Relatório do Projeto de CG

João Pimentel (a80874) Rodolfo Silva (a81716)
Pedro Gonçalves (a82313)

Março 2019

Universidade do Minho
Mestrado Integrado em Engenharia Informática
Computação Gráfica
Grupo 25
1ª Fase

Conteúdo

Lista de Figuras

1 Introdução

Este relatório aborda a resolução da primeira fase do projeto prático da Unidade Curricular, Computação Gráfica.

Esta etapa está divida em duas fases, sendo que a primeira consiste na criação de um programa gerador de ficheiros com os vértices de uma figura e a segunda na elaboração de um motor que permita a leitura desses mesmos ficheiros e os transforme em figuras *3D*.

Além disso, a etapa tem como propósito a solidificação do conhecimento relativo a primitivas gráficas, bem como o uso da ferramenta *OpenGL*.

2 Primitivas Gráficas

2.1 Plano

Para desenhar um plano utilizando o *OpenGL* é necessário apenas representar dois triângulos, conhecendo os valores dos lados X e Z, neste caso. Esta divisão do plano resulta em dois triângulos retângulos, como se pode observar na Figura 1.

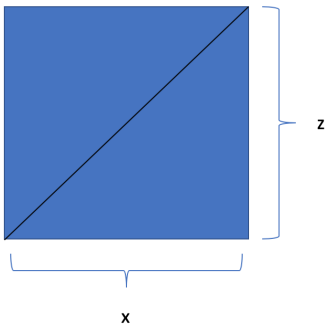

Figura 1 - Plano formado por dois triângulos.

Assim, após a codificação e execução do código que implementa o problema, obteve-se o resultado apresentado na Figura 2.

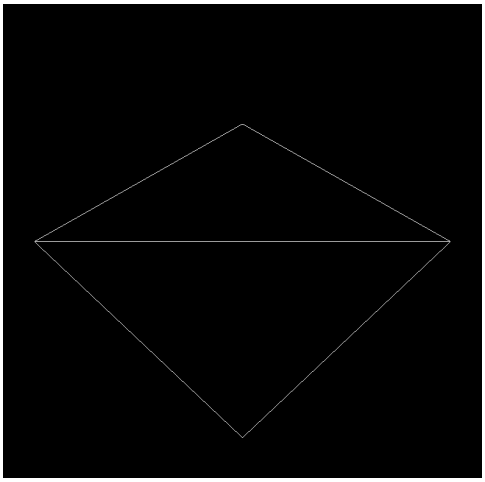

Figura 2 - Representação de um plano em *OpenGL*.

2.2 Caixa

De modo a retratar uma caixa, existem três medidas essenciais a conhecer, como se vê na Figura 3, sendo estas a altura, o comprimento e a largura. Além disso, foi pedido que cada face fosse dividida em N partições, levando à existência de um parâmetro extra.

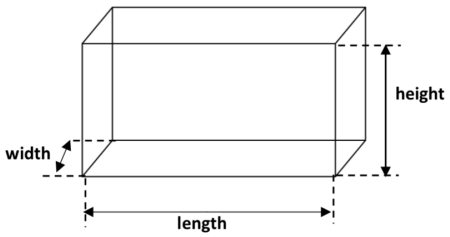

Figura 3 - Esboço de uma caixa.

Assim, a ideia passou por utilizar planos de modo a desenhar a figura completa, tendo em consideração a orientação dos triângulos de modo a ser visível pelo exterior.

No que toca à equação por trás de cada ponto, começou-se por gerar todas as colunas de uma fila até passar à próxima, seguindo a ideia apresentada na Figura 4, que retrata a face frontal da caixa.

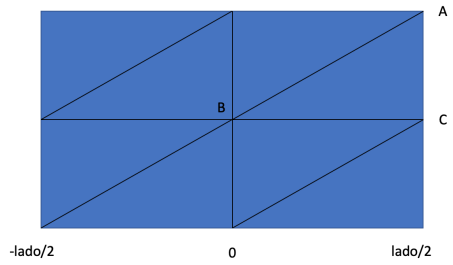

Figura 4 - Divisão de uma face de uma caixa em triângulos.

