

UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

Computação Gráfica - Fase 2
Grupo 38

Bernardo Emanuel Magalhães Saraiva - A93189

José João Cardoso Gonçalves - A93204

Mariana Rocha Marques - A93198

Rui Filipe Coelho Moreira - A93232

Ano Lectivo 2021/2022

Conteúdo

1	Introdução	3
1.1	Contextualização	3
1.2	Objetivos da 2 ^a fase	3
1.3	Alterações efectuadas na 2 ^a fase	3
2	Estrutura do código	4
2.1	Engine	4
2.2	XML Parser	4
2.3	Sistema Solar	4
2.4	Torus	7
3	Conclusão	10

1 Introdução

1.1 Contextualização

O presente relatório tem como objetivo documentar e formalizar o procedimento efetuado para a implementação da segunda fase do Trabalho Prático de Computação Gráfica, constituinte do 2º semestre do 3º ano de Engenharia Informática.

A presente fase do Trabalho Prático dá seguimento ao que foi proposto na fase anterior, pelo que se trabalhou tomando como estrutura base o código anteriormente desenvolvido.

1.2 Objetivos da 2ª fase

Para a segunda fase do Trabalho Prático da UC de Computação Gráfica, foi proposto aos alunos a actualização do *engine* de modo a ser possível aplicar transformações geométricas, mais propriamente translações, rotações e mudanças de escala face aos modelos desenvolvidos.

Para tal, é necessário alterar a função de leitura do XML de forma a ler os novos parâmetros, entre eles *translate*, *rotate* e *scale*. É necessário também implementar um sistema hierárquico, onde em cada nodo "*group*", é possível conter modelos (acrescentando a figura *Torus* aos implementados na fase 1) e um conjunto de transformações, para além de possíveis sub-nodos com as suas próprias transformações e modelos. Os sub-grupos, para além das suas transformações, herdaram as transformações do seu grupo mãe.

Para esta fase, foi também necessário o desenvolvimento de um modelo estático do Sistema Solar, incluindo o sol, planetas e luas definidos em hierarquia.

1.3 Alterações efectuadas na 2ª fase

Para alcançar os objetivos propostos para a segunda fase, as alterações surgiram essencialmente ao nível do *engine*. Para além do ficheiro anteriormente referido, foram implementadas novas classes, que muito superficialmente se poderá dizer que têm como objetivo de guardar estruturas que possam ser úteis para o reconhecimento e desenho das novas funcionalidades. Estas novas classes serão descritas e abordadas em detalhe no próximo capítulo.

Além do referido, também foram adicionadas algumas movimentações de câmara novas para auxiliar na visualização do sistema solar. Estas funcionalidades permitem aproximar e distanciar a câmara do referencial, sendo possível ter uma percepção da totalidade do sistema solar, ou contemplar o detalhe das figuras que o constituem.

2 Estrutura do código

2.1 Engine

O *Engine* corresponde ao módulo que trata da leitura dos ficheiros *XML* e a visualização da representação gráfica proveniente dos ficheiros *.3d* resultantes do *generator*. De acordo com o referido em 1.2, o Engine desenvolvido na primeira fase necessitou de sofrer algumas alterações, para que, além de cumprir com o enunciado, pudesse melhorar a sua interpretação, bem como as suas funcionalidades. Para alcançar este objetivo, foi desenvolvida a função *drawModels*, que em conjunto com a classe *tree.cpp* permite aplicar as transformações geométricas aos modelos desejados.

2.2 XML Parser

Na fase 2, a dificuldade no *parsing* do XML foi aumentada, dado que surgiram novas funcionalidades como por exemplo o sistema hierárquico (grupos). Devido a estas mudanças foram adicionadas algumas classes para suprir as novas necessidades. Entre as classes criadas, destaca-se a **Tree** que contém toda a informação do XML organizada da seguinte forma:

- Câmara - Possui os valores presentes no XML para posicionar a câmara corretamente.
- Group - Possui os modelos a desenhar, bem como os seus subgrupos e as transformações a aplicar a cada um destes. Relativamente aos modelos, estes são armazenados através da classe *Figure*, que é constituída por um vetor de pontos, possibilitando desta forma a sua representação através do Engine.

