

Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Computação Gráfica

Ano Letivo de 2022/2023

Fase 2 – Transformações Geométricas

Grupo 6

A96434 Francisca Quintas Monteiro de Barros
A97588 Joana Isabel Freitas Pereira
A94870 Rafael Picão Ferreira Correia

14 de abril de 2023

Índice

Lista de Figuras	3
1 Introdução	4
2 Engine	5
3 Testes	6
3.0.1 Teste 1	6
3.0.2 Teste 2	7
3.0.3 Teste 3	7
3.0.4 Teste 4	8
4 Sistema Solar	9
5 Conclusão	11

Lista de Figuras

2.1	Exemplo da configuração de um ficheiro XML	5
3.1	Teste 1	6
3.2	Teste 2	7
3.3	Teste 3	7
3.4	Teste 4	8
4.1	Modelo do Sistema Solar	10

1 Introdução

O presente relatório é referente à segunda fase do trabalho prático desenvolvido pelo grupo 06 no âmbito da Unidade Curricular de Computação Gráfica, lecionada no curso de Licenciatura em Engenharia Informática no 2º Semestre do ano letivo 2022/2023.

Este projeto tem como objetivo o desenvolvimento de um mecanismo 3D baseado num cenário gráfico, através de ferramentas como o OpenGL e o C++, também usadas nas aulas práticas.

O projeto está dividido em 4 fases e para cada fase, serão fornecidos arquivos de configuração XML para testes e avaliação.

Nesta segunda fase o grupo apenas alterou a parte da *engine* com o objetivo de suportar transformações geométricas e criou um modelo estático do sistema solar.

2 Engine

Na fase atual, a *engine* foi modificada de modo a estar pronta para receber arquivos que suportam transformações geométricas, conforme mencionado anteriormente. Além disso, possuem também um sistema de hierarquia em que é possível aninhar várias transformações para diferentes modelos.

Na imagem abaixo, podemos observar dois exemplos de transformações, ambas de translação. O primeiro modelo sofre uma translação de 4 unidades no eixo x, enquanto que o segundo modelo (cone.3d) sofre duas translações: uma de 4 unidades no eixo x e outra de 4 unidades no eixo y. Isso ocorre porque o segundo modelo é "filho" do primeiro e está aninhado na sua hierarquia de transformações.

```
<group>
  <transform>
    <translate x="4" y="0" z="0"/>
  </transform>
  <models>
    <model file="sphere.3d" />
  </models>
  <group>
    <transform>
      <translate x="0" y="4" z="0"/>
    </transform>
    <models>
      <model file="cone.3d" />
    </models>
  </group>
</group>
```

Figura 2.1: Exemplo da configuração de um ficheiro XML

Essa capacidade de gerir hierarquias de transformações permite a criação de modelos mais complexos a partir da combinação de modelos mais simples.

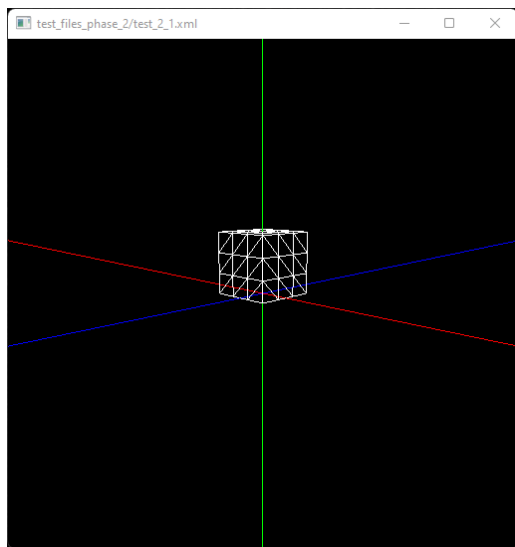
Na parte do *parse* do ficheiro XML, tal como na primeira fase foi utilizada a biblioteca RapidXML, o que nos permitiu armazenar toda a informação do ficheiro numa árvore. Posteriormente, quando se chega à fase de desenhar os modelos, na função *renderScene*, o grupo percorre a árvore e realiza as transformações necessárias para desenhar cada modelo.

De forma a suportar a hierarquia é utilizada a função *glPushMatrix()* no início de cada *group* e *glPopMatrix()* no final. Isto permite redefinir a matriz e garantir que as transformações daquele *group* apenas afetem os seus descendentes e não outros *group*.

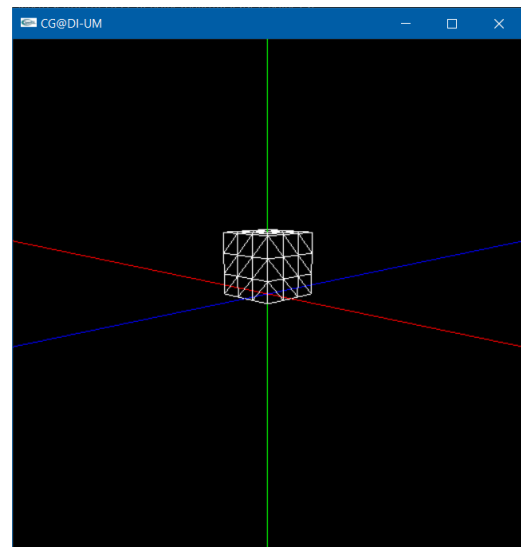
3 Testes

Para testar a *Engine*, o grupo utilizou os ficheiros de teste fornecidos pelo professor e é possível afirmar que todos os modelos renderizados ficaram idênticos às imagens de teste. Isso significa que a *Engine* foi capaz de interpretar corretamente os dados dos ficheiros, realizar as transformações necessárias e desenhar os modelos de forma precisa e fiel às imagens fornecidas.

