



**Universidade do Minho**

**Primitivas Gráficas**  
**Unidade Curricular de Computação Gráfica**  
Licenciatura em Ciências da Computação  
Universidade do Minho

Bruno Jardim  
(A91680)

Inês Presa  
(A90355)

Tiago Carriço  
(A91695)

Tiago Leite  
(A91693)

13 de março de 2022

# Índice

<b>1</b>	<b>Contextualização</b>	<b>2</b>
1.1	Enunciado . . . . .	2
<b>2</b>	<b>Apresentação das soluções</b>	<b>3</b>
2.1	Gerador . . . . .	3
2.1.1	Plano . . . . .	3
2.1.2	Caixa . . . . .	4
2.1.3	Esfera . . . . .	5
2.1.4	Cone . . . . .	6
2.1.5	Cilindro . . . . .	7
2.1.6	Torus . . . . .	8
2.2	Motor Gráfico . . . . .	9
<b>3</b>	<b>Conclusão</b>	<b>10</b>

# Capítulo 1

## Contextualização

No âmbito da unidade curricular de Computação Gráfica da Licenciatura em Ciências da Computação foi proposta o desenvolvimento em *OpenGL* de um motor gráfico genérico que terá como função a criação de um sistema solar. Desenvolvimento esse que deve ser composto por quatro etapas.

### 1.1 Enunciado

Nesta primeira etapa foi proposta a criação de duas aplicações:

- **Gerador**

Funcionalidade: Gerar ficheiros com os pontos no espaço necessários para desenhar primitivas gráficas, como por exemplo, planos, cubos, esferas e cones.

- **Motor Gráfico**

Funcionalidade: Interpretar um ficheiro XML com a configuração da câmara e os ficheiros com os pontos previamente gerados, e desenhar as primitivas.

## Capítulo 2

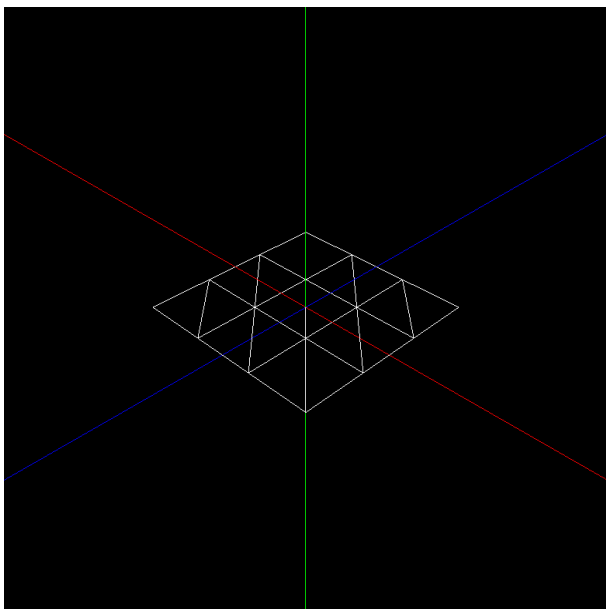
# Apresentação das soluções

### 2.1 Gerador

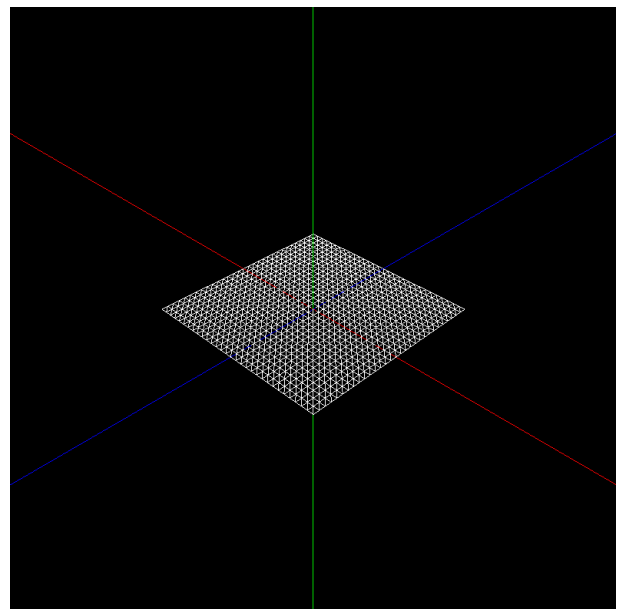
O gerador permite, através de um conjunto de parâmetros, gerar diversos tipos de primitivas gráficas, escrevendo as coordenadas dos pontos que compõem a primitiva pretendida num ficheiro com extensão *.3d* para mais tarde ser carregado e utilizado pelo motor gráfico.

#### 2.1.1 Plano

Para gerar o plano iterou-se no comprimento e largura do mesmo através de dois ciclos *for*, partindo do canto cujas coordenadas  $x$  e  $z$  são negativas, percorrendo cada linha e desenhando simultaneamente os dois triângulos que constituem cada secção do plano.



(a) `./generator plane 1 3 plane.3d`

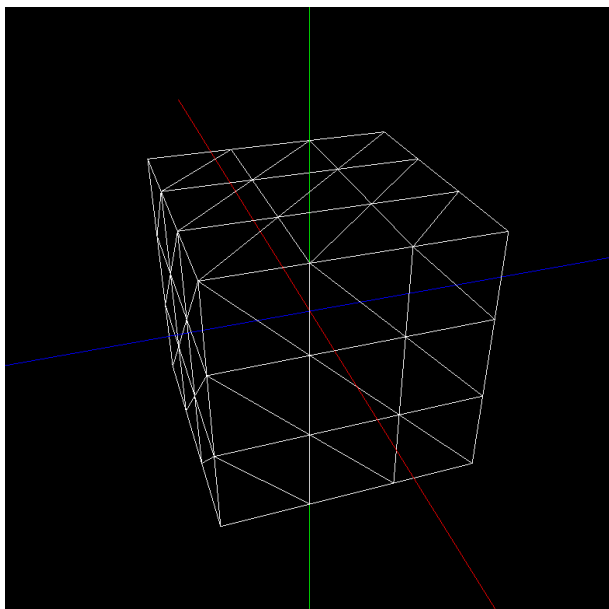


(b) `./generator plane 1 30 plane.3d`

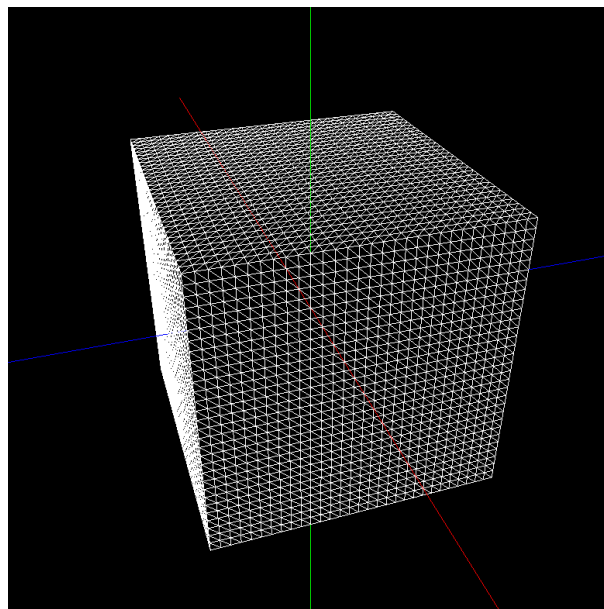
Figura 2.1: Exemplos de planos

### 2.1.2 Caixa

Com a primitiva do plano feita, facilmente se chegou à caixa, uma vez que esta é obtida através de 6 planos paralelos, dois a dois e a cada um dos planos coordenados. Sendo que se deu atenção à sua orientação para que sejam visíveis de fora da caixa.



