



**Universidade do Minho**  
Escola de Engenharia

UNIVERSIDADE DO MINHO  
DEPARTAMENTO DE INFORMÁTICA  
COMPUTAÇÃO GRÁFICA

Trabalho Prático Nº 2

Diogo Araujo (A89517)  
António Silva (A89558)  
Pedro Novais (A78211)  
Gonçalo Soares (A84441)

4 de abril de 2021



Diogo Araújo



António Silva



Gonçalo Soares



Pedro Novais

## Conteúdo

<b>1</b>	<b>Fase 2 – Transformações geométricas</b>	<b>3</b>
1.1	Introdução . . . . .	3
1.2	Ficheiro XML . . . . .	3
1.3	Estruturas de dados utilizadas . . . . .	3
1.4	Algoritmos utilizados . . . . .	5
1.5	Resultado obtido . . . . .	5
1.6	Conclusão . . . . .	6

## Fase 2 – Transformações geométricas

### 1.1 Introdução

Esta fase do projeto consiste em criar um modelo usando transformações geométricas. O seu desenvolvimento tem por base a leitura de um ficheiro no formato XML para a criação de um modelo estático do sistema solar, incluindo o sol, planetas e respetivas luas.

O ficheiro XML deverá ser criado tendo por base um conjunto de regras definidas no enunciado e deve ainda guardar informação relativa às cores e transformações a aplicar sobre as formas geométricas fornecidas pelo Generator (desenvolvido na fase anterior).

O presente documento visa apresentar as melhores soluções que o grupo encontrou para a evolução do modelo proposto.

### 1.2 Ficheiro XML

O ficheiro XML apresentado pelo grupo visa a criação do sistema solar requerido constituído pelo sol, planetas e apenas a lua que orbita o nosso Planeta. Nele é guardado a informação sobre as cores e as transformações a aplicar para a melhor simulação do sistema pedido. O grupo tentou ainda que o sistema ficasse o mais idêntico possível com o sistema solar tal como o conhecemos, atribuindo o tamanho correto e a cor aos respetivos planetas (sol e lua também).

Exemplo válido de XML que utilizado para representar o nosso sol:

```
<group>
  <translate X="-100" Y="1" Z="0"/>
  <scale X="5.5" Y="5.5" Z="5.5"/>
  <!-- SUN -->
  <models>
    <model file="sphere.3d" diffR="240" diffG="107" diffB="0"/>
  </models>
</group>
```

### 1.3 Estruturas de dados utilizadas

Para o desenvolvimento da aplicação idealizada, o grupo recorreu á definição de um conjunto de estruturas de dados que achamos adequadas para guardar toda a informação que pretendíamos e achamos relevante.

A estratégia utilizada foi: guardar um identificador (um inteiro) nas transformações (translações, rotações e escalas) e, para cada figura encontrada dentro do group associado (no ficheiro XML), guardar esse mesmo identificador. Assim, poderemos identificar inequivocamente todas as transformações associadas a uma certa figura. Importante será ainda dizer que, *groups* "filho" herdam as transformação aplicadas aos *groups* "pai". As estruturas desenvolvidas foram então:

- vertice - Guarda três float e um inteiro. Estrutura responsável por guardar a cada tuplo de 3 pontos fornecidos pelo Generator e que constituem um triângulo. O inteiro servirá para guardar o número total de pontos da figura a desenhar. Esta estrutura é herdada da primeira fase do trabalho.
- translate - Guarda três float e um inteiro. Estrutura responsável por guardar o vetor de translação lido a partir do ficheiro XML e o identificador da figura á qual se irá aplicar a translação.
- rotate - Guarda quatro float e um inteiro. Estrutura responsável por guardar o ângulo e a que eixo é aplicada a rotação. Tal como na estrutura translate, há um identificador que associa a rotação á figura.
- scale - Guarda três float e um inteiro. Estrutura responsável por guardar o coordenada X, coordenada Y e coordenada Z associada a uma escala. Tal como na estrutura translate e rotate, há um identificador que associa a escala á figura.
- figure. Guarda uma string, um conjunto de elementos do tipo vertice, um conjunto de inteiros e um array de float. Estrutura responsável por guardar o nome da figura a desenhar, o conjunto de pontos necessários para o desenho da figura, o conjunto dos identificadores (que associam a figura em questão a todas as transformações que lhe serão aplicadas) e a cor da figura.
- figures - Estrutura responsável por guardar todas as figuras e transformações associadas ao projeto e ao desenvolvimento do modelo estático.

## 1.4 Algoritmos utilizados

Os algoritmos que suportam a aplicação vão de encontro às estruturas de dados e á forma como guardamos a informação que pretendemos. Nesta fase do projeto existem então duas etapas: a leitura do ficheiro XML (e respetivo armazenamento da informação retirada a partir dele nas estruturas de dados) e o algoritmo de percorrer as estruturas de dados e ir desenhando o modelo pretendido. Importante salientar a utilização dos métodos **glPushMatrix()** e **glPopMatrix()**, já pré-definidos, como forma de guardar o sistema de coordenadas antes de desenhar qualquer figura. Quanto às transformações geométricas, recorreremos às funções pré-definidas **glTranslatef()**, **glRotatef()** e **glScalef()**

Ciclo for que percorre as estrutura de dados:

```
for (int i = 0; i < globalFigs->figuras.size(); i++)
{
    glPushMatrix();
    applyTransf(i);
    draw(globalFigs->figuras[i].vertices, globalFigs->figuras[i].rgb);
    glPopMatrix();
}
```

## 1.5 Resultado obtido

O resultado do projeto desenvolvido foi o seguinte:

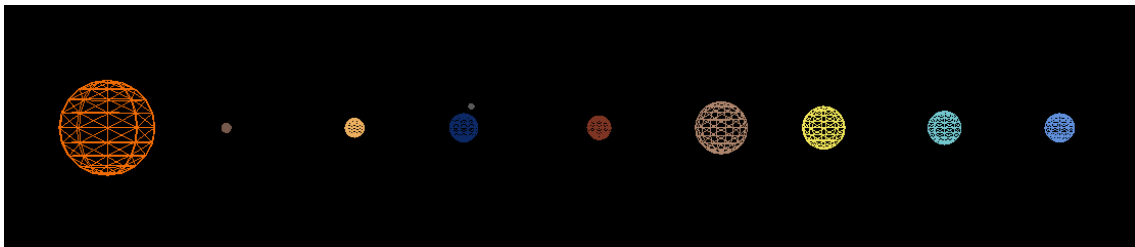


Figura 1.1: Exemplo de Sistema Solar gerado

## 1.6 Conclusão

Nesta fase do projeto, houve duas etapas essenciais que consistiam na elaboração do ficheiro XML e de seguida a sua leitura e elaboração das figuras.

A elaboração do XML está estritamente ligada com a construção de estruturas para guardarmos apropriadamente a informação lida, sendo que esta foi uma das maiores dificuldades a superar pois obrigou a uma maior consciencialização da memória.

De seguida, com os dados na estruturas recorremos ao desenho das figuras sendo que recorremos aos algoritmos utilizados anteriormente mas tendo sempre em consideração o uso das funções *rotate* e *translate* que nos permitiu colocar as figuras com diferentes perspectivas em diferentes referenciais.

Em suma, este trabalho foi importante para aprofundar e cimentar tudo o que foi dado que consistia em desenho de figuras recorrendo a triângulos, até a elaboração de várias figuras em diferentes planos, recorrendo a estruturas para guardar o conteúdo lido de XML.