3.0.1 Teste 1



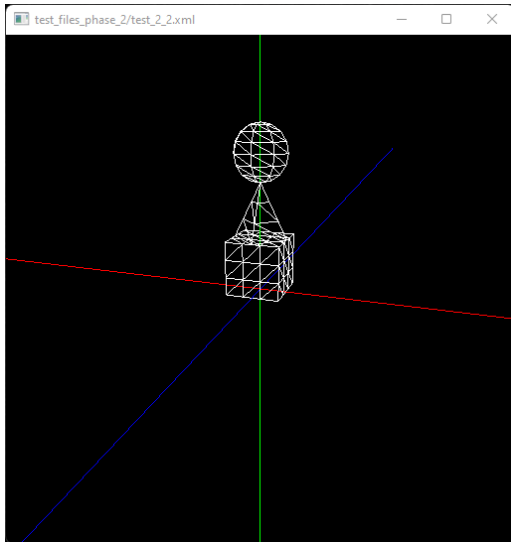
(a) Resultado esperado



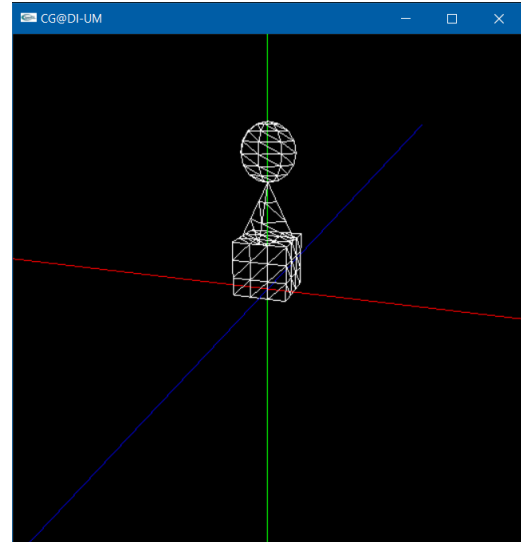
(b) Nosso resultado

Figura 3.1: Teste 1

3.0.2 Teste 2



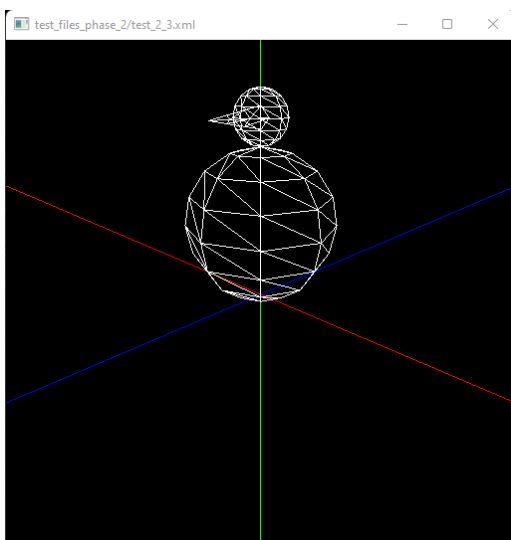
(a) Resultado esperado



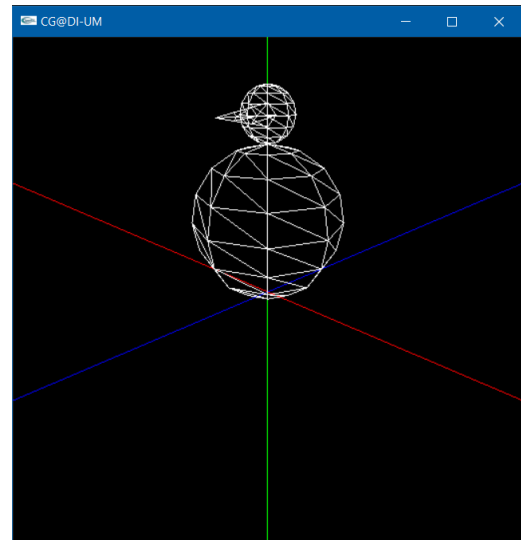
(b) Nosso resultado

Figura 3.2: Teste 2

3.0.3 Teste 3



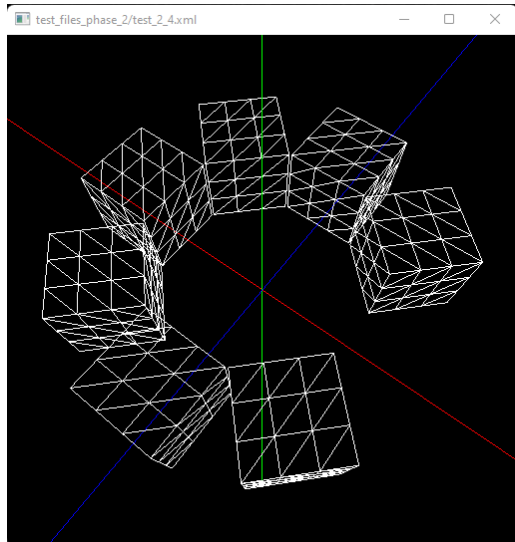
(a) Resultado esperado



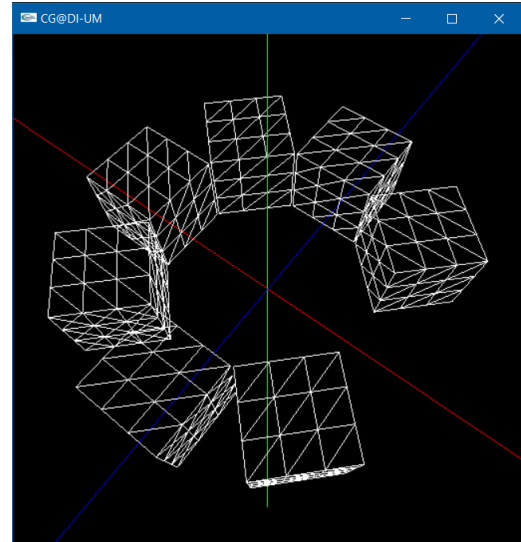
(b) Nosso resultado

Figura 3.3: Teste 3

3.0.4 Teste 4



(a) Resultado esperado



(b) Nosso resultado

Figura 3.4: Teste 4

4 Sistema Solar

Após a implementação das transformações na *Engine*, foi proposto a realização de uma demonstração prática criando um modelo estático do sistema solar, incluindo o sol, os planetas e as luas, definidos numa hierarquia.

Para criar este modelo, foram tomadas duas decisões importantes: Primeiramente, foi decidida a construção dos planetas não à escala real, de forma a ser possível inclui-los todos numa única imagem. Para isso reduziu-se as distâncias entre eles e foram definidos tamanhos arbitrários para cada planeta, lua e para o próprio sol. Em segundo lugar, para incluir todas as luas, foi criado um script em Python que construiu o ficheiro de configuração do sistema solar, de acordo com o número correto de luas para cada planeta. No caso do planeta Saturno foi feita uma tentativa de um "anel" com as luas à sua volta, apesar de isso não acontecer na realidade.

De seguida utilizou-se o *Generator* para criar os modelos de cada planeta, que consistem em esferas de raios diferentes. Os seus valores são os seguintes:

- Sol - 200;
- Mercúrio - 5;
- Vénus - 8;
- Terra - 10; lua - 3.5;
- Marte - 6; luas - 1.5;
- Júpiter - 25; luas - 5;
- Saturno - 20; luas- 2, 3.5 ou 4;
- Urano - 15; luas - 3;

O ficheiro de configuração do sistema solar está localizado na pasta `engine/config/config.xml`, e os modelos estão na pasta `engine/models`. Em semelhança à fase anterior, a *Engine* utiliza o ficheiro `config.xml` da pasta `config` como ficheiro de configuração e qualquer modelo a ser utilizado deve estar localizado na pasta `models`.

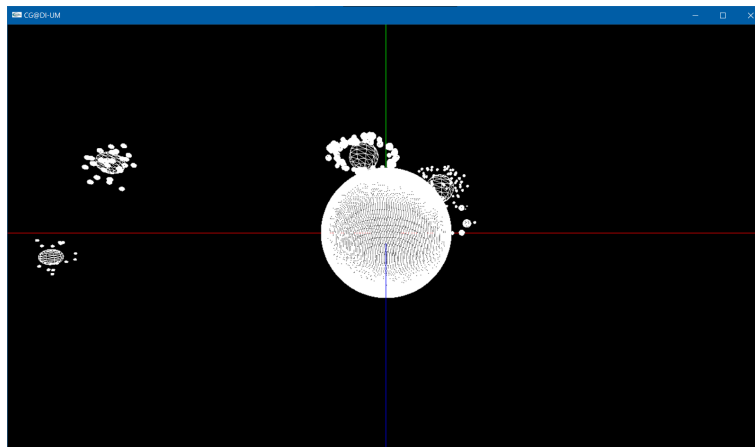


Figura 4.1: Modelo do Sistema Solar

5 Conclusão

Concluimos que a segunda fase deste projeto foi um sucesso, uma vez que fomos capazes de cumprir todos os requisitos propostos no enunciado, melhorando o engine para ter capacidades de transformações geométricas e construí-mos um modelo do sistema solar. Esses programas foram criados de forma extremamente satisfatória, demonstrando a nossa capacidade de aplicar os conhecimentos lecionados até o momento tanto nas aulas teóricas como nas práticas.

Estamos confiantes que o sucesso desta segunda fase, nos ajudará na conclusão das próximas fases e num bom projeto final.