(a) `./generator box 2 3 box.3d`

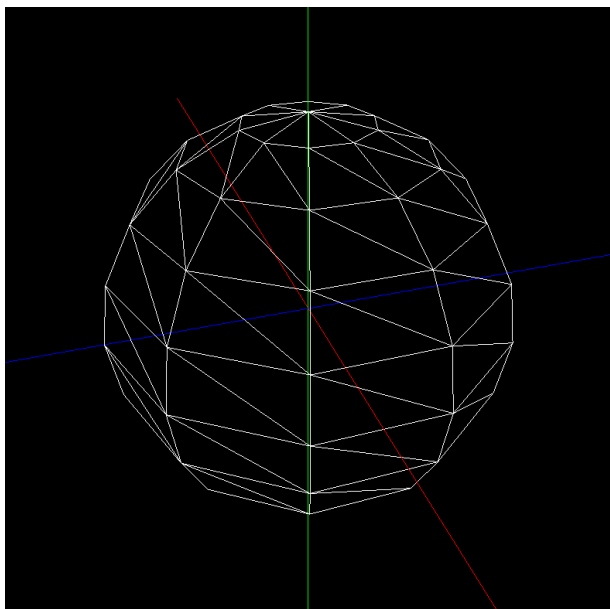


(b) `./generator box 2 30 box.3d`

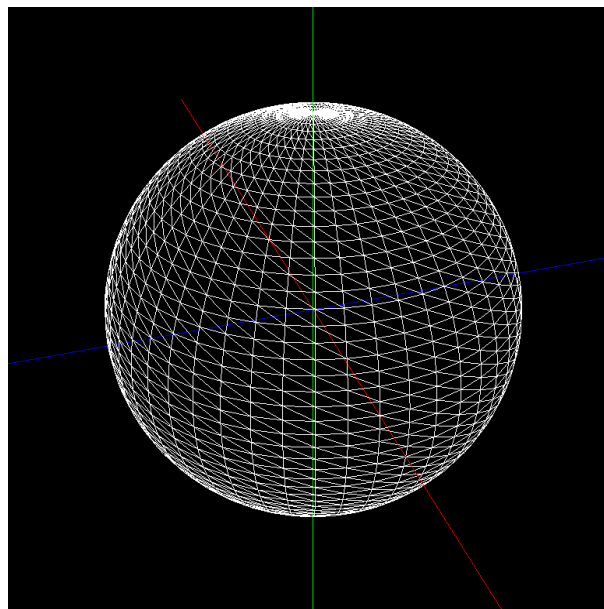
Figura 2.2: Exemplos de caixas

### 2.1.3 Esfera

Começando no pólo superior e iterando por cada *slice* da esfera, construiu-se o sólido, utilizando coordenadas polares para que fosse possível descrever com precisão as coordenadas de cada ponto da esfera, tornando assim a representação o mais fiel possível.



