

Universidade do Minho  
Licenciatura em Engenharia Informática

## Computação Gráfica

Fase 2 - Grupo 20

André Silva - A87958

Armando Silva - A87949

Joana Oliveira - A87956

João Nunes - A87972

Abril 2022



# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Estrutura do Ficheiro XML</b>	<b>4</b>
<b>3</b>	<b>Engine/Motor</b>	<b>5</b>
3.1	Estrutura de dados implementada . . . . .	5
3.2	Leitura do Ficheiro XML . . . . .	5
3.3	Desenho das Figuras . . . . .	6
<b>4</b>	<b>Extras</b>	<b>7</b>
4.1	Escala . . . . .	7
<b>5</b>	<b>Sistema Solar</b>	<b>8</b>
5.1	Criação do ficheiro XML . . . . .	8
<b>6</b>	<b>Conclusão</b>	<b>9</b>
<b>A</b>	<b>Ficheiro XML</b>	<b>10</b>

## Lista de Figuras

1	Excerto de um ficheiro XML . . . . .	4
2	Variáveis da classe <i>Group</i> . . . . .	5
3	Sistema Solar . . . . .	8

# 1 Introdução

Nesta segunda fase do projeto, é proposto o desenvolvimento de uma representação do sistema solar num cenário gráfico 3D. Esta representação tem de ser realizada num ficheiro XML, que utiliza a primitiva esfera desenvolvida na fase anterior, que depois é interpretado pelo motor (*engine*).

Para a realização desta fase, foi necessária a atualização do motor, criado na fase anterior, para conseguir interpretar e armazenar corretamente os novos elementos dos ficheiros XML. Foi preciso, também, realizar uma melhoria na função que desenha as figuras armazenadas nas estruturas.

De forma a completar esta fase e o projeto em si, o grupo decidiu implementar, para além das funcionalidades já implementadas na fase anterior, a opção de escalar as figuras durante a sua visualização, que permite aproximar e desaproximar a câmara da figura. Todas estas adições e atualizações vão ser explicadas ao longo deste relatório.

## 2 Estrutura do Ficheiro XML

Nesta segunda fase, é necessária a criação de um ficheiro XML mais complexo do que o da fase anterior. Neste, para além das definições da câmara e os ficheiros .3d necessários (e realizados na fase anterior), foram ainda adicionadas transformações a fazer aos sólidos, como cor, rotação, translação e escala. De realçar também que podem agora existir agrupamentos de vários sólidos no XML, através dos *groups* encadeados. Desta forma as transformações aplicadas ao *groups* principal aplicam-se aos seus subgrupos.

Na imagem abaixo é possível observar a representação das diferentes transformações no ficheiro XML e também a hierarquia de *groups* pois a Lua é um subgrupo da Terra.

```
<group>
  <!--TERRA-->
  <transform>
    <color r="0.0" g="0.59" b="0.86"/>
    <rotate angle="233" x="0" y="1" z="0"/>
    <translate x="4.5" y="0" z="0"/>
    <scale x="0.3" y="0.3" z="0.3"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
  <group>
    <!--LUA-->
    <transform>
      <color r="1.0" g="1.0" b="1.0"/>
      <translate x="0" y="1" z="2"/>
      <scale x="0.25" y="0.25" z="0.25"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
</group>
```

Figura 1. Excerto de um ficheiro XML

### 3 Engine/Motor

Nesta fase do projeto é nos pedido uma atualização do engine/motor de forma a permitir fazer transformações geométricas. Para isso, é necessário alterar a forma de como o ficheiro XML é lido e também criar uma nova estrutura de dados que guarde informações do ficheiro. Esta estrutura é denominada *Group* e é povoada na função **parseGroup** que posteriormente é utilizada para desenhar com a função **drawFigures**.

#### 3.1 Estrutura de dados implementada

Como já foi anteriormente dito, o ficheiro XML contém novas informações, i.e, as transformações, e por isso, é necessário criar uma nova estrutura que guarde as transformações geométricas para as respetivas figuras.

Esta corresponde à nova classe *Group* que é constituída por várias variáveis que guardam as inúmeras transformações, o nome da figuras que são transformadas e também contém uma variável da classe *group* de forma a respeitar possíveis transformações hierárquicas.

```
private:
    vector<string> file;
    vector<float> color;
    vector<float> rotate;
    vector<float> translate;
    vector<float> scale;
    vector<Group> subgroups;
```

Figura 2. Variáveis da classe *Group*

#### 3.2 Leitura do Ficheiro XML

A leitura do ficheiro XML teve que ser alterada devido às mudanças da estrutura do mesmo e também à adição da nova estrutura de dados.

