



Universidade do Minho

Mestrado Integrado em Engenharia Informática

Licenciatura em Ciências da Computação

Unidade Curricular de Computação Gráfica

Ano Lectivo de 2018/2019

Relatório da 2ª Fase

André Sá - A76361 (LCC)

Moisés Ramires – A80499 (LCC)

Paulo Barbosa – A81480 (LCC)

Sebastião Freitas – A71074 (MIEI)

Março, 2019



Índice

- 1- Enunciado/ Resumo**
- 2- Definições**
- 3- Exemplos**
- 4- Sistema Solar Estático**

Enunciado /Resumo

Phase 2 – Geometric Transforms

This phase is about creating hierarchical scenes using geometric transforms. A scene is defined as a tree where each node contains a set of geometric transforms (translate, rotate and scale) and optionally a set of models. Each node can also have children nodes.

Example of a configuration XML file with a single group:

```
<scene>
  <group>
    <translate X=5 Y=0 Z=2 />
    <rotate angle=45 axisX=0 axisY=1 axisZ=0 />
    <models>
      <model file="sphere.3d" />
    </models>
  </group>
</scene>
```

Example of a group with a child group:

```
<scene>
  <group>
    <translate X=1 />
    <models>
      <model file="sphere.3d" />
    </models>
    <group>
      <translate Y=1 />
      <models>
        <model file="cone.3d" />
      </models>
    </group>
  </group>
</scene>
```

In the second example the child group will inherit the geometric transforms from the parent.

Geometric transformations can only exist inside a group, and are applied to all models and subgroups.

Note: the order of the geometric transforms is relevant.

The required demo scene for this phase is a static model of the solar system, including the sun, planets and moons defined in a hierarchy.

Nesta fase apenas precisamos de nos preocupar com o nosso **Engine** e o nosso ficheiro **XML**. Pretende-mos implementar as transformações geométricas com o uso de xml.

Definições

Decidimos que os nossos grupos seriam estruturas que contêm 3 vetores: gt para transformações geométricas, modelos (ficheiros .3d que contêm os pontos associados a uma figura) e os respetivos subgrupos. As translações, rotações e escalas são lidas do nosso xml através de funções definidas para o carregamento/load das mesmas. Os nossos subgrupos são lidos de maneira recursiva.

Exemplos

De seguida iremos apresentar diversos ficheiros xml que demonstram uma variedade das transformações e subgrupos a serem aplicados.

1.Cone

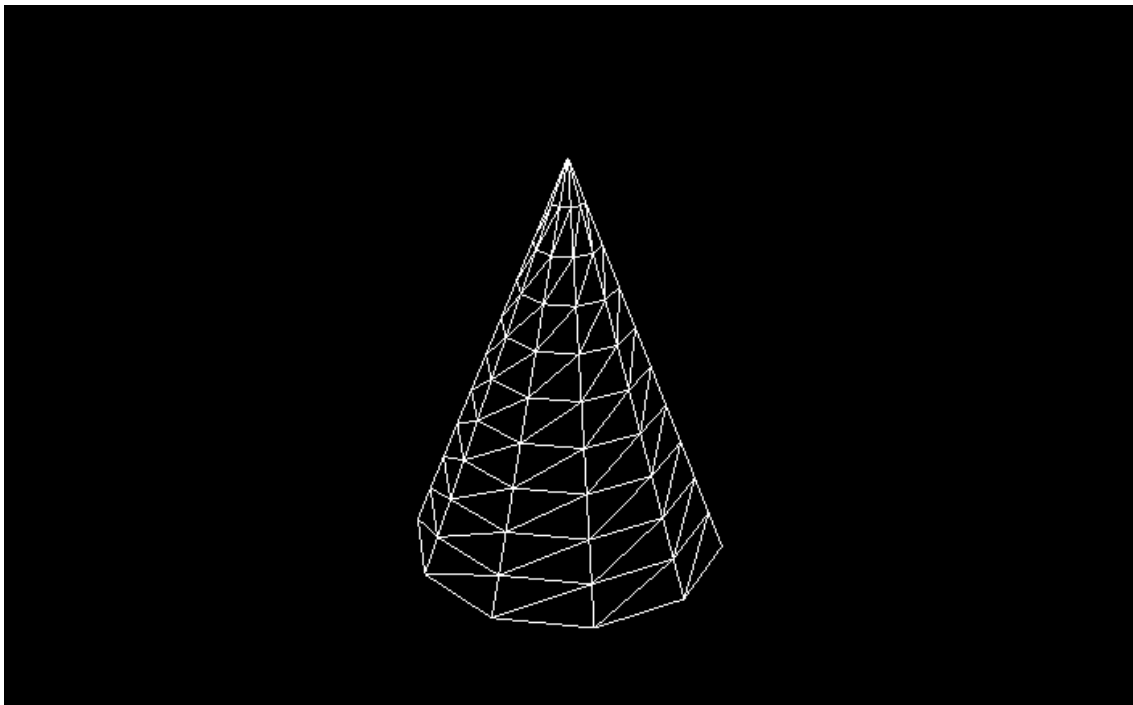


Figura – Cone

2.box -> translação -> box

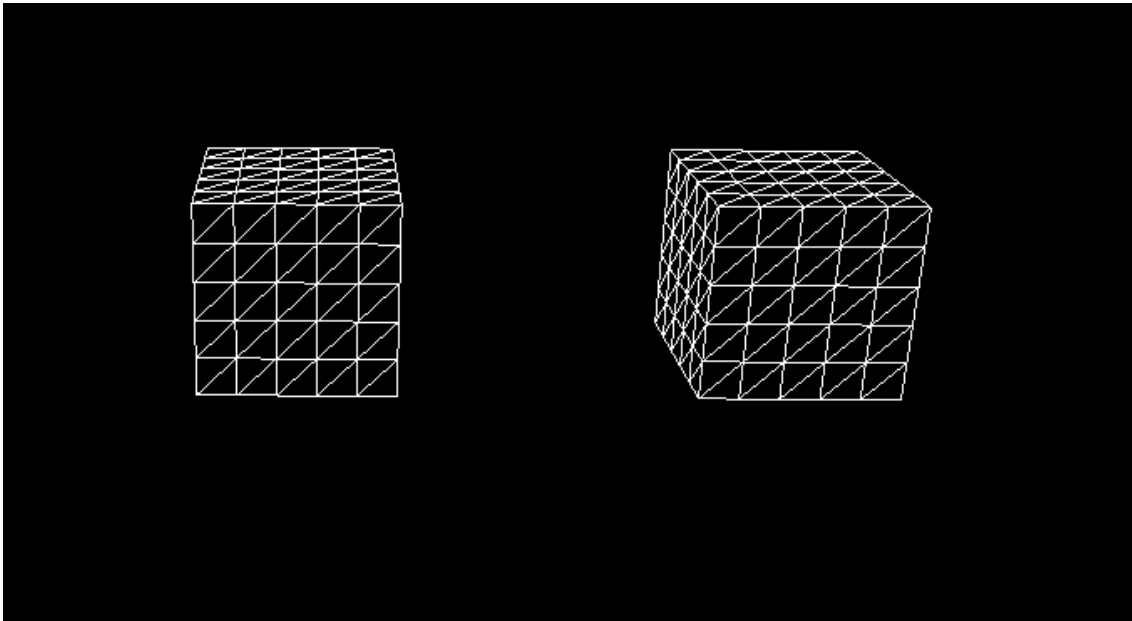


Figura – “Box” e a uma “Box” com mesma dimensões após translação

3.box -> rotação -> box

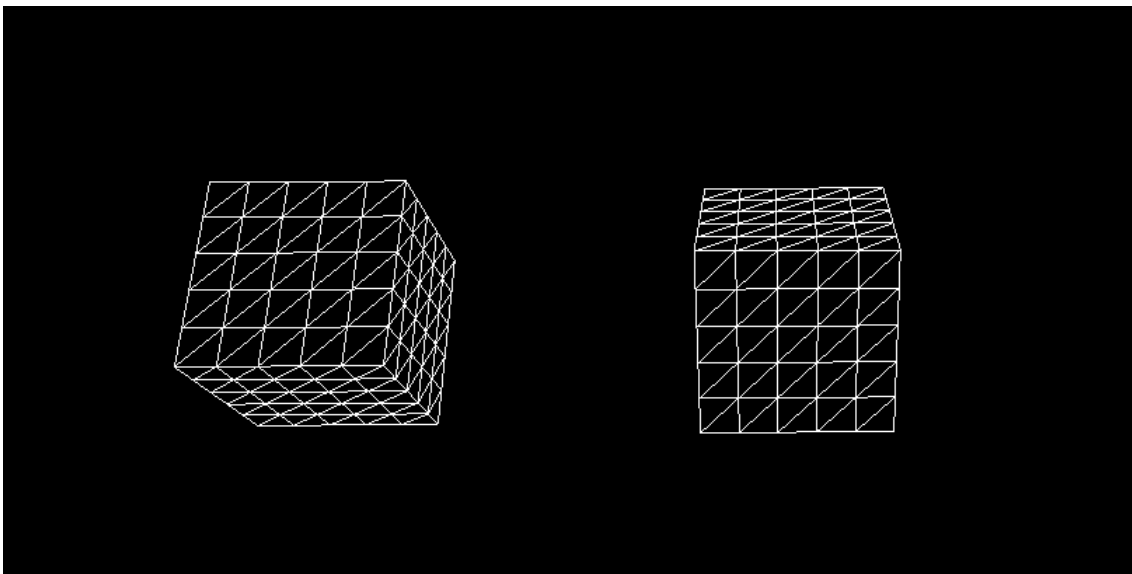


Figura – “Box” e a uma “Box” com mesma dimensões após translação e uma rotação final

4.box -> escala -> box

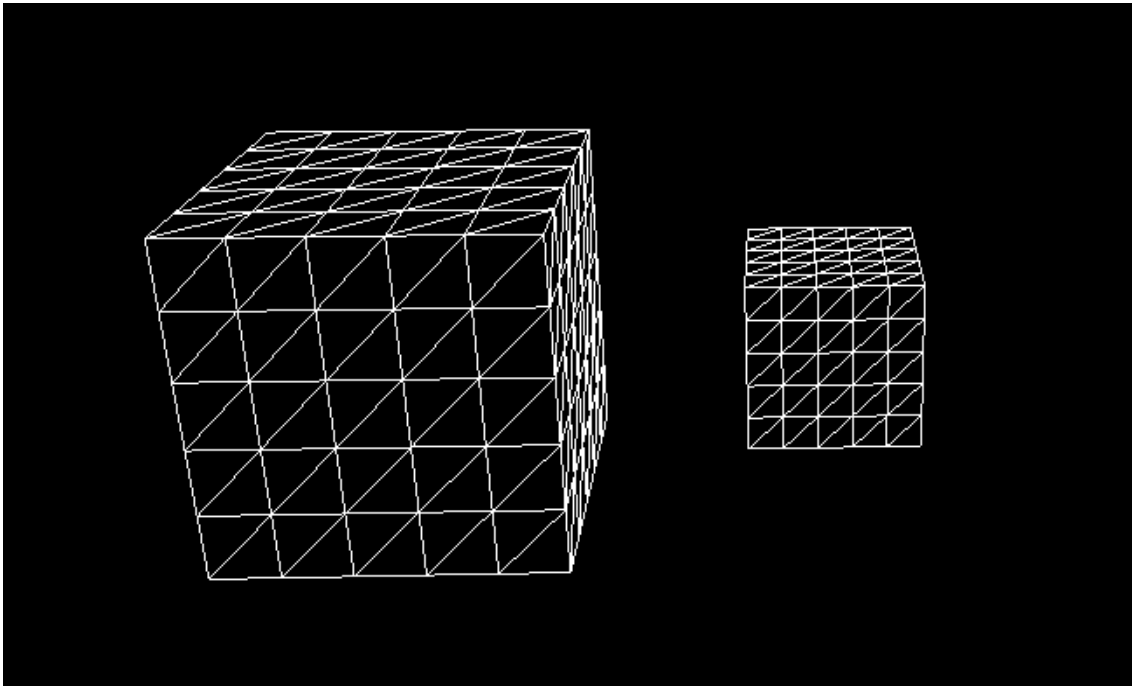


Figura – “Box” e a uma “Box” com mesma dimensões após translação e uma escala final

5.Exemplo complexo

```
<scene>
  <group>
    <models>
      <model file="../../assets/sphere.3d"/>
    </models>
    <group>
      <translate X="5" Y="0" Z="0"/>
      <models>
        <model file="../../assets/cone.3d"/>
      </models>
      <group>
        <translate X="0" Y="10" Z="0"/>
        <rotate ANGLE="45" X="1" Y="0" Z="0"/>
        <models>
          <model file="../../assets/cylinder.3d"/>
        </models>
        <group>
          <translate X="10" Y="0" Z="0"/>
          <scale X="2" Y="2" Z="2"/>
          <models>
            <model file="../../assets/box.3d"/>
          </models>
        </group>
      </group>
    </group>
  </group>
</scene>
```

Figura – Ficheiro xml

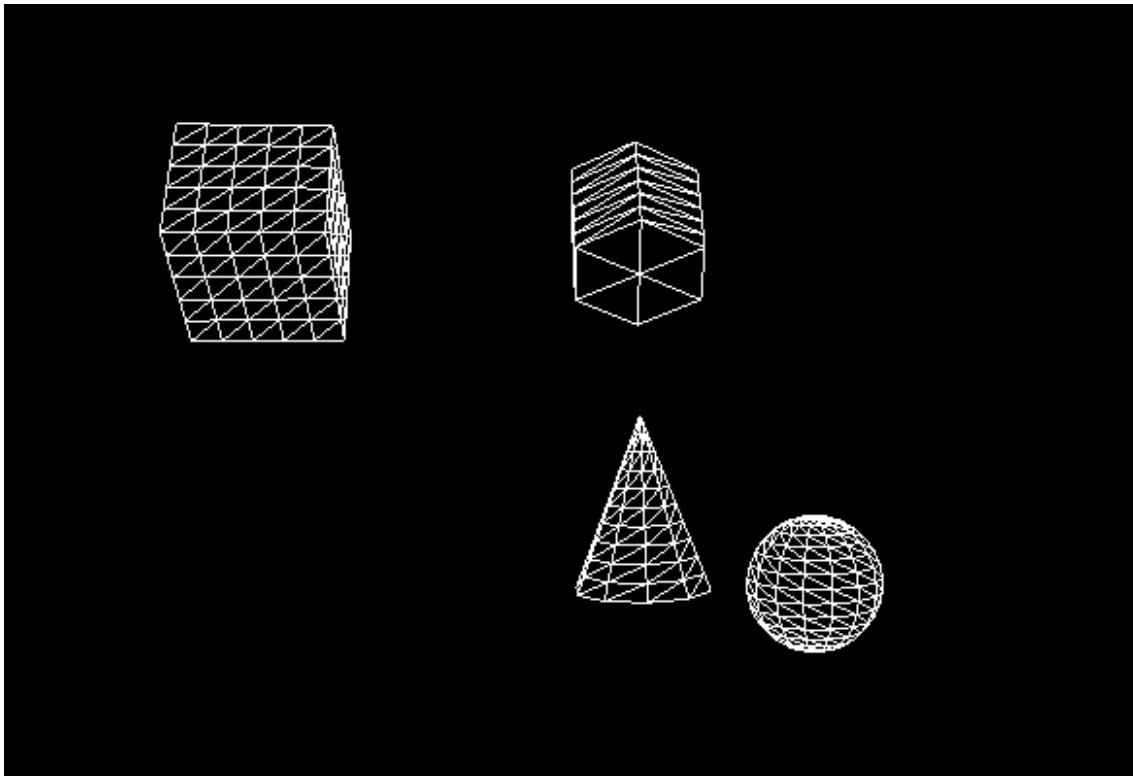


Figura – Resultado do xml interpretado pelo engine

Sistema Solar Estático

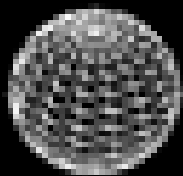


Figura – Sistema Solar Estático