De forma a armazenar a informação do XML nestas novas classes, foi então necessário redefinir a função *readFile* que atua sobre os diferentes cabeçalhos no XML: *Camera*, *Group*, *Transform*, *Models* e *Model*. Através desta função, é aberto o ficheiro a ser representado e lidas as configurações relativas à câmara através da função *readCamera* e de seguida é lido o *group*, bem como todos os seus subgrupos através da chamada recursiva da função *groupParser*, que possibilita a leitura de todos os modelos, bem como das transformações associadas a cada grupo.

2.3 Sistema Solar

De maneira a obter um modelo do Sistema Solar que seja o mais próximo da realidade, o grupo teve em conta a escala real. Porém, foram reguladas algumas distâncias e escalas com o objetivo de facilitar a visualização por completo do Sistema Solar, bem como o detalhe envolvido.

Relativamente ao desenho dos planetas para formar uma representação do sistema solar, todos os modelos partem de uma primitiva inicial da esfera e do *torus* (se necessário). As-

sim, com o objetivo de representar as dissemelhanças de tamanho que os caracterizam, são efetuadas as transformações necessárias aquando da sua representação.

Em seguida demonstra-se um excerto de código que serve de exemplo ao referido, e que trata do desenho do Planeta Saturno, sendo dividido no desenho de uma esfera correspondente ao planeta em si e ao seu anel.

```
1 <world>
2   <camera>                                <!--configuracao da camara-->
3     <position x="150" y="50" z="50" />
4     <lookAt x="0" y="0" z="0" />
5     <up x="0" y="1" z="0" />
6     <projection fov="60" near="1" far="1000" />
7   </camera>
8
9   <group>
10    <group>                                <!--desenho do planeta saturno-->
11      <transform>
12        <translate x="63" y="0" z="0"/>
13        <scale x="3.5" y="3.5" z="3.5"/>
14      </transform>
15      <models>
16        <model file="sphere.3d"/>
17      </models>
18    </group>
19    <group>                                <!--desenho do anel de saturno-->
20      <transform>
21        <rotate angle="10" x="1" y="0" z="0" />
22        <translate x="63" y="0" z="0" />
23        <scale x="7" y="0" z="7"/>
24      </transform>
25      <models>
26        <model file="torus.3d" />
27      </models>
28    </group>
29  </group>
30 </world>
```

Para uma maior percepção do objetivo de cada transformação, a seguinte imagem serve de exemplo para explicar cada uma das transformações implementadas e do seu funcionamento.

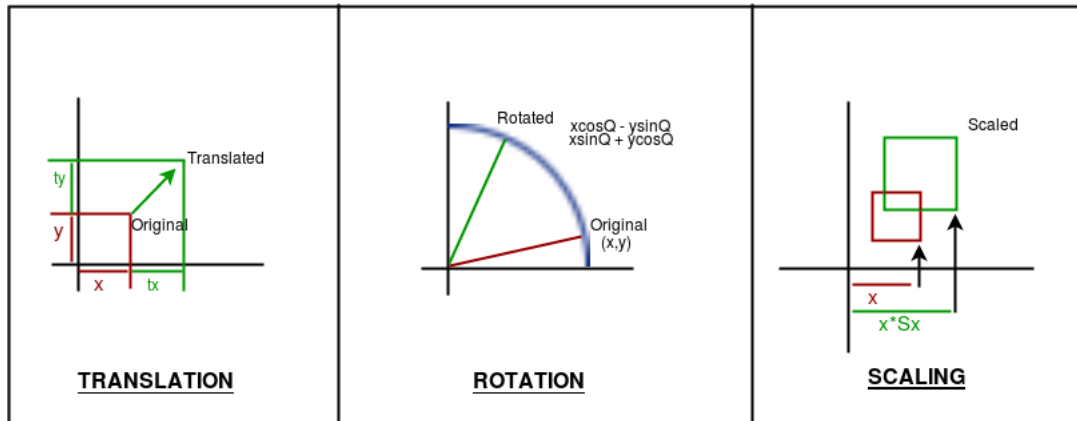


Figura 1: Transformações aplicáveis às figuras

Para a representação do Sistema Solar foram desenvolvidas 2 alternativas, com o objetivo de compreender e obter uma noção da escala das figuras, bem como da sua representação. Deste modo, foi desenvolvida uma versão em que os planetas se encontram alinhados com o eixo dos x do referencial, e uma representação em que se pretende simular as órbitas dos planetas.