Relativamente à função **parseXML** apenas foi retirada a leitura dos *models* em relação à Fase 1. Esta agora trata de colocar os parâmetros certos para a câmara e também de iterar *group* a *group*, chamando a cada iteração a nova função criada. Esta nova função, denominada por **parseGroup**, recolhe toda a informação relativamente a transformações, *models* e subgrupos e armazena-as numa variável da classe *Group*, que posteriormente é guardada numa variável global, que contém todos os *groups* do ficheiro XML.

### 3.3 Desenho das Figuras

Para desenhar as figuras é utilizada a função existente na **renderScene**, **drawFigures**, que é chamada durante a iteração da variável global que contém todos os grupos presentes no ficheiro XML.

Esta função aplica todas as transformações geométricas existentes no *group* recebido. De seguida, a cada ficheiro .3d encontrado, chama a função **parse3D**, já existente na Fase 1, mas que no entanto foi alterada de forma a retornar todos os pontos necessários para desenhar a figura em vez de os armazenar numa variável global. Posteriormente estes pontos são iterados e desenhados com outra função, já existente também na Fase 1, chamada **drawTriangle**. Por fim, a função **drawFigures** é chamada recursivamente caso existam subgrupos, de forma a respeitar a hierarquia das figuras.

## 4 Extras

Para completarmos esta fase do trabalho, o grupo decidiu implementar a funcionalidade de aproximação e desaproximação da câmara à figura, com o nome de escala.

Infelizmente, devido a alguns problemas durante a atualização do funcionamento de partes fundamentais do motor, algumas funcionalidades implementadas na primeira fase tiveram de ser desativadas. Respetivamente, o avanço e recuo entre figuras e a alteração do modo de desenho durante a visualização dos cenários. Estas que serão novamente exploradas nas próximas fases com a intenção de voltar a ativá-las.

### 4.1 Escala

De modo a ser possível aproximar e desaproximar a câmara das figuras, foi implementada uma variável global que é incrementada quando é pressionada a tecla *UP\_ARROW* e decrementada quando pressionada a tecla *DOWN\_ARROW*. Isto é possível usando a função *glScalef*.



## 5 Sistema Solar

Para atingirmos o objetivo desta segunda fase, desenvolvemos de raiz um ficheiro XML, aceite pelo nosso programa, que representa um cenário estático do sistema solar. De modo a permitir que a sua representação fosse o mais realista possível, mas ainda assim perceptível, optamos por fazer um balanço entre as dimensões reais e uma aproximação dessas dimensões.

O ficheiro contém, então, todos os planetas ordenados pela sua distância relativa ao sol. Além disso foram também incluídas as luas de todos os planetas, com exceção de Mercúrio e Vénus, sendo que apenas Terra e Marte contêm o número real. O anel de Saturno foi também desenvolvido. Na imagem seguinte é possível visualizar o conteúdo do ficheiro XML, `solarSystem.xml` presente nos anexos, já desenhado.