Ou seja, o ponto A terá as coordenadas $(comprimento/2, altura, largura/2)$, o B terá $(comprimento - largura/divisões, altura - altura/divisões, largura/2)$ e o ponto C terá $(comprimento/2, altura - altura/divisões, largura/2)$. Aplicando esta fórmula num ciclo, torna-se simples gerar um conjunto de faces com as divisões pretendidas, de forma a formar uma caixa.

Utilizando a *framework OpenGl*, obteve-se a Figura 5.

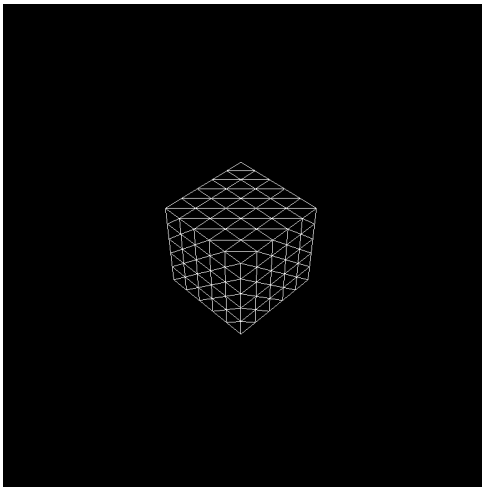

Figura 5 - Representação de uma caixa em *OpenGL*.

2.3 Cone

Para representar um cone é necessário o fornecimento de 4 variáveis: raio, altura, número de *slices* e, por fim, o número de *stacks*.

Começando por utilizar coordenadas polares de modo a obter a base do cone, como se vê na Figura 6, teve que se dividir o ângulo 2π pelo número de *slices*, sendo este o valor a utilizar no ciclo da base.

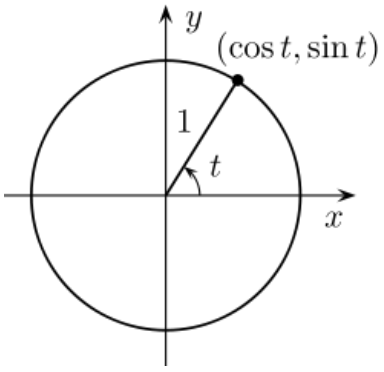

Figura 6 - Coordenadas polares.

Em seguida, tiveram que ser geradas todas as camadas da lateral do

cone. De forma a fazê-lo, a altura foi dividida pelo número de *stacks*, permitindo que, juntamente com a utilização de um ciclo, se formassem pequenos triângulos em fila, com a inclinação necessária.

Assim, foi obtida a Figura 7.

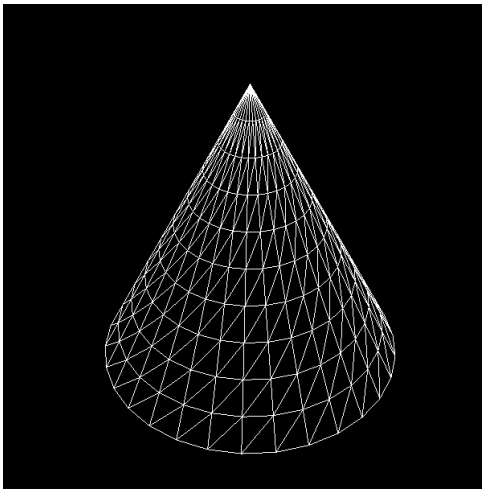

Figura 7 - Representação de um cone em *OpenGL*.

2.4 Esfera

A principal ideia para desenvolver uma esfera é a utilização de coordenadas esféricas, retratadas na Figura 8.

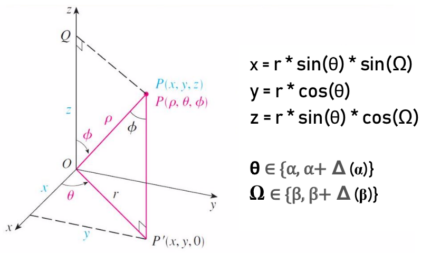

Figura 8 - Coordenadas esféricas.

Assim, com auxílio desta noção, cada conjunto de dois triângulos obedecerá à fórmula retratada na Figura 8, tendo em consideração que será aplicada num ciclo. Note-se que $\Delta\beta = \pi/stacks$ e $\Delta\alpha = 2\pi/slices$.

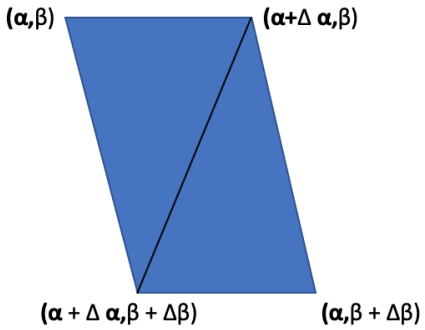

Figura 8 - Explicação do desenho de uma esfera.

Assim, a implementação de uma esfera em *OpenGL* gerou a Figura 9.

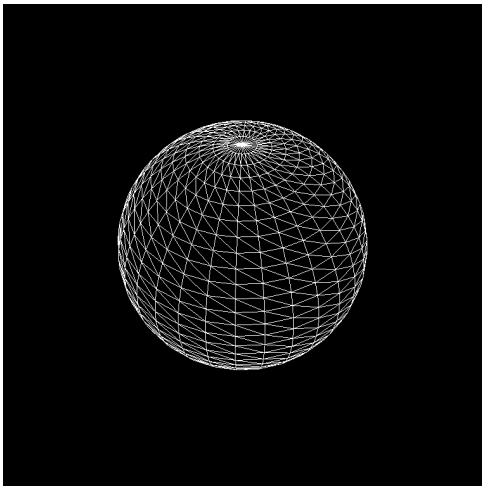

Figura 9 - Representação de uma esfera em *OpenGL*.

3 Generator e Engine

De modo a corresponder aos requisitos fornecidos, teria que existir um programa *generator* que escrevesse em ficheiro os pontos de uma figura e um *engine* que os lesse e desenhasse numa janela.

Utilizando as bibliotecas de *C++*, foi simples codificar o primeiro programa, bastando escrever os valores das coordenadas dos pontos, ficando um por linha. Por questões de controlo, a primeira linha do ficheiro indica o número total de linhas com conteúdo útil.

No que toca ao *engine*, foi necessário implementar um *parser* de *XML*, extraindo os ficheiros onde era suposto encontrar a informação dos pontos a desenhar. Assim, com auxílio da biblioteca *tinyxml2*, estes foram facilmente encontrados.

Em seguida, a cada leitura de um ficheiro com pontos implica que estes necessitam de ser guardados em memória, tendo sido, por isso, definida uma estrutura de dados que possui nove valores, relativos às coordenadas de um triângulo, para que seja, depois, desenhado. Como são lidos muitos mais do que apenas um triângulo, foi utilizado um vetor de triângulos para armazenar todos os encontrados para uma única figura. Dito isto, de modo a desenhar mais que uma componente, é utilizado um vetor de vetores, representando a totalidade das figuras a implementar. Assim, aplicando uma função que, recebendo este vetor, desenhe todas as figuras necessárias, a primeira fase do problema fica completamente realizada.

4 Conclusões e Trabalho Futuro

O desenvolvimento desta primeira fase do projeto permitiu ao grupo solidificar o conhecimento sobre primitivas gráficas, bem como a forma de as implementar utilizando a *framework OpenGL*.

Posto isto, o grupo considera que esta primeira fase foi um sucesso, permitindo boas bases para a elaboração das seguintes etapas do projeto.