

Computação Gráfica (3^o ano)

Trabalho Prático

Relatório

João Miguel Faria Leite
(A94285)

Joel Vieira Barros
(A94692)

Sandro Manuel Fernandes Duarte
(A94731)

10 de março de 2023

Resumo

Este documento visa apresentar o trabalho prático realizado no âmbito da Unidade Curricular de Computação Gráfica, que tem como objetivo o desenvolvimento de várias primitivas gráficas em 3D.

No desenvolvimento das primitivas gráficas foram utilizados conceitos tais como a definição de objetos através de vértices e faces, transformações geométricas e projeções.

Após a conclusão desta fase, foram obtidos os resultados esperados, com as primitivas a corresponder às especificações propostas no enunciado.

Conteúdo

1	Introdução	2
2	Estrutura do código	3
2.1	Generator	3
2.2	Engine	3
3	Primitivas	4
3.1	Plano	4
3.2	Box	4
3.3	Sphere	5
3.4	Cone	5
4	Conclusão	6

Capítulo 1

Introdução

Para a primeira fase do Trabalho Prático foi desenvolvido um programa que permite elaborar diferentes primitivas gráficas, de acordo com os inputs recebidos pelo utilizador, podendo estas ser um plano, uma caixa, uma esfera e um cone.

Nesta fase do projeto foram propostas duas aplicações distintas: o Generator - capaz de criar ficheiros 3D com informação dos modelos (os vértices de cada primitiva) e o Engine - que permite visualizar os modelos criados no Generator, através de um ficheiro XML.

Para a implementação deste projeto, utilizou-se a linguagem de programação C++ e a biblioteca gráfica OpenGL.

Capítulo 2

Estrutura do código

De modo a estruturar o projeto foram utilizadas diferentes diretorias: uma com informação relativa ao Generator e outra em relação ao Engine.

2.1 Generator

Nesta diretoria encontra-se todo o código associado à geração das coordenadas dos vértices, para a representação de vários modelos.

Após estas terem sido geradas, foram armazenadas num ficheiro .3d na diretoria onde se encontra o executável, caso não lhes tivesse sido atribuído um caminho. Neste ficheiro, a primeira linha contém o número total de floats. Nas linhas seguintes, por sua vez, encontram-se 3 floats, que corresponderam a 1 vértice. Estes aparecem por ordem x,y,z.

2.2 Engine

Esta diretoria tem como principal objetivo a leitura dos ficheiros XML e a visualização da representação gráfica proveniente dos ficheiros .3d (resultantes do Generator). Para efetuar o parsing do XML recorreu-se à biblioteca TinyXML2.

Capítulo 3

Primitivas

Este projeto tem como objetivo a implementação das diferentes primitivas: plano, caixa, esfera e cone. De modo a gerar as diferentes primitivas, é necessário fornecer diferentes comandos ao Generator, tais como:

- **sphere** radius slice stacks file
- **cone** radius height stacks slice file
- **box** length divs file
- **plane** length file

3.1 Plano

Para a criação de um plano são utilizados os seguintes parâmetros:

- **length:** comprimento do lado do plano
- **divisions:** números de divisões a serem feitas em cada eixo

Como se trata de um plano em XZ, todos os pontos terão a coordenada y a 0.

Para este plano ser construído, utilizamos dois ciclos de forma a encontrar os vértices dos triângulos. Começamos pelo lado negativo do eixo X e Z, canto superior esquerdo do plano XZ, afastado metade do comprimento do plano, de modo a que a figura ficasse centrada na origem. Iteramos até ao lado positivo do eixo do X, encontrando os pontos dos vértices dos dois triângulos que constituem cada quadrado da divisão.

Ao acabar uma divisão, aumentamos o valor Z, encontrando assim o início da seguinte.

3.2 Box

Os parâmetros da caixa são:

- **length:** comprimento do lado
- **division:** números de divisões a serem feitas

A Box é nada mais do que seis planos centrados em cada eixo.

Utilizamos a mesma técnica que a do plano anterior, com a diferença de que em cada face iteramos pontos em eixos diferentes. Encontramos a origem de cada face adicionando ou subtraindo metade do comprimento do lado da caixa. Em cada eixo a sua respectiva coordenada fica "trancada" e apenas iteramos os outros dois eixos. As faces viradas para baixo e para trás fazem triângulos com a ordem invertida, sendo contra os ponteiros do relógio, de modo a estarem viradas contra a câmara.

3.3 Sphere

Parâmetros da esfera:

- **radius:** comprimento do raio
- **slices:** número de fatias horizontais
- **stacks:** número de fatias verticais

A Esfera segue o mesmo padrão para encontrar os triângulos, com a exceção de que utilizamos coordenadas esféricas para mapear os vértices.

Cada iteração desloca, na superfície de uma esfera, um ponto através de um ângulo alfa, eixo do X, e um ângulo beta, eixo do Y.

Quando encontramos um triângulo na parte de cima conseguimos automaticamente encontrar o seu simétrico na parte de baixo, invertendo o sinal do ângulo beta.

3.4 Cone

Parâmetros da esfera:

- **radius:** comprimento do raio
- **hight:** altura do cone
- **slices:** número de fatias horizontais
- **stacks:** número de fatias verticais

O Cone utiliza coordenadas polares.

Começamos por desenhar uma circunferência na origem e, a cada passo, diminuimos o seu raio e altura, construindo-o assim por camadas. Para encontrar os vértices, tivemos a necessidade de calcular o raio e a altura da camada seguinte.

Todas as primitivas seguem o mesmo esquema para encontrar os pontos dos triângulos. No primeiro, começamos pelo canto inferior esquerdo, seguido pelo inferior direito e terminamos no canto superior esquerdo. No seguinte, de forma a completar o quadrado, repetimos o canto inferior direito, seguindo pelo canto superior esquerdo e acabando com a repetição do canto superior esquerdo.

Capítulo 4

Conclusão

Com o desenvolvimento da primeira fase do projeto, foi-nos possível consolidar os conhecimentos adquiridos nas aulas, em particular, a produção de gráficos 3D, através da renderização de primitivas gráficas.

Ao longo desta fase, enfrentamos alguns obstáculos, sobretudo no cálculo dos pontos das primitivas. O primeiro grande desafio foi encontrado na navegação num plano 3D, quando foi necessário encontrar triângulos na orientação certa. Para além disso, sentimos dificuldades na utilização de coordenadas polares e esféricas, que vieram trazer mais entropia à linha de raciocínio. Por fim, encontramos um obstáculos nos erros de arredondamento, devido à má utilização de operações matemáticas de variáveis inteiras com virgulas flutuantes. No entanto, acreditamos que essas dificuldades foram ultrapassadas.

Esta fase permitiu-nos ainda expandir o conhecimento da linguagem C++, da API OpenGL e dos ficheiros do tipo XML.

Em suma, consideramos que foi elaborado um trabalho bastante positivo e que, apesar das dificuldades encontradas, foram superados muitos desafios, tendo-se concluído, dessa forma, todos os requisitos com sucesso.