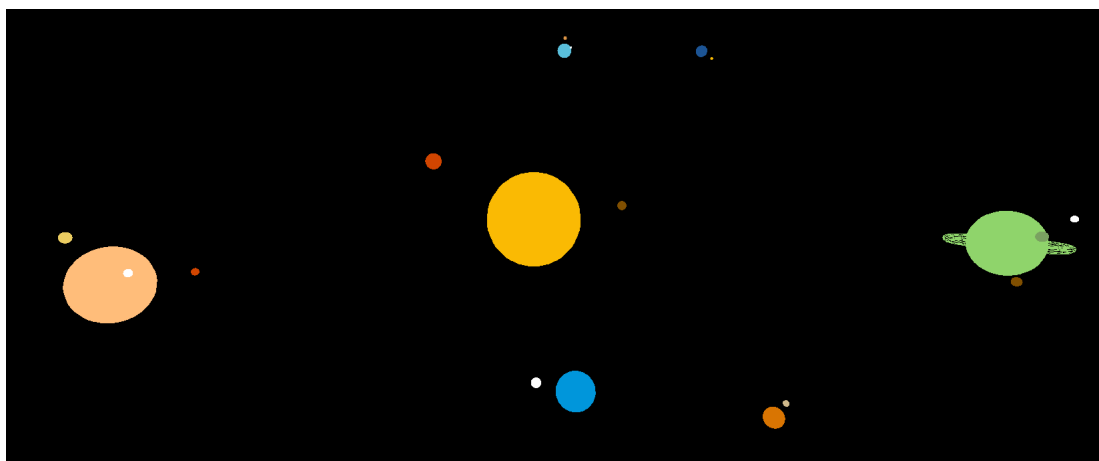


Figura 3. Sistema Solar

### 5.1 Criação do ficheiro XML

A criação do ficheiro XML foi feita com base na estrutura que nos foi apresentada em que cada *group* deverá conter as respetivas transformações (*translate*, *rotate*, *scale*, *color*), que serão também depois aplicadas aos restantes *group* lá contidos, bem como pelo menos um ficheiro `.3d` associado ao *models*.

Relativamente às transformações existentes, o *translate* teve em conta as distâncias entre planetas, o *rotate* permitiu que todos os planetas se encontrassem em diferentes fases de órbita, o *scale* foi utilizado de modo a aproximar o tamanho de cada planeta ao tamanho real em relação ao sol e, por fim, a *color* permitiu dar aos planetas cores que se aproximam das suas verdadeiras, utilizando o sistema RGB.

## 6 Conclusão

Dado por concluída a segunda fase deste projeto, consideramos importante realçar todos os pontos positivos e negativos, e ainda, efetuar uma análise crítica final do trabalho realizado.

O cenário estático do sistema solar desenvolvimento está bem construído e visualmente agradável, com um ótimo balanço entre a realidade e uma visão perceptível dos planetas. A implementação da funcionalidade de aproximar e desaproximar do cenário, possibilita ao utilizador vários pontos únicos de observação do sistema solar.

Apesar do nosso sistema solar estar bastante completo, alguns detalhes poderiam ser implementados, como o Cinturão de Kuiper. A câmara atual implementada na aplicação não fornece, ao utilizador, liberdade total para a observação do cenário. Ponto este que pode ser melhorado com a implementação de uma câmara em que o utilizador pode movimentar-se pelo cenário com liberdade total.

Para concluir, dado que o trabalho realizado cumpre com todos os requisitos propostos depois de todas as dificuldades terem sido ultrapassadas, o grupo considera que este está positivo.

## A Ficheiro XML

```
<world>
  <camera>
    <position x="10" y="10" z="10"/>
    <lookAt x="0" y="0" z="0"/>
    <up x="0" y="1" z="0"/>
    <projection fov="60" near="1" far="1000"/>
  </camera>
  <group>
    <!--SOL-->
    <transform>
      <color r="0.98" g="0.73" b="0.01"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
  <group>
    <!--MERCURIO-->
    <transform>
      <color r="0.5" g="0.31" b="0.0"/>
      <rotate angle="-30.0" x="0.0" y="1.0" z="0.0"/>
      <translate x="2" y="0" z="0"/>
      <scale x="0.1" y="0.1" z="0.1"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
  <group>
    <!--VENUS-->
    <transform>
      <color r="0.82" g="0.27" b="0.0"/>
      <rotate angle="90" x="0" y="1" z="0"/>
      <translate x="3.5" y="0" z="0"/>
      <scale x="0.2" y="0.2" z="0.2"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
  <group>
    <!--TERRA-->
    <transform>
      <color r="0.0" g="0.59" b="0.86"/>
      <rotate angle="233" x="0" y="1" z="0"/>
      <translate x="4.5" y="0" z="0"/>
      <scale x="0.3" y="0.3" z="0.3"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
</world>
```

```

<group>
  <!--LUA-->
  <transform>
    <color r="1.0" g="1.0" b="1.0"/>
    <translate x="0" y="1" z="2"/>
    <scale x="0.25" y="0.25" z="0.25"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
</group>
</group>
<group>
  <!--MARTE-->
  <transform>
    <color r="0.85" g="0.46" b="0.007"/>
    <rotate angle="260" x="0" y="1" z="0"/>
    <translate x="6" y="0" z="0"/>
    <scale x="0.15" y="0.15" z="0.15"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
  <group>
    <!--LUA1-->
    <transform>
      <color r="1.0" g="1.0" b="1.0"/>
      <translate x="0" y="-1" z="-1.25"/>
      <scale x="0.25" y="0.25" z="0.25"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
  <group>
    <!--LUA2-->
    <transform>
      <color r="0.82" g="0.73" b="0.55"/>
      <translate x="0" y="1" z="-1.5"/>
      <scale x="0.3" y="0.3" z="0.3"/>
    </transform>
    <models>
      <model file="sphere.3d"/>
    </models>
  </group>
</group>
<group>
  <!--JUPITER-->
  <transform>
    <color r="1.0" g="0.74" b="0.48"/>
    <rotate angle="150" x="0" y="1" z="0"/>
    <translate x="8" y="0" z="0"/>
    <scale x="0.7" y="0.7" z="0.7"/>
  </transform>