Em seguida apresentam-se as imagens de ambas as representações:

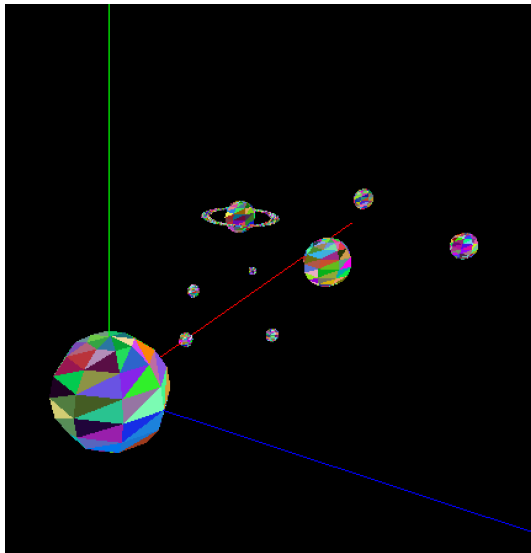


Figura 2: Sistema Solar

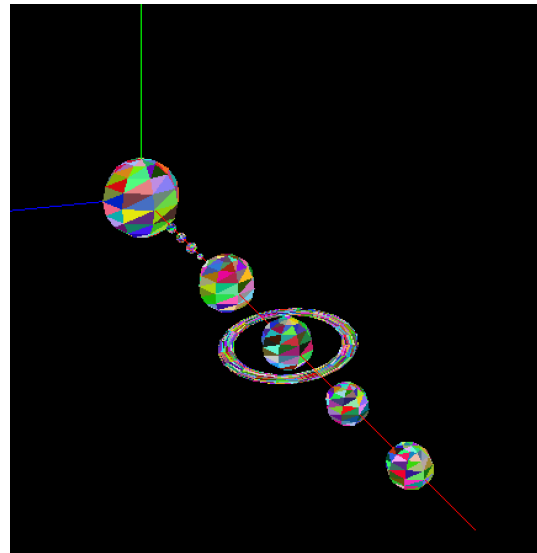


Figura 3: Sistema Solar Alinhado

2.4 Torus

De modo a desenhar o anel de Saturno (verificar se é so isto que desenhamos), acrescentamos uma figura ao *generator*, o *Torus*. Para o desenho desta figura é necessário passar 4 argumentos:

- *innerRadius*: é o raio interior do *Torus* ('r' da Figura 4)
- *outerRadius*: é o raio exterior do *Torus* ('R' da Figura 4)
- *slices*
- *stacks*

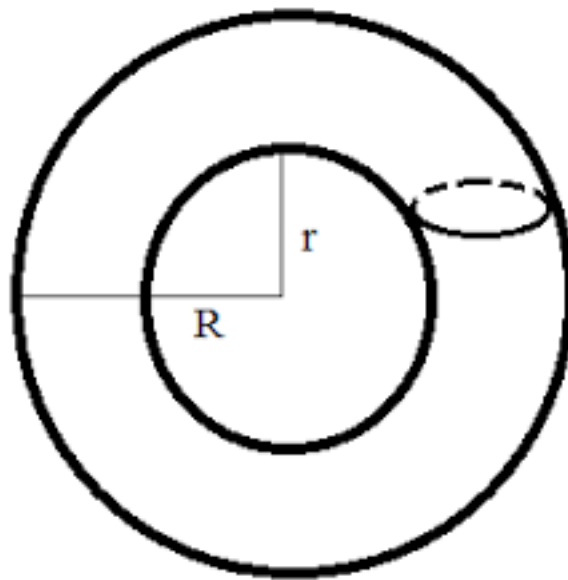


Figura 4: Raio interior e exterior do Torus

Com este *innerRadius* e *outerRadius*, calculamos o *radius* (distância entre o centro do tubo até ao centro do *torus*, 'R' na Figura 5) e o *ringRadius* (raio que define a espessura do torus, 'r' na Figura 5) da seguinte forma:

