotação anteriormente aplicada aos planetas, isto porque as transformações dos grupos são herdadas pelos subgrupos. Caso aplicássemos a translação na lua após a rotação do planeta, os resultados não seriam os esperados, pelo que tivemos primeiro que desfazer a rotação do planeta, aplicando na lua uma rotação que é a inversa da rotação do planeta.

```

<group>
  <!-- Terra -->
  <translate X="7.96" Z="0" />
  <scale X="0.15" Y="0.15" Z="0.15" />
  <rotate axisX="0.0" axisY="0.0" axisZ="1.0" angle="23.45" />
  <models>
    <model file="sphere.3d" />
  </models>
</group>
<group>
  <!-- Lua -->
  <rotate axisX="0.0" axisY="0.0" axisZ="1.0" angle="-23.45" />
  <translate X="3.0" Z="0.0" />
  <scale X="0.3" Y="0.3" Z="0.3" />
  <models>
    <model file="sphere.3d" />
  </models>
</group>
</group>

```

Figura 2.1: Desfazer da rotação aplicada ao planeta

Como as escalas aplicadas foram uniformes não nos tivemos que preocupar com a ordem das rotações relativamente às escalas.

Capítulo 3

Engine

3.1 Processar o ficheiro xml

Como referimos anteriormente o nosso xml tem agora transformações geométricas sobre modelos. Para isso tivemos de alargar o conhecimento de tags xml para conseguirmos retirar os valores de cada transformação.

```
<group>
  <!-- Saturno -->
  <translate X="71.70" Z="0" />
  <scale X="0.35" Y="0.35" Z="0.35" />
  <rotate axisX="0.0" axisY="0.0" axisZ="1.0" angle="26.73" />
  <models>
    <model file="sphere.3d" />
  </models>
</group>
```

Figura 3.1: Exemplo de grupo xml a processar

Recorrendo mais uma vez ao *tinyXML*, processamos as tags com a ajuda da função *elemento_atributos* identificando cada uma das transformações e chamando a função respetiva.

3.2 Transformações

De forma a simplificar o processo de agregação, bem como depois desfazer das mesmas, decidimos então pensar nelas como um vetor de *transformações_3d*.

```
struct transformacao_3d {
    float var[4][4];
};
```

Figura 3.2: Estrutura *transformações_3d*

Como se pode ver na figura anterior criamos uma estrutura que contém uma matriz de pontos que representa uma determinada transformação. Aplicamos os conceitos presentes na sebenta disponibilizada bem como nas aulas teóricas

para podermos agregar as diversas transformações (translate, scale e rotate) na mesma matriz. Para isso adicionamos ao *auxiliar_engine* funções para tratar cada uma das transformações.

3.2.1 Grupos

Para solucionar a questão dos grupos e sub-grupos do XML, foi necessário acrescentar um vector que retém o histórico das transformações. Este vector atualiza exclusivamente em dois casos, na saída e entrada de um grupo. Na entrada de um grupo guarda o tamanho do vector de transformações, assim mantendo uma referência ao estado das transformações desse grupo. À saída é consultado o último valor no vector de histórico, para descartar todas as transformações até esse valor, e por sua vez descartando também esse valor de histórico.

Capítulo 4

Teste do Sistema Solar

Com vista a testar o funcionamento da engine com o xml criado para a representação do sistema solar decidimos gerar uma esfera recorrendo ao generator com raio 1 e 10 slices e stacks.

```
$ ./generator sphere 1 10 10 sphere.3d
```

Criamos uma pasta que contém tanto o ficheiro xml como a *sphere.3d*. Para correr temos apenas de entrar na pasta.

```
$ cd solarSystem
```

Compilar a engine na mesma com os seguintes comandos.

```
$ cmake ../engine/  
$ make
```

Agora basta passar como argumento o ficheiro xml do sistema solar.

```
$ ./engine SolarSystem.xml
```

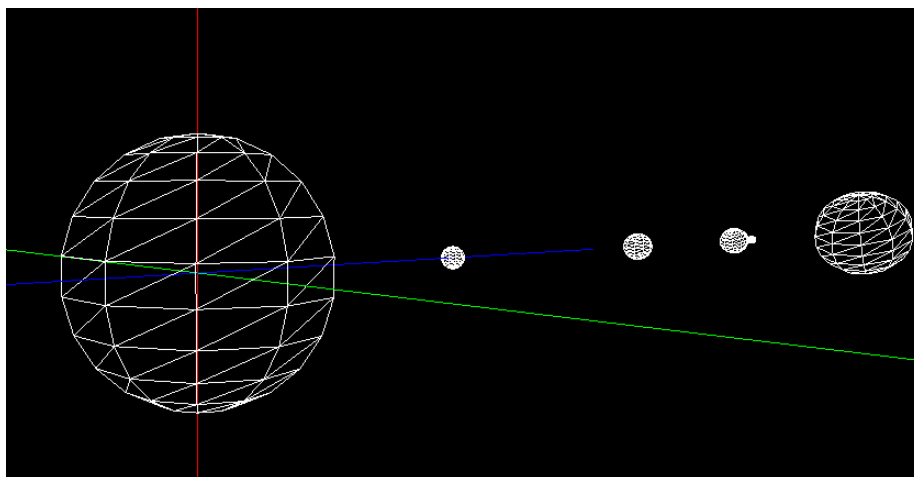


Figura 4.1: Sistema solar de perto

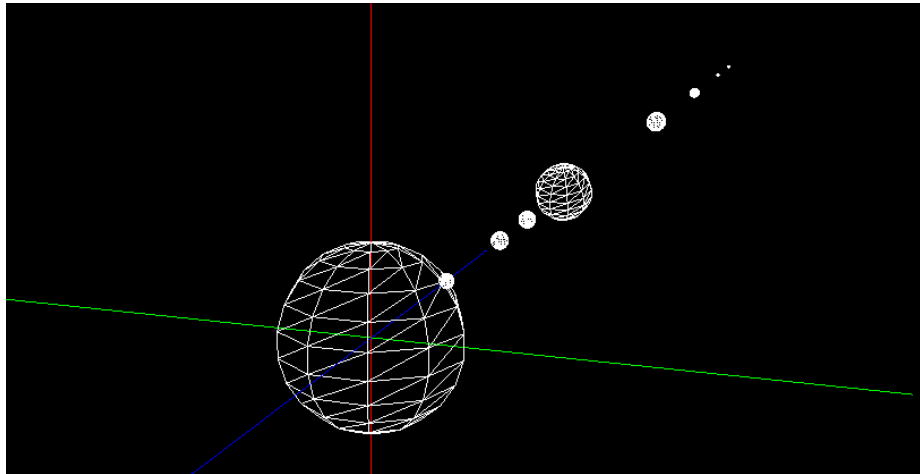


Figura 4.2: Sistema solar

Capítulo 5

Conclusão e Análise de Resultados

Nesta fase aprendemos a desenhar um modelo de um sistema composto por diversos elementos diferentes, a representar esses elementos do sistema como um conjunto de pequenos sistemas dentro uns dos outros e a fazer o processo de aplicação de transformações geométricas através da utilização de matrizes, da operação de composição dessas estruturas e de um vetor que guarda matrizes.