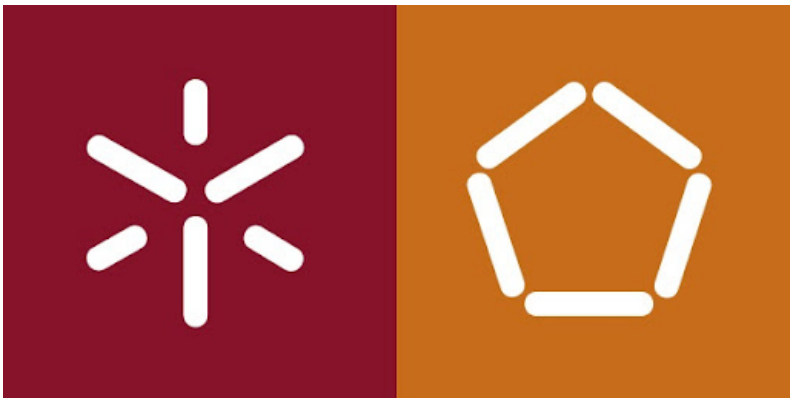
# Universidade do Minho

## Departamento de informática



# Computação Gráfica

Relatório da primeira fase

Grupo nº28


Hugo dos Santos Martins
A95125

Ivo Miguel Alves Ribeiro
A96726


João Bernardo Teixeira Escudeiro
A96075


**8 de março de 2023**

# Índice:

# Introdução e objetivos da primeira Fase:

A primeira fase do trabalho, que consiste na criação de primitivas gráficas, surge no âmbito da Unidade Curricular de Computação Gráfica. A par do desenvolvimento da matéria lecionada nas aulas Teóricas, bem como dos exercícios realizados nas aulas Teórico-Práticas, é pedido que nesta fase precoce do projeto sejam criadas duas aplicações distintas, cada uma delas com funções distintas mas que ambas juntas irão funcionar numa simbiose para a criação das figuras pedidas pelo utilizador.

# Metodologia Utilizada:

O grupo optou por seguir uma metodologia que se assemelhava ao funcionamento das aulas Teórico-Práticas, em que era disponibilizado um conjunto de ficheiros, e a partir dos mesmos serem completados com a respetiva criação de figuras e movimentação da posição da câmera.

A partir de um script fornecido optou-se por proceder primeiro à criação das figuras e respetiva criação de vértices , para posteriormente se verificar que tudo estava a ser realizado corretamente, e proceder à criação das duas aplicações: o *generator* e o *engine*.

**Plano**: O plano foi a figura mais simples, dado o facto de todas as coordenadas relativamente ao eixo dos yy serem iguais. O plano criado  é sempre paralelo ao plano xOz, e necessita de dois parâmetros para ser gerado, que são o tamanho lateral e o número de divisões. Os vértices são calculados através de contas simples, pois aqui não há ângulos, apenas se calcula as posições relativas aos eixos do xx e dos yy. São gerados os primeiros três vértices do triângulo ,e de seguida são gerados os mesmos vértices mas na ordem inversa para formar o quadrado. É de notar que a posição dos vértices é gerada com cálculos diferentes caso o número de divisões seja ou não par.

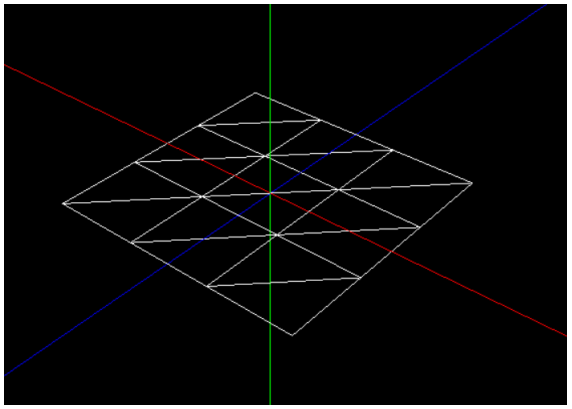


Figura 1- Plano gerado com 3 de tamanho lateral e 3 divisões.

**Cone** : O cone foi a primeira figura a ser gerada . Para a geração do Cone são necessários três parâmetros : o raio da base, a altura, o número de fatias e o número de *stacks*. O cone nada mais é do que um conjunto de triângulos que formam a superfície cónica. Após proceder ao cálculo de todos os ângulos necessários e posterior cálculo dos vértices chegou-se à função que após inseridos os parâmetros as figuras eram criadas. A nossa função começa por definir o ângulo entre cada segmento do cone que é calculado dividindo $2\pi$ pelo número de segmentos. Após isto é calculada a altura de cada segmento vertical, dividindo a altura por o número de *stacks* fornecido. De seguida ela calcula o raio da base em cada fatia vertical. Segue-se o cálculo da posição de cada vértice na base, seguido pelo cálculo da posição de cada vértice na lateral do cone.

