

Licenciatura em Ciências da Computação
Computação Gráfica
Graphical primitives

Filipe Barbosa
A77252

Luís Bigas
A76964

Hugo Ferreira
A78555

24 de Fevereiro de 2019

Conteúdo

1	Introdução	2
1.1	Introdução	2
2	Generator	3
2.1	Utilização	3
2.2	Formas Geométricas criadas	3
2.2.1	Plane	3
2.2.2	Box	4
2.2.3	Sphere	5
2.2.4	Cone	6
3	Engine	8
3.1	Utilização	8
3.2	Funcionamento	8
4	Conclusão	9

Capítulo 1

Introdução

1.1 Introdução

No âmbito da unidade curricular computação gráfica, foi proposta a elaboração de um projeto com objetivo a criação de um software gerador de objetos 3D. Nesta fase do trabalho são criadas duas aplicações: uma para criar os vértices de triângulos que representam as figuras pretendidas(generator), outra para realizar a leitura do output do primeiro e com essa informação desenhar as figuras num espaço tridimensional(engine). Para solucionar o problema usamos o C++, em conjunto com as bibliotecas freeglut para a interação com openGL, e TinyXML2 para a leitura de um ficheiro formato XML.

Capítulo 2

Generator

2.1 Utilização

Para realizar a compilação do programa é utilizado o comando **make gen**.

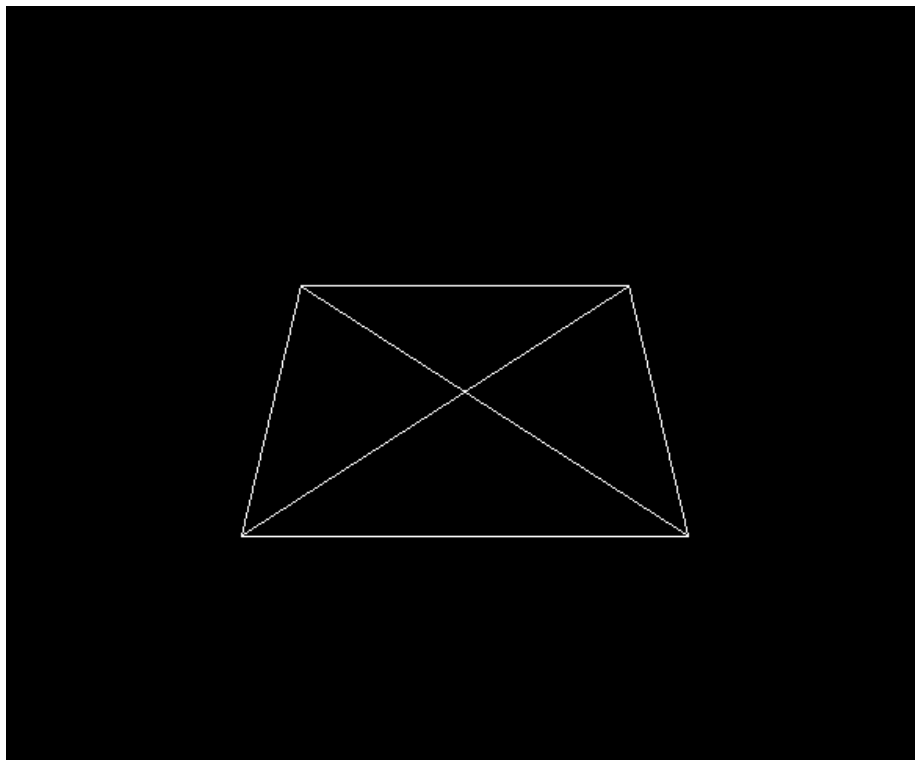
Como argumentos, recebe os parâmetros necessários para a criação da figura geométrica pretendida e o nome do ficheiro output onde são guardados os vértices.

```
./generator "forma pretendida" "pontos" "ficheiro output".3d
```

2.2 Formas Geométricas criadas

2.2.1 Plane

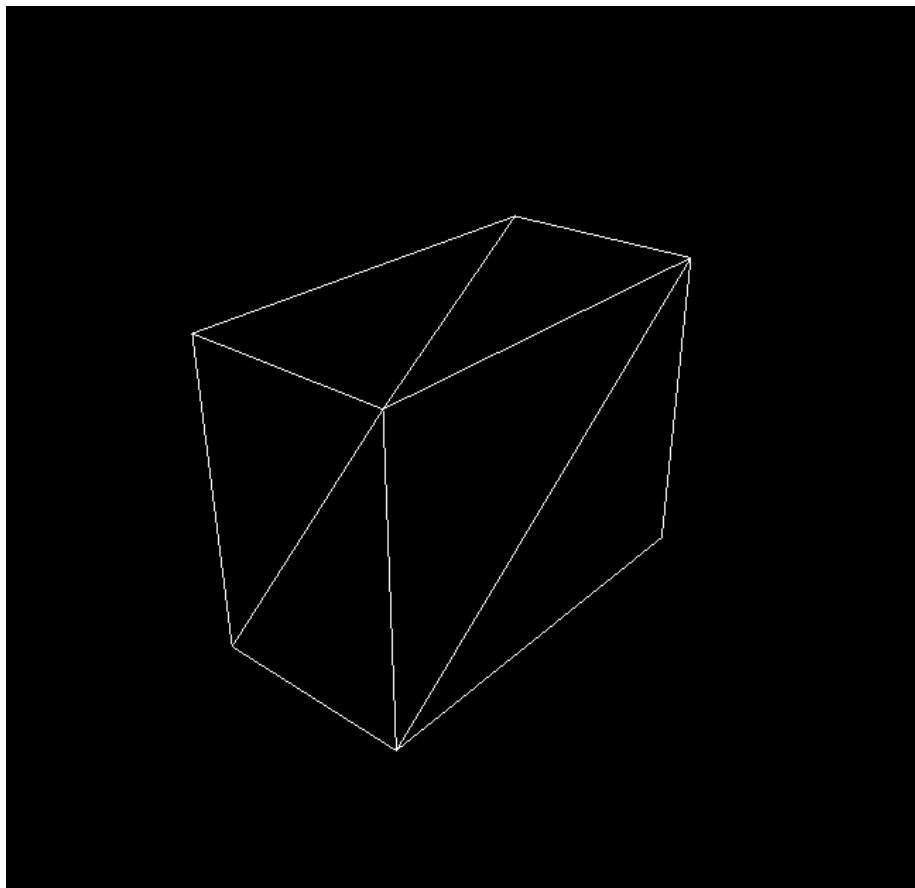
Para criar o plano usamos a função **draw_plane()** que recebe como parâmetro o nome do ficheiro onde vão ser guardadas as coordenadas e o tamanho das arestas. O plano é construído no eixo XZ, centrado na origem, usando quatro triângulos. Começamos por dividir o tamanho da aresta por 2 para centrar o plano na origem.



```
./generator plane 2 plane.3d
```

2.2.2 Box

Para construir a caixa, constituída por 6 faces iguais, usamos a função **draw_box()** que recebe como parâmetros o nome do ficheiro onde vão ser guardadas as coordenadas e as dimensões x,y,z. Começamos por dividir as coordenadas x,y e z por 2 para saber a distância do dentro da caixa em cada um dos eixos, sendo posteriormente calculado os pontos de 2 triângulos para cada uma das 6 faces.



`./generator box 2 3 4 box.3d`

2.2.3 Sphere

Na criação da esfera é utilizada a função **draw_sphere()**, que recebe os seguintes parâmetros: nome do ficheiro output, raio, slices(número de subdivisões na zona equatorial da esfera), e stacks(número de subdivisões entre os pólos). Para o cálculo dos pontos dos triângulos pertencentes à esfera utilizamos as seguintes formulas de trigonometria:

$$\begin{aligned}x &= \text{raio} * \cos(B) * \sin(a) \\y &= \text{raio} * \sin(B) \\z &= \text{raio} * \cos(B) * \cos(a)\end{aligned}$$

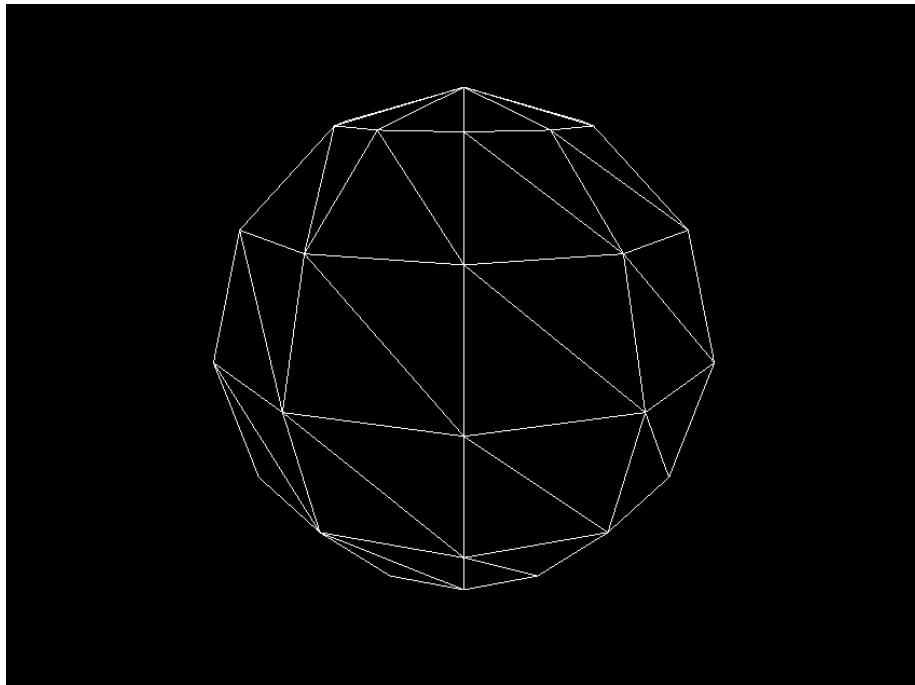
No caso deste trabalho as variáveis foram definidas desta forma:

$$\begin{aligned}B &= (\text{PI} / 2 - j) * (1 / \text{stacks}) \\a &= i * (1 / \text{slices})\end{aligned}$$

Estas são utilizadas dentro de dois ciclos, o primeiro percorre as stacks da

esfera, o segundo percorre as slices.

Em cada Iteração são desenhados 2 triângulos na vertical, definidos através de 4 pontos da esfera.



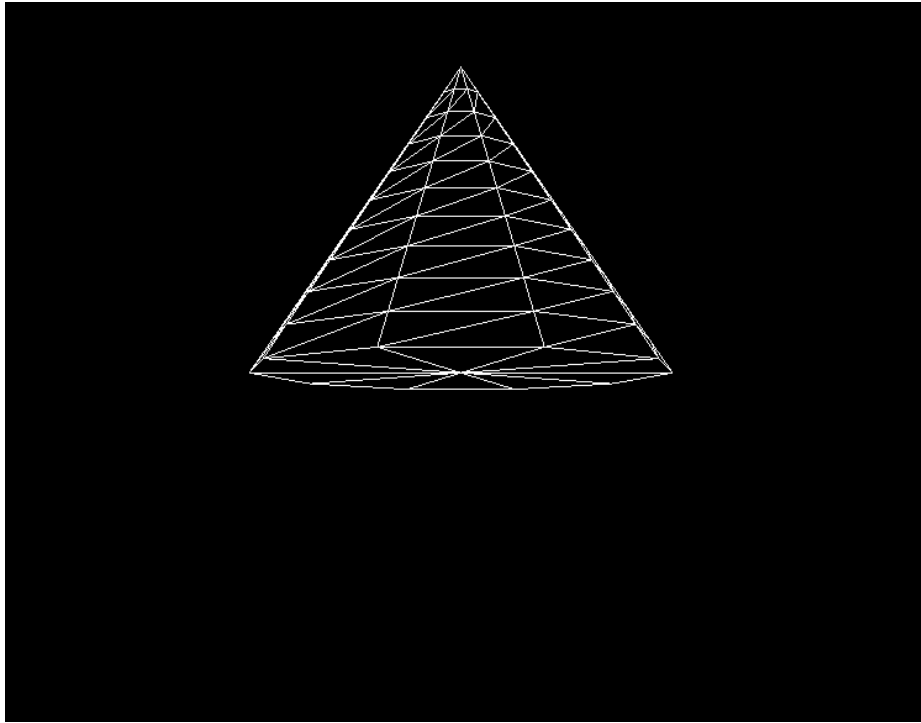
```
./generator sphere 2 10 10 sphere.3d
```

2.2.4 Cone

Para criar o cone usamos a função **draw_cone()** que recebe como parâmetros o nome do ficheiro output, raio, altura, slices e stacks.

Para a construção do cone é feito primeiro o calculo dos pontos que geram a base circular.

Depois calculamos a altura das diferentes stacks que vão existir a partir de altura/stacks, usamos depois um ciclo para sabermos em que slice estamos e dentro deste mesmo ciclo temos outro que nos vai dar a stack em que estamos e calculamos os dois triângulos que vão compôr a stack dessa slice.



./generator cone 2 3 10 10 cone.3d

Capítulo 3

Engine

3.1 Utilização

O ficheiro executável é compilado utilizando o comando **make eng**. A sua execução é realizada com o comando **./engine config.xml**, este irá abrir uma janela gráfica com o conteúdo especificado no ficheiro xml.

Após a abertura da janela é possível ver o objecto pretendido, sendo possível mudar o ângulo de visualização com as setas do teclado, e aumentar/reduzir o nível de zoom com as teclas J e K.

3.2 Funcionamento

O ficheiro executável **engine** recebe como parâmetro o ficheiro config.xml que contem os nomes dos ficheiros criados pelo gerador com as coordenadas dos objetos. Com recurso ao parser **TinyXML**, o ficheiro XML é lido e as coordenadas dos pontos contidas nos ficheiros criados pelo gerador são armazenadas em três vectores com recurso à função **lerCoord**. Após concluída a leitura dos ficheiros e o armazenamento dos pontos, é iniciada a geração dos pontos necessários à criação dos triângulos que compõem as figuras geométricas através de um ciclo que percorre as posições dos vetores usando a função **glVertex3f**.

Capítulo 4

Conclusão

Após a realização desta primeira fase do trabalho conseguimos realizar a criação de diferentes formas geométricas, mesmo após algumas dificuldades no cálculo geométrico. Além disso conseguimos o alargamento dos nossos conhecimentos a novas ferramentas computacionais, que nos possibilita expandir o nosso desenvolvimento de software para um mundo tridimensional.