(a) `./generator sphere 1 10 10 sphere.3d`

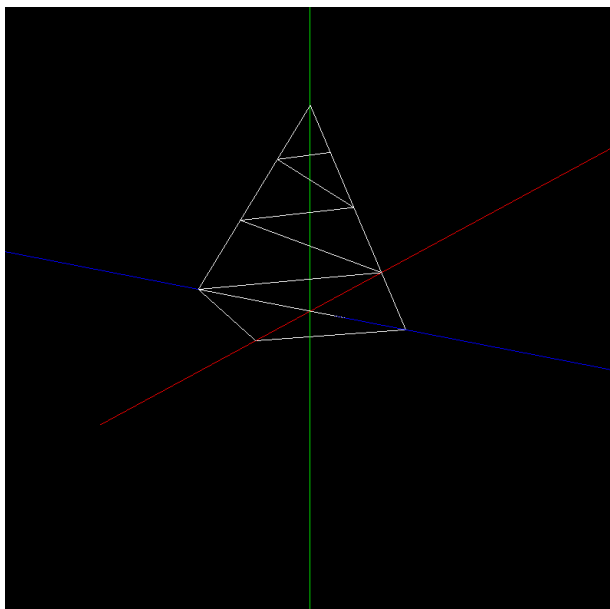


(b) `./generator sphere 1 50 50 sphere.3d`

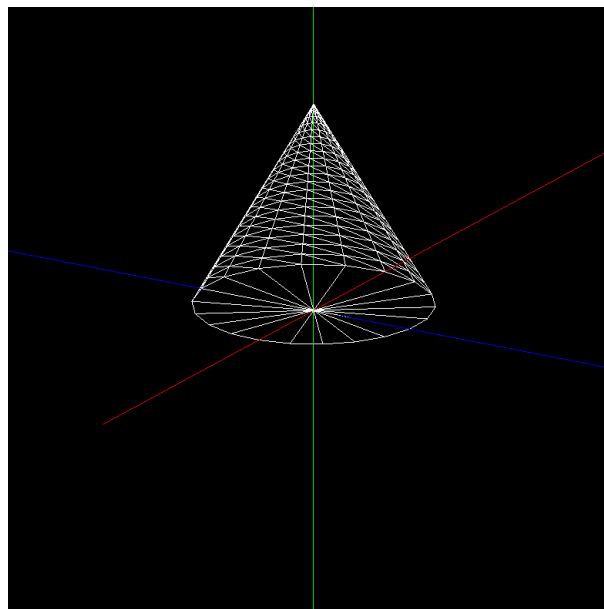
Figura 2.3: Exemplos de esferas

### 2.1.4 Cone

Tendo como ponto de partida a origem do referencial criou-se um círculo sobre o plano  $y = 0$ , este será a base do cone. Com isto foi desenhada, *stack* por *stack*, a face do cone, mais uma vez, utilizando coordenadas polares e o teorema de Tales para descobrir o valor do raio em cada *stack*.



(a) `./generator cone 1 2 4 3 cone.3d`

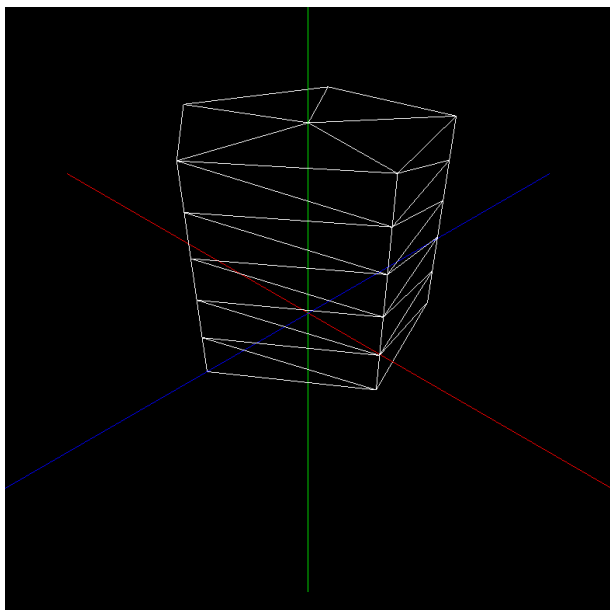


(b) `./generator cone 1 2 20 20 cone.3d`

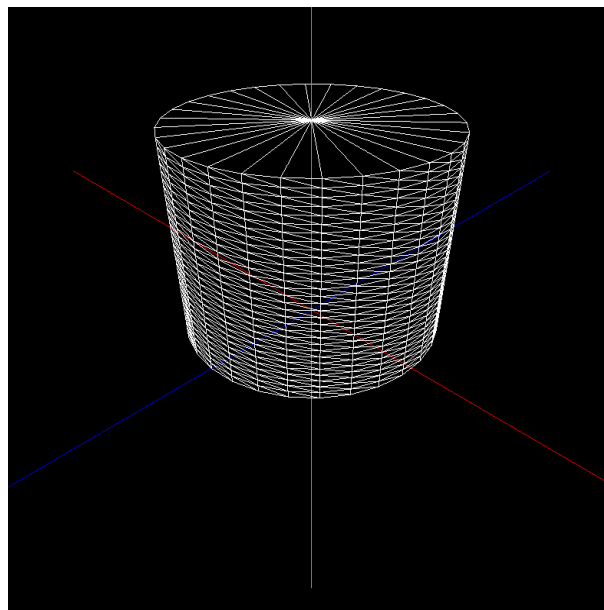
Figura 2.4: Exemplos de cones

### 2.1.5 Cilindro

Utilizando o cilindro desenvolvido nas aulas, foi introduzido um novo parâmetro, o numero de *stacks*, pois assim consegue-se um nível de detalhe superior no cilindro. Com a estratégia usada para construir o cone, fez-se uma alteração para que o raio do cilindro não diminuísse com o aumento da altura, em oposição ao que acontece no caso do cone.



