Licenciatura em Engenharia Informática
Universidade do Minho

Computação Gráfica

1ª Fase – Primitivas Gráficas Simples

André Geraldes 67673

Patrícia Barros 67665

Sandra Ferreira 67709

Conteúdo

Índice de Figuras

Introdução

Este trabalho foi desenvolvido no âmbito da Unidade Curricular de Computação Gráfica, pertencente ao plano de estudos do 3º ano da licenciatura em Engenharia Informática.

Este projeto será constituído por 4 fases distintas com o objetivo final de criar um *motor 3D*. Nesta primeira fase foi-nos proposto o desenvolvimento de uma aplicação que crie um ficheiro onde irão estar armazenados os triângulos necessários ao desenho de primitivas de sólidos geométricos e também o desenvolvimento de um pequeno motor que lê de um ficheiro XML os triângulos previamente gerados e constrói a respetiva primitiva.

Desenvolvimento

Gerador

A primeira etapa do desenvolvimento do nosso projeto foi a aplicação "gerador" que recebendo o nome do sólido a desