

Trabalho Prático CG - 2022/2023

Fase 1 - Primitivas Gráficas

Licenciatura em Ciências da Computação

Universidade do Minho

Grupo 11

Gonçalo Silva
(A95696)

Nelson Almeida
(A95652)

Nuno Costa
(A97610)

9 de Março de 2023

Conteúdo

1	Introdução	2
1.1	Um belo Projeto	2
2	Engine	6
2.1	Leitura do ficheiro de configuração	6
2.2	Estruturação dos dados	6
2.3	Desenho dos modelos	9
2.4	Execução do Engine	9
3	Concepção/desenho da Resolução	10
3.1	Estruturas de Dados	10
3.2	Algoritmos	10
4	Testes realizados e Resultados	11
4.1	Teste nº1	11
4.2	Teste nº2	12
4.3	Teste nº3	13
4.4	Teste nº4	14
4.5	Teste nº5	15
5	Conclusão	17
A	Código do Programa	18

Capítulo 1

Introdução

Nesta primeira frase do trabalho prático da UC de Computação Gráfica, foi-nos proposto o desenvolvimento de dois programas, sendo eles o *generator* e o *engine*. O *generator* tem a função de gerar um ficheiro com as informações do modelo requerido, isto é, criar um ficheiro com os vértices que constituem o modelo. Já o *engine*, é responsável pela leitura de um ficheiro de configuração (escrito em XML) e pela exibição dos modelos obtidos nessa mesma leitura. De notar que este trabalho foi desenvolvido na linguagem de programação C++, juntamente com a API de computação gráfica *OpenGL*.

1.1 Um belo Projeto

Área: Processamento de Linguagens

Vamos agora indicar o conteúdo habitual da introdução de qualquer relatório.

Enquadramento do tema proposto

Contexto do tema que é abordado ao longo do documento

Problema o problema que se quer resolver e o objetivo do projeto relatado

Objetivo do relatório

Resultados ou Contributos – pontos a evidenciar

Estrutura do documento o que é abordado em cada capítulo.

Na Figura 1.1 podemos ver o Layout dos Parâmetros do Formato de Página.

Letras gregas são estas $\alpha\beta\gamma\delta$ que aqui demonstro EM FORMATO INLINE

Exemplo simples de fração múltipla

$$\frac{\frac{a*b+c}{4-3}}{3*5}$$

simples EM FORMATO de DESTAQUE (nova linha)

Exemplo de LISTAS ENUMERADAS com LETRAS:

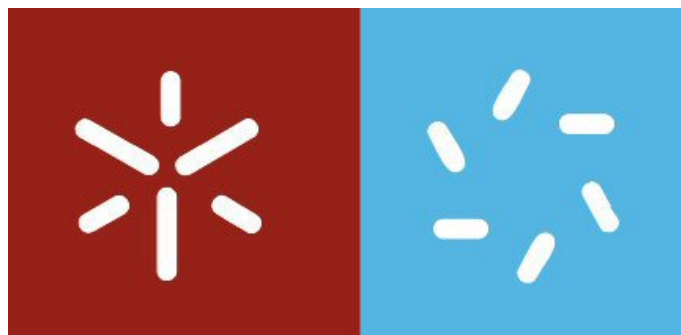


Figura 1.1: Legenda da Figura

- a) Listar todas as Pessoas identificadas, sem repetições;
- b) Listar os Países e Cidades marcadas;
- c) Listar as Organizações.

Mais exemplos de LISTAS ENUMERADAS mas agora com NUMEROS e outras marcas:

- 1. Listar todas as Pessoas identificadas, sem repetições;
 - i) Listar todas as Pessoas identificadas, sem repetições;
 - ii) Listar os Países e Cidades marcadas;
 - iii) Listar as Organizações.
- 2. Listar os Países e Cidades marcadas;
 - 2.1) Listar todas as Pessoas identificadas, sem repetições;
 - 2.2) Listar os Países e Cidades marcadas;
 - 2.3) Listar as Organizações.
- 3. Listar as Organizações.
 - 1) Listar todas as Pessoas identificadas, sem repetições;
 - 2) Listar os Países e Cidades marcadas;
 - 3) Listar as Organizações.

A mesma enumeração mas no standard DESCRITIVO:

Etape 1: Listar todas as Pessoas identificadas, sem repetições;

Etape 2: Listar os Países e Cidades marcadas;

Etape 3: Listar as Organizações.

Texto com cores

texto em azul

texto em vermelho

texto em verde

texto em Castanho

Texto destacado

Texto NEGRITO isto é um texto a NEGRITO

Texto fonte SANS SERIF isto é um texto SANS SERIF

Texto fonte MÁQUINA isto é um texto fonte MÁQUINA

Texto a ITALICO isto é um texto a ITALICO

Texto a SUBLINHADO isto é um texto a SUBLINHADO

Tamanhos de LETRA

largas – large

maiores – Large

muito grandes – LARGE

enormes – huge

as maiores – Huge

minúscula – tiny

muito pequena – scriptsize

bastante pequena – footnotesize

pequena – small

normal – normalsize

Exemplo de tabelas

cell1	cell2	cell3
cell4	cell5	cell6
cell7	cell8	cell9

Tabela 1.1: Tabela Basica

cell1 dummy text dummy text dummy text	cell2	cell3
cell5	cell1 dummy text dummy text dummy text	cell6
cell7	cell8	exemplo de uma linha muito extensa

Tabela 1.2: Tabela em formato array

col1	col2	col3
Multiple row	cell2	cell3
	cell5	cell6cell6
	cell8	cell9cell9cell9
cell4	cell5	cell6

Tabela 1.3: Tabela com multilinhas

Horário de Tópicos em Matemática - MAT					
Horário	Seg	Ter	Qua	Qui	Sex
13:00-14:40					
14:55-16:35					TURMA N
16:35-18:15	TURMA N		TURMA N		
18:15-19:00					

Tabela 1.4: Tabela com multicolumnas

Para mais informações sobre LATEX consultar o livro.

Para mais informações sobre LATEX consultar o livro¹.

¹acessível em http://www.ptep-online.com/ctan/lshort_port.pdf

Capítulo 2

Engine

2.1 Leitura do ficheiro de configuração

Como sugerido, utilizou-se o *tinyxml2* no auxílio à leitura do XML proveniente do ficheiro de configuração. O código do *engine* possui duas funções responsáveis pela mesma, sendo elas:

1. `void readXML(char* filename);`
2. `void readGroupXML(tinyxml2::XMLElement *group);`

De notar que a segunda função é apenas uma auxiliar da primeira função.

2.2 Estruturação dos dados

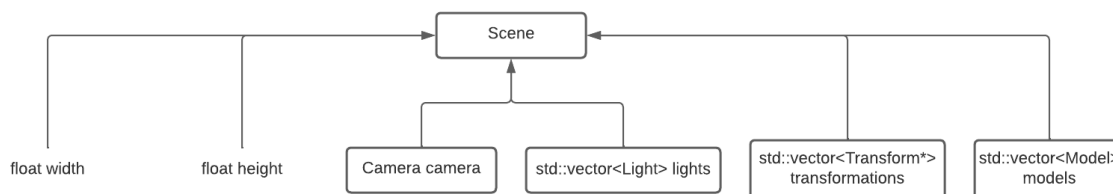


Figura 2.1: Estrutura de dados *Scene*

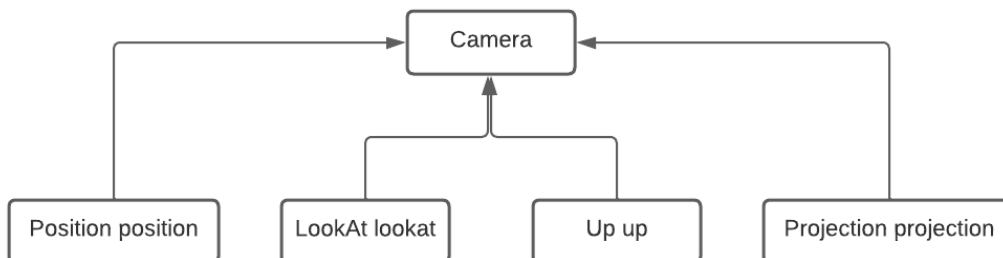


Figura 2.2: Estrutura de dados *Camera*

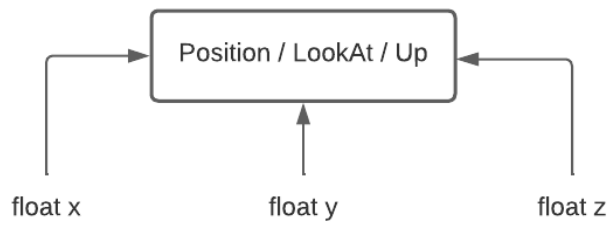


Figura 2.3: Estruturas de dados *Position*, *LookAt* e *Up*

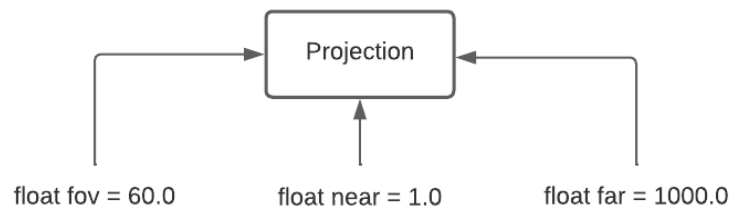


Figura 2.4: Estrutura de dados *Projection*

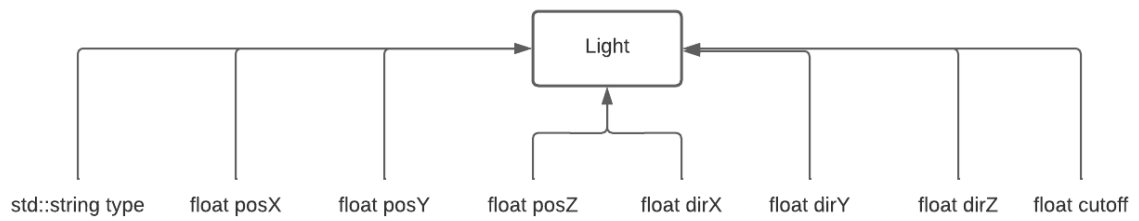


Figura 2.5: Estrutura de dados *Light*

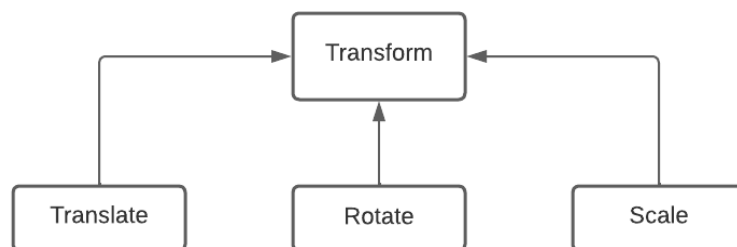


Figura 2.6: Classe *Transform*

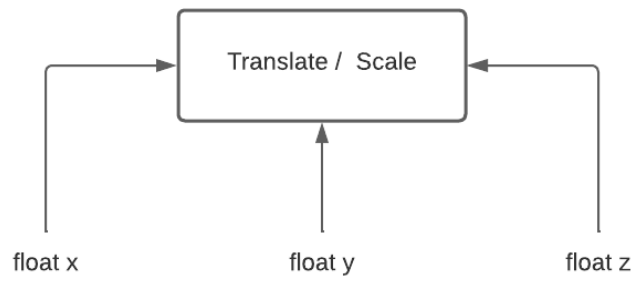


Figura 2.7: Classes *Translate* e *Scale*

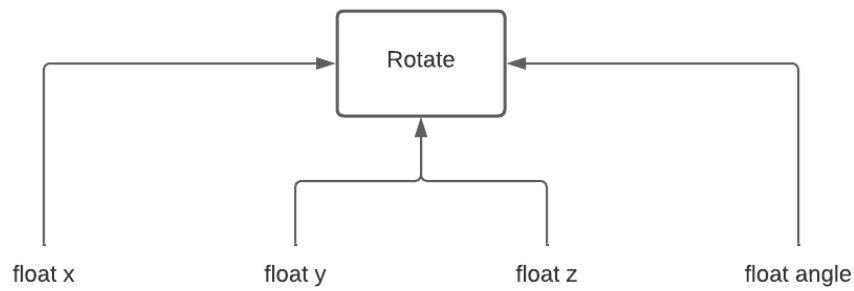


Figura 2.8: Classe *Rotate*

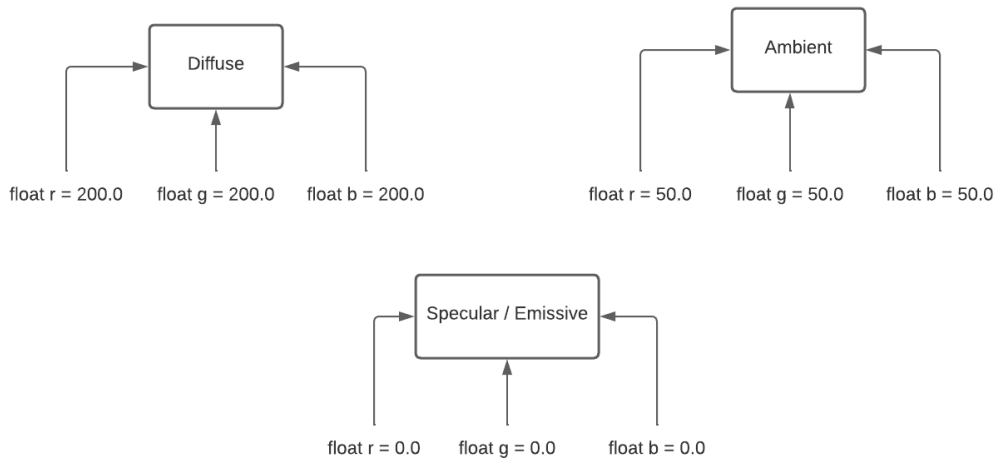


Figura 2.9: Estruturas de dados *Diffuse*, *Ambient*, *Specular* e *Emissive*

As figuras exibidas acima representam as classes e estruturas de dados utilizadas para a estruturação dos dados obtidos na leitura do ficheiro de configuração. De notar que os valores descritos nas figuras indicam os

valores padrão das variáveis em questão, isto é, os valores que as variáveis tomam quando não especificadas no ficheiro de configuração.

2.3 Desenho dos modelos

No que toca ao desenho dos modelos, desenvolveu-se as seguintes funções:

- `void drawAxis();`
Desenha os eixos x , y e z ;
- `void drawModels();`
Desenha os modelos descritos no ficheiro de configuração utilizando a função auxiliar `drawFigure`;
- `void drawFigure(std::string filename);`
Desenha os pontos contidos no ficheiro passado como argumento, resultando na figura pretendida. Utiliza a função `tokenize` como auxiliar;
- `void tokenize(std::string const &str, const char* delim, std::vector<float> &out);`
Recebe uma linha com coordenadas, separa-as e coloca-as num vetor, onde a posição 0, 1 e 2 do vetor correspondem às coordenadas x , y e z , respetivamente;

2.4 Execução do Engine

O comando utilizado para execução do programa *engine* é:

\$: `engine <configFile.xml>`

O "configFile.xml" corresponde ao ficheiro de configuração que se pretende utilizar na execução.

Capítulo 3

Concepção/desenho da Resolução

3.1 Estruturas de Dados

3.2 Algoritmos

Capítulo 4

Testes realizados e Resultados

4.1 Teste nº1

Listing 4.1: XML do Teste 1

```
1 <world>
2   <window width="512" height="512" />
3   <camera>
4     <position x="5" y="-2" z="3" />
5     <lookAt x="0" y="0" z="0" />
6     <up x="0" y="1" z="0" />
7     <projection fov="60" near="1" far="1000" />
8   </camera>
9   <group>
10    <models>
11      <model file="../../3d/cone_1_2_4_3.3d" /> <!-- generator cone 1 2 4 3
              cone_1_2_4_3.3d -->
12    </models>
13  </group>
14 </world>
```

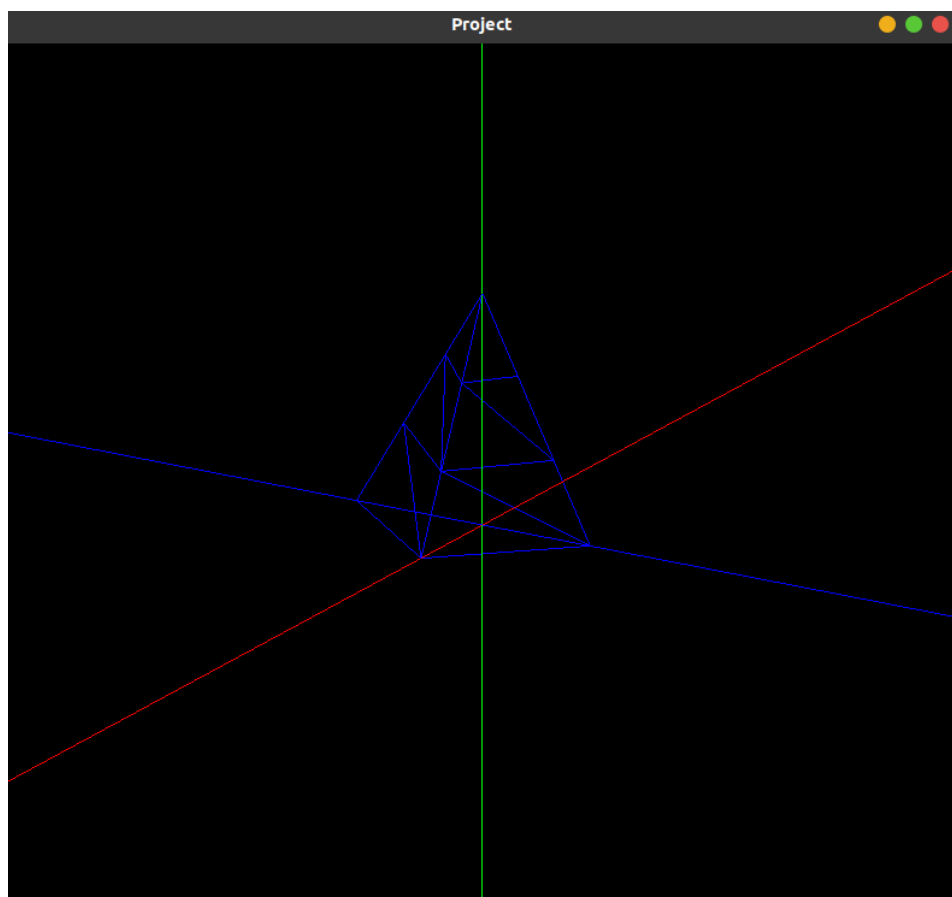


Figura 4.1: Resultado do Teste 1

4.2 Teste nº2

Listing 4.2: XML do Teste 2

```
1 <world>
2   <window width="512" height="512" />
3   <camera>
4     <position x="5" y="-2" z="3" />
5     <lookAt x="0" y="0" z="0" />
6     <up x="0" y="1" z="0" />
7     <projection fov="20" near="1" far="1000" />
8   </camera>
9   <group>
10    <models>
11      <model file="../../3d/cone_1_2_4_3.3d" /> <!-- generator cone 1 2 4 3
12        cone_1_2_4_3.3d -->
13    </models>
14  </group>
15</world>
```

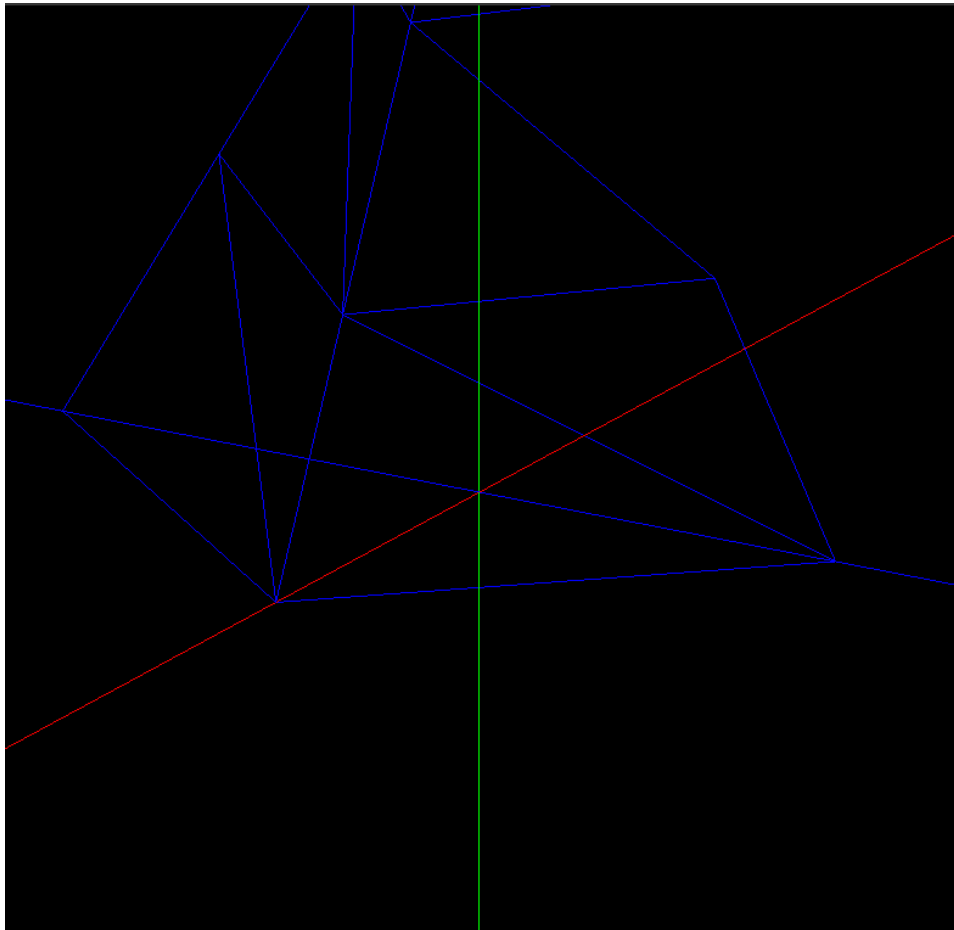


Figura 4.2: Resultado do Teste 2

4.3 Teste nº3

Listing 4.3: XML do Teste 3

```
1 <world>
2   <window width="512" height="512" />
3   <camera>
4     <position x="3" y="2" z="1" />
5     <lookAt x="0" y="0" z="0" />
6     <up x="0" y="1" z="0" />
7     <projection fov="60" near="1" far="1000" />
8   </camera>
9   <group>
10    <models>
11      <model file="../../3d/sphere.1_10_10.3d" /> <!-- generator sphere 1 10 10
12        sphere.1_10_10.3d -->
13    </models>
14  </group>
15 </world>
```

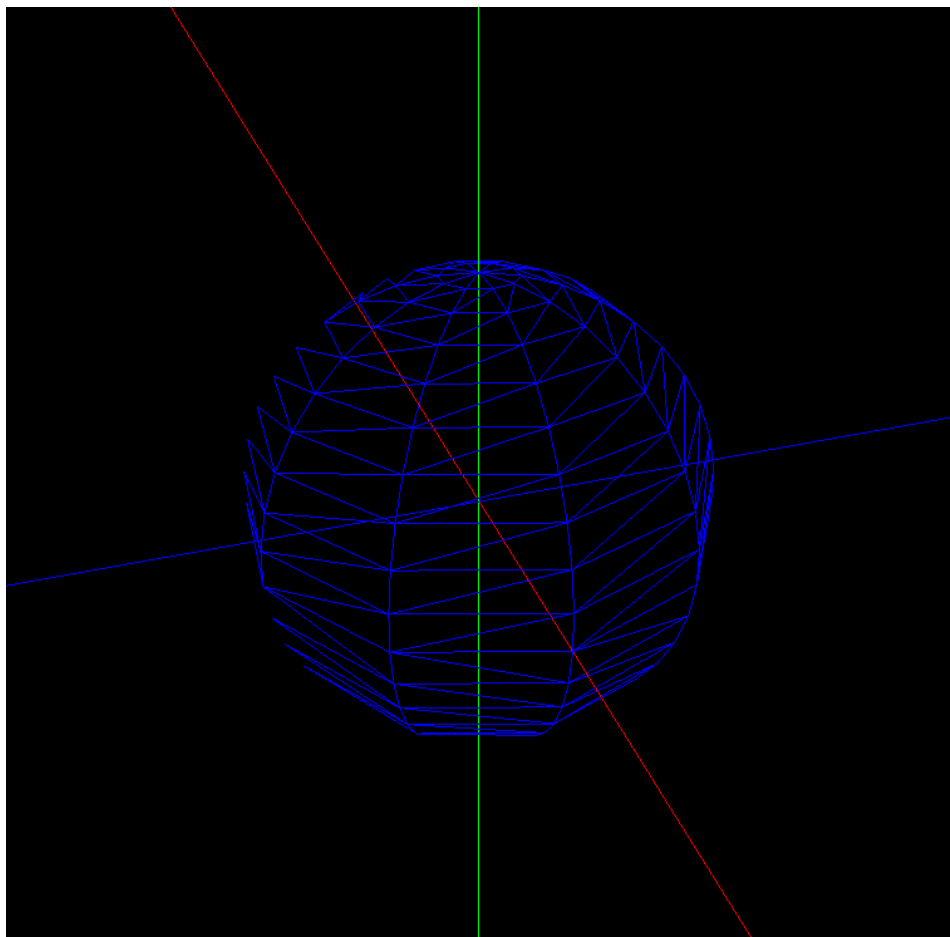


Figura 4.3: Resultado do Teste 3

4.4 Teste nº4

Listing 4.4: XML do Teste 4

```
1 <world>
2   <window width="512" height="512" />
3   <camera>
4     <position x="3" y="2" z="1" />
5     <lookAt x="0" y="0" z="0" />
6     <up x="0" y="1" z="0" />
7     <projection fov="60" near="1" far="3.5" />
8   </camera>
9   <group>
10    <models>
11      <model file="../../3d/box_2_3.3d" /> <!-- generator box 2 3 box_2_3.3d
12        -->
13    </models>
14  </group>
15</world>
```

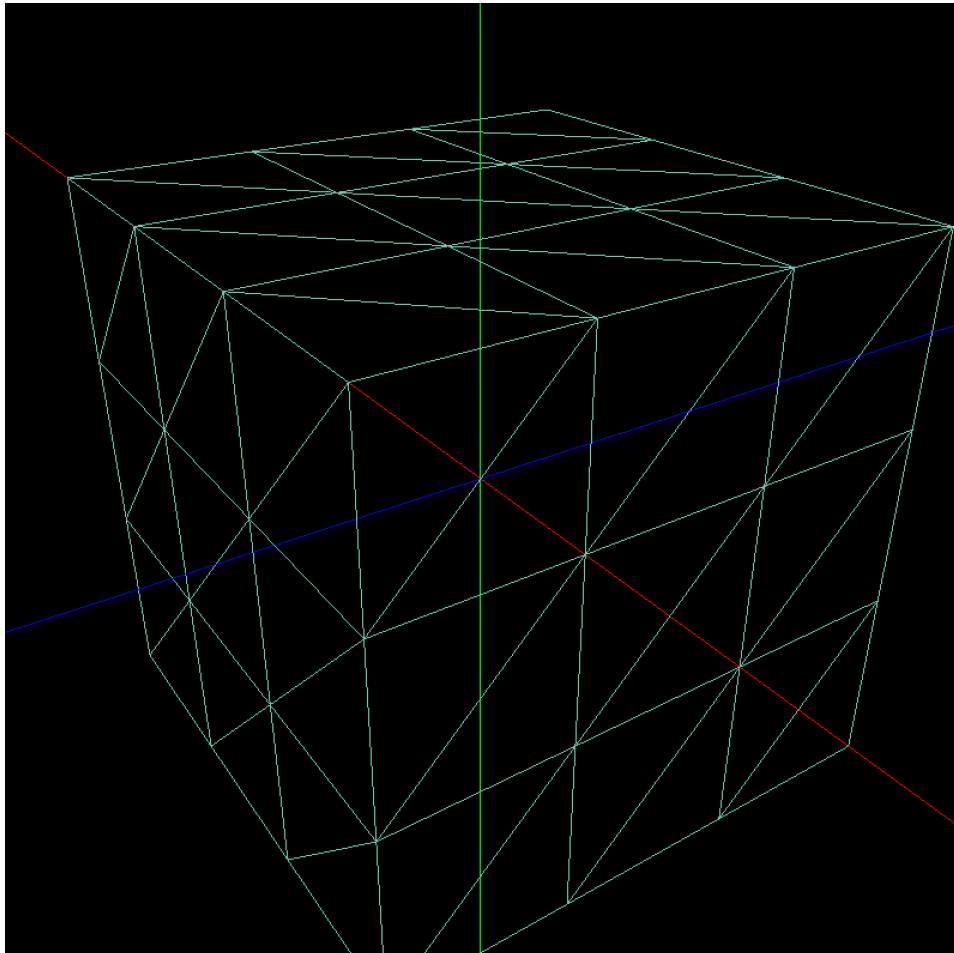


Figura 4.4: Resultado do Teste 4

4.5 Teste nº5

Listing 4.5: XML do Teste 5

```
1 <world>
2   <window width="512" height="512" />
3   <camera>
4     <position x="3" y="2" z="1" />
5     <lookAt x="0" y="0" z="0" />
6     <up x="0" y="1" z="0" />
7     <projection fov="60" near="1" far="1000" />
8   </camera>
9   <group>
10    <models>
11      <model file = "../3d/plane_2_3.3d" /> <!-- generator plane 2 3 plane_2_3
12        .3d -->
13      <model file = "../3d/sphere_1_10_10.3d" /> <!-- generator sphere 1 10 10
14        sphere_1_10_10.3d -->
15    </models>
16  </group>
17 </world>
```

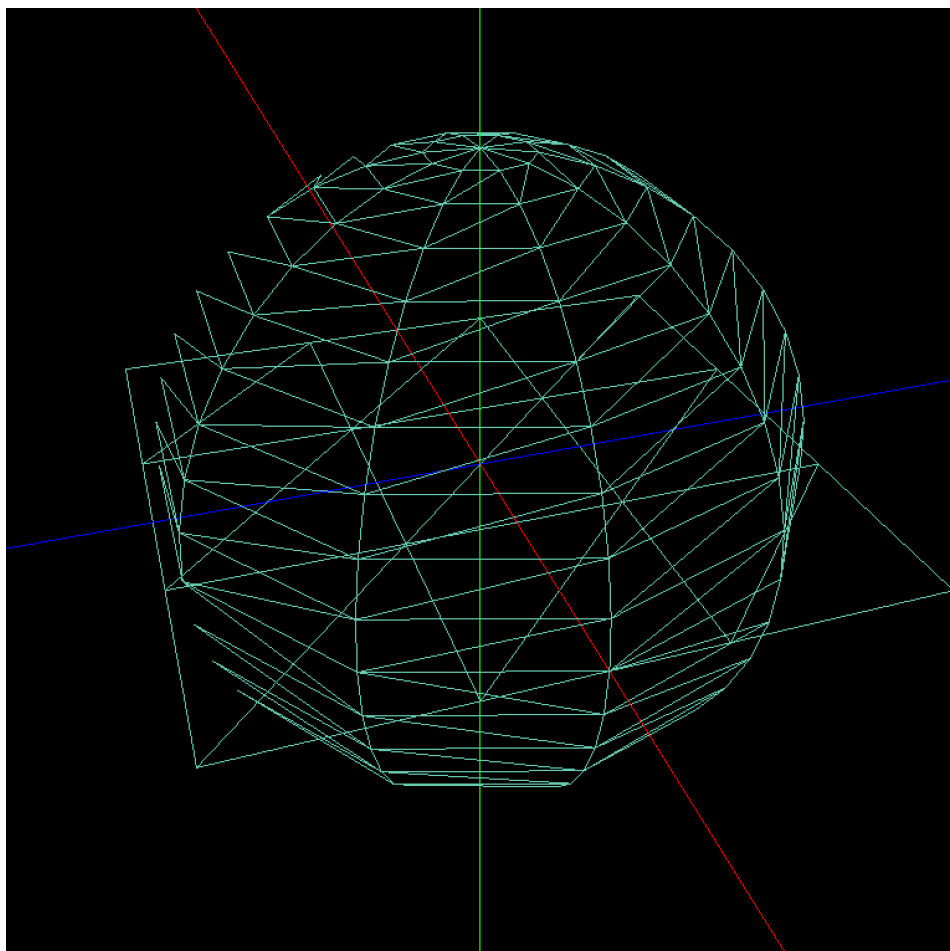


Figura 4.5: Resultado do Teste 5

Capítulo 5

Conclusão

Síntese do Documento [?, ?].

Estado final do projecto; Análise crítica dos resultados [?].

Trabalho futuro.

Apêndice A

Código do Programa

Lista-se a seguir o CÓDIGO AnTLR [?] do programa Darius [?] que foi desenvolvido.

```
public class Aula()
{
    int n, m;
    int max(int a, int b)
    {
        .....
        return(max);
    }
}
```

Lista-se a seguir UM TEXTO (COM O COMANDO VERBATIN)

```
    aqui deve aparecer o código do programa,
    tal como está formatado no ficheiro-fonte "darius.java"
    um pouco de matematica $\$\$
    caso indesejável $\varepsilon$
```

Lista-se a seguir (ver a Listing A.1 abaixo) UM TEXTO não processado porque fixado COM O COMANDO LISTING que embora parecido com o Verbatim é muito mais sofisticado.

Listing A.1: Exemplo de uma Listagem

1	ou entao aparecer aqui neste sitio um pouco de matematica \$
2	como alternativa ao anterior.
3	e aqui mais um teste ε

É ainda possível IMPORTAR UMA LISTAGEM DIRECTAMENTE DO FICHEIRO tal como se ilustra na Listing ??.