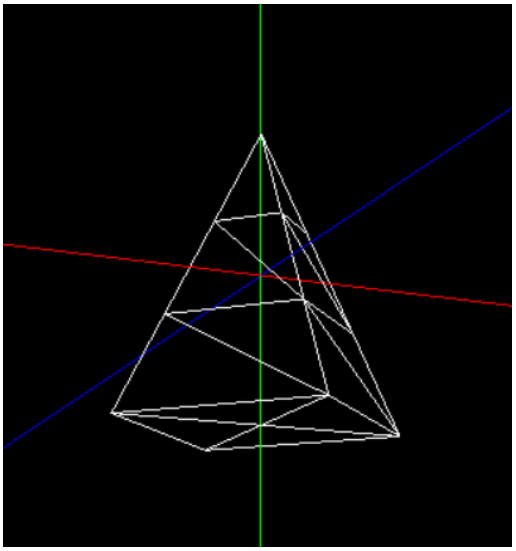


Figura 2- Cone gerado com 1 de raio, 2 de altura, 4 fatias e 3 *stacks*.

**Esfera**: Após a criação do cone passou-se para a criação da esfera. Esta figura necessitava de três parâmetros, nomeadamente o raio, o número de fatias e o número de *stacks*. A esfera foi o sólido que mais trabalho deu devido à complexidade dos ângulos do mesmo. Após uma tentativa inicial partindo utilizando várias circunferências, rapidamente nos apercebemos que não era a melhor forma de pensar pois para a criação das *stacks* seria bastante mais difícil. Após um vasto número de tentativas e erros, o grupo conseguiu, finalmente, gerar esferas válidas. O cálculo dos vértices é realizado da seguinte forma: Para cada fatia longitudinal é calculado um ângulo “theta1” que varia entre 0 e $\pi$, representado a posição da fatia em relação aos pólos norte e sul da esfera. É também calculado o ângulo “theta2” para a próxima fatia longitudinal. É também calculado um ângulo “phi1” que varia entre 0 e $2\pi$, representando a posição da fatia relativamente ao meridiano principal da esfera. Da mesma forma é calculado o ângulo “phi2” que auxilia no cálculo da próxima fatia longitudinal.

Através da utilização de coordenadas esféricas de raio,"theta1" e “phi1” são calculadas as coordenadas cartesianas do primeiro vértice.Da mesma forma se comporta para os vértices seguintes.

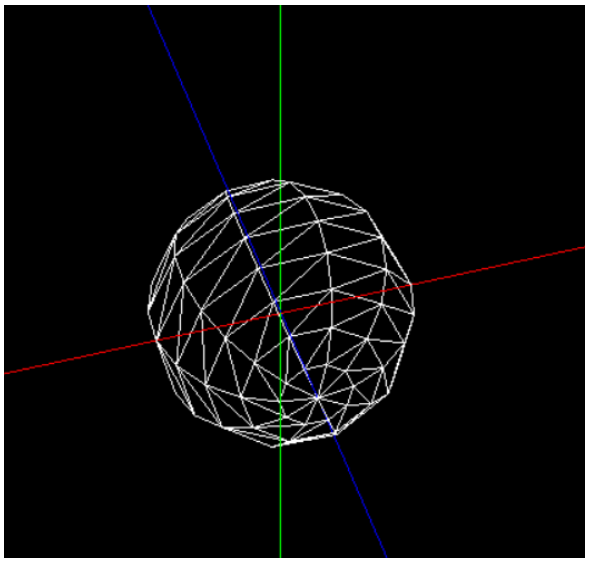


Figura 3- Esfera gerada com 1 de raio, 10 fatias e 10 *stacks*.

**Cubo**: Após a criação do plano, a criação do cubo ficou bastante mais fácil, visto que um cubo nada mais é do que um conjunto de 6 planos paralelos dois a dois e perpendiculares entre si. Para a criação desta figura é necessária a introdução dos seguintes parâmetros: tamanho e divisões. A função de criação usa dois “loops for” aninhados para calcular as coordenadas de todos os vértices. O cálculo das coordenadas de cada vértice são baseados no tamanho do cubo e número de *stacks*.

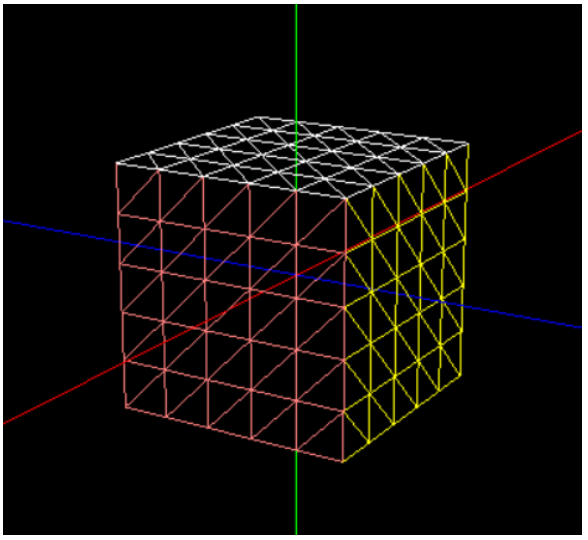


Figura 4- Cubo gerado com 1 de tamanho lateral e 5 divisões.

Após o grupo verificar que as figuras estavam a ser geradas corretamente, decidiu dividir o trabalho nas duas aplicações pedidas: o ***Generator*** e a ***Engine***.

# ***Generator*** **:**

Como pedido no enunciado, foi criada a aplicação do Generator. Como o próprio nome indica esta aplicação quando invocada vai receber um conjunto de argumentos, um deles indicativo de qual será a figura a gerar e outros os argumentos imprescindíveis para a criação das figuras (altura ou raio, fatias, *stacks* ou lado). Assim, e após fazer o parse dos argumentos, é chamada a função correspondente à figura pedida que escreve os vértices resultantes num ficheiro cujo caminho também é fornecido nos argumentos. Após esta aplicação ser executada corretamente obtém-se um ficheiro resultado que contém toda a informação relativa aos vértices da figura a gerar (com os parâmetros inseridos), pronta a ser lida posteriormente pela *engine*.

# ***Engine*** **:**

Para o *engine* é necessário a existência de um ficheiro xml. Este ficheiro contém toda a informação relativa às figuras (apontador para o ficheiro em que se encontra a informação relativa aos vértices gerados pelo generator), posições da câmera, e posteriormente translações, ou rotações relevantes (em fases futuras).

Para fazer o Parser do ficheiro o grupo recorreu a uma biblioteca sugerida no enunciado deste trabalho prático a *tinyxml*. Esta biblioteca auxilia no *parsing* do conteúdo do ficheiro em formato xml. Assim, foi necessário a leitura linha a linha e armazenamento da informação em variáveis globais. Exemplos destas variáveis são por exemplo as posições x,y e z da câmera que variam mediante o conteúdo do ficheiro xml. De igual forma funciona para os vértices cujas posições estão contidas num ficheiro auxiliar mencionado na aba “<model>” do xml. O conteúdo deste *model* é um apontador para o ficheiro gerado pelo *generator* que será lido através de um parser adicional que faz o parsing linha a linha da informação dos vértices  dos vértices e guarda numa variável global (vetor) que irá posteriormente ser utilizada para gerar as figuras. Após os vetores estarem carregados de informação relativa aos vértices, as figuras são geradas e é feito o display da janela.

Nesta primeira fase do trabalho optou-se pela não criação de uma Makefile, e para compilar e correr a *engine* é necessário correr os seguintes comandos:

```
ivo@ivoribeiro:~/Downloads$ gcc engine.cpp -o engine -lGL -lGLU -lglut -lstdc++ -lm -ltinyxml
ivo@ivoribeiro:~/Downloads$ ./engine test.xml
```

Figura 5- Comandos para compilar e executar a *engine* com as bibliotecas necessárias.

```
ivo@ivoribeiro:~/Downloads$ g++ generator.cpp -o generator
ivo@ivoribeiro:~/Downloads$ ./generator plane 1 3 plane.3d
```

Figura 6- Comandos para compilar e executar o generator com os argumentos.

Adicionalmente, e após a criação das figuras, o grupo adicionou algumas funcionalidades no que toca à parte visual, em que clicando nas setas ou movendo o rato é possível observar a câmera a mover-se e a alterar a sua posição permitindo ao observador ver o sólido de diferentes perspectivas.

# ***Sumário e discussão de resultados*:**

Tendo em conta os objetivos da primeira fase do projeto que se baseiam na criação de sólidos a partir de um ficheiro xml que contém toda a informação relativa às figuras, o grupo considera que o trabalho desenvolvido foi positivo, visto que estes objetivos foram todos concluídos na íntegra.

Fazendo uma reflexão do trabalho realizado, o grupo teve algumas dificuldades de adaptação inicial visto que era uma linguagem nova que nunca tinha sido utilizada. Na realização do parser surgiram também algumas dificuldades, pois a biblioteca tinyxml era uma biblioteca com muitas funções novas, com as quais o grupo não estava familiarizado. Na criação dos sólidos, o mais difícil foi a esfera visto que era necessária a inclusão de dois ângulos.