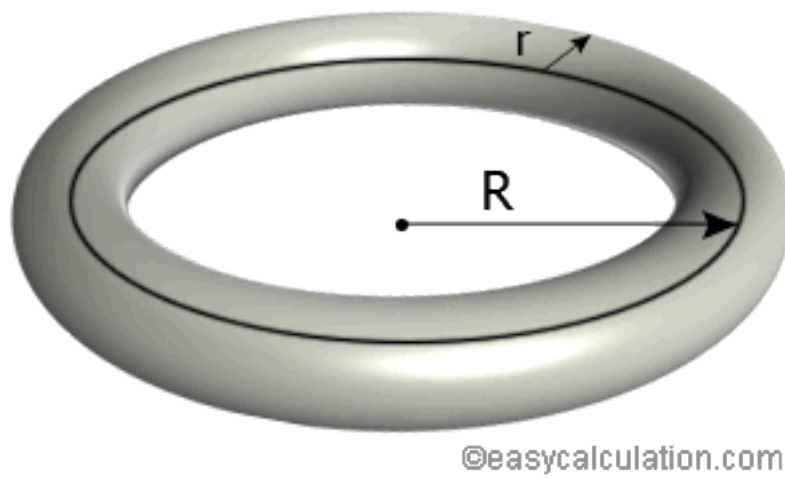


Figura 5: Radius e RingRadius do Torus

```
radius = (innerRadius + outerRadius) / 2.0f;  
ringRadius = (outerRadius - innerRadius) / 2.0f;
```

Após termos calculado estes dois raios, iteramos sobre as *stacks* e *silces*, geramos os pontos e obtemos a seguinte figura:

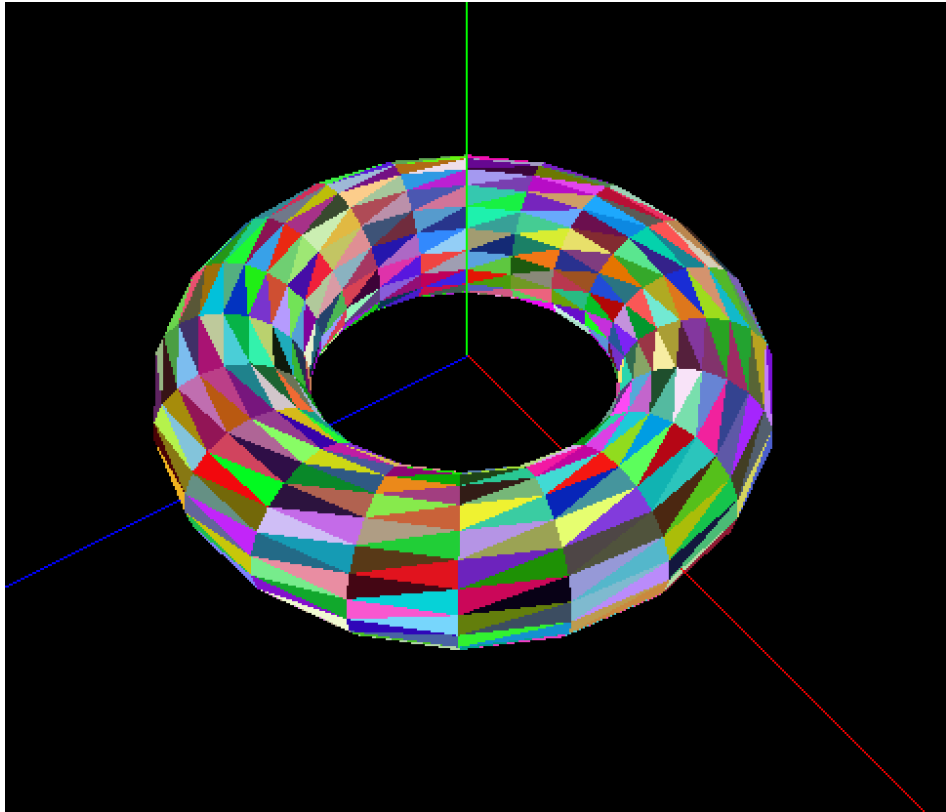


Figura 6: Torus desenhada com $innerRadius = 1$, $outerRadius = 2$, $slices = 20$ e $stacks = 20$

3 Conclusão

A elaboração desta fase do trabalho prático permitiu a representação de um modelo inicial do sistema solar.

Nesta parte, para desenhar os diferentes planetas, utilizamos a esfera, a qual já tinha sido implementada na fase anterior. Além disso, foi criado um polígono extra, o *Torus*, para o desenho de anéis de planetas.

Com o desenvolvimento desta fase, foi possível consolidar conhecimentos relacionados com a manipulação de rotações, translações e escalas para permitir que o modelo do sistema solar ficasse mais próximo da realidade.

Para além destes conhecimentos, a presente fase do trabalho permitiu expandir ainda mais, o conhecimento na linguagem C++ e na manipulação de ficheiros XML.