(a) `./generator cylinder 2 3 5 5 cylinder.3d`



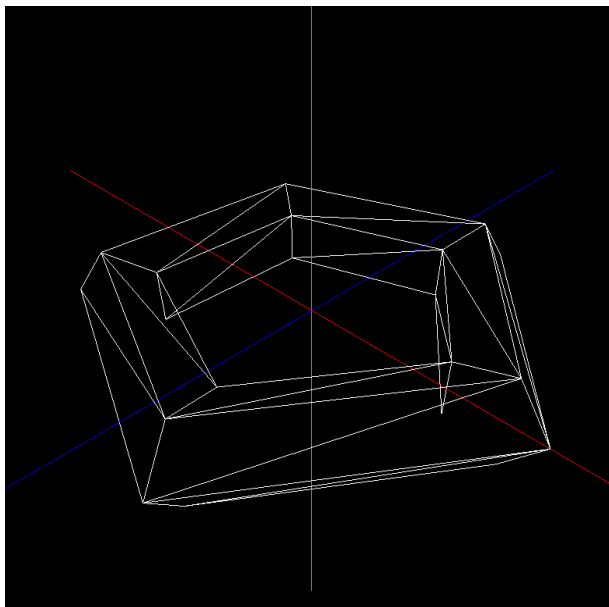
(b) `./generator cylinder 2 3 30 30 cylinder.3d`

Figura 2.5: Exemplos de cilindros

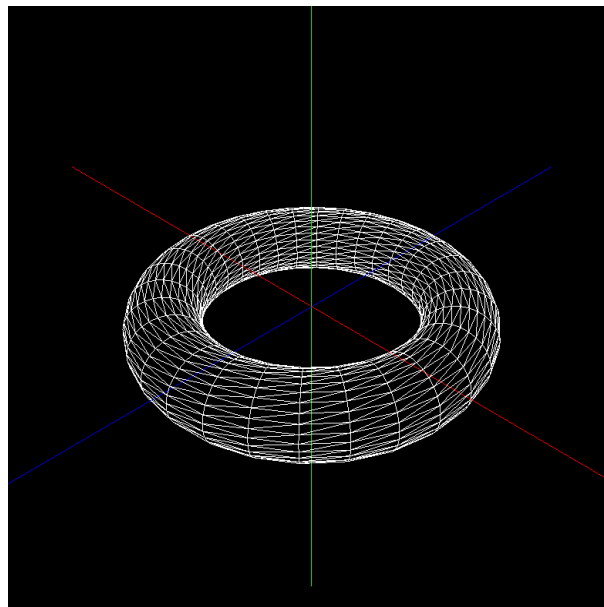


### 2.1.6 Torus

Através das fórmulas disponibilizadas **aqui**, conseguiu-se gerar uma representação de um *torus*, algo que futuramente vai permitir desenhar os vários anéis dos planetas do sistema solar.



(a) `./generator torus 5 3 5 5 torus.3d`



(b) `./generator torus 5 3 30 30 torus.3d`

Figura 2.6: Exemplo de torus

## 2.2 Motor Gráfico

O motor gráfico interpreta um ficheiro XML e com a informação nele contido irá gerar uma cena. Para que o motor tivesse esta funcionalidade, tomou-se vantagem do *parser* **tinyXML**. Após ser executado o *parser* extrai a seguinte informação do ficheiro:

- Câmara
  - Posição da câmara (*position*)
  - Ponto para onde a câmara está a olhar (*lookAt*)
  - Inclinação da câmara (*up*)
  - *View Frustum*
- Modelos
  - Nome dos ficheiros *.3d* a carregar

Após carregar os ficheiros *.3d* os pontos neles contidos são armazenados num vetor de vetores de *Point*, uma estrutura de dados previamente definida, que representa um ponto no espaço. Cada vetor armazena os pontos constituintes de uma primitiva gráfica, permitindo assim carregar múltiplas primitivas na mesma cena.

Mais ainda, o motor gráfico tem implementado uma câmara em modo *explorer* que permite rodar em torno da cena através das *arrowkeys* e com as teclas *Page Up* e *Page Down* possibilita aproximar e afastar, respetivamente, a câmara da origem do referencial.

## Capítulo 3

# Conclusão

A principal dificuldade enfrentada durante esta fase do trabalho foi, inicialmente a adaptação às coordenadas polares, logo ambos o cone e a esfera tornaram-se um desafio. No entanto considera-se que foi ultrapassada agilmente.

A adaptação à linguagem foi também uma dificuldade, já que nenhum elemento do grupo tinha experiência prévia com a linguagem em questão, nomeadamente **C++**.

Por último, a utilização do **parser de XML** de terceiros também condicionou o nosso progresso uma vez que o grupo não estava familiarizado com a ferramenta.

Concluindo, apesar das dificuldades enfrentadas, os objetivos foram atingidos prontamente, no que o grupo considera ser um balanço muito positivo.