```

```

<models>
  <model file="sphere.3d"/>
</models>
<group>
  <!--LUA1-->
  <transform>
    <color r="0.82" g="0.27" b="0.0"/>
    <translate x="-2" y="0" z="0"/>
    <scale x="0.1" y="0.1" z="0.1"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
</group>
<group>
  <!--LUA2-->
  <transform>
    <color r="1.0" g="1.0" b="1.0"/>
    <translate x="-1" y="1" z="-1"/>
    <scale x="0.1" y="0.1" z="0.1"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
</group>
<group>
  <!--LUA3-->
  <transform>
    <color r="0.92" g="0.8" b="0.38"/>
    <translate x="1" y="1" z="1"/>
    <scale x="0.15" y="0.15" z="0.15"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
</group>
<group>
  <!--LUA4-->
  <transform>
    <color r="0.55" g="0.42" b="0.26"/>
    <translate x="1" y="-1" z="1"/>
    <scale x="0.15" y="0.15" z="0.15"/>
  </transform>
  <models>
    <model file="sphere.3d"/>
  </models>
</group>
</group>
<group>
  <!--SATURNO-->
  <transform>
    <color r="0.56" g="0.83" b="0.42"/>
    <rotate angle="-50" x="0" y="1" z="0"/>
    <translate x="9.5" y="0" z="0"/>

```

```

        <scale x="0.65" y="0.65" z="0.65"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--ANEL-->
        <transform>
            <rotate angle="-30" x="1" y="0.0" z="0.0"/>
        </transform>
        <models>
            <model file="torus.3d"/>
        </models>
    </group>
    <group>
        <!--LUA1-->
        <transform>
            <color r="0.48" g="0.63" b="0.38"/>
            <translate x="0" y="1" z="1"/>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA2-->
        <transform>
            <color r="1.0" g="1.0" b="1.0"/>
            <translate x="1.5" y="1" z="0"/>
            <scale x="0.1" y="0.1" z="0.1"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA3-->
        <transform>
            <color r="0.5" g="0.31" b="0.0"/>
            <translate x="1" y="-1.5" z="0"/>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
    <!--URANO-->
    <transform>
        <color r="0.35" g="0.75" b="0.86"/>
        <rotate angle="39" x="0" y="1" z="0"/>
        <translate x="10.5" y="0" z="0"/>

```

```

        <scale x="0.25" y="0.25" z="0.25"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA1-->
        <transform>
            <color r="1.0" g="1.0" b="1.0"/>
            <translate x="-1.5" y="1" z="1"/>
            <scale x="0.15" y="0.15" z="0.15"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
    <group>
        <!--LUA2-->
        <transform>
            <color r="0.87" g="0.59" b="0.28"/>
            <translate x="1" y="1.5" z="0"/>
            <scale x="0.25" y="0.25" z="0.25"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
<group>
    <!--NEPTUNO-->
    <transform>
        <color r="0.11" g="0.33" b="0.58"/>
        <rotate angle="15" x="0" y="1" z="0"/>
        <translate x="12" y="0" z="0"/>
        <scale x="0.2" y="0.2" z="0.2"/>
    </transform>
    <models>
        <model file="sphere.3d"/>
    </models>
    <group>
        <!--LUA-->
        <transform>
            <color r="0.98" g="0.73" b="0.01"/>
            <translate x="0" y="-1" z="2"/>
            <scale x="0.25" y="0.25" z="0.25"/>
        </transform>
        <models>
            <model file="sphere.3d"/>
        </models>
    </group>
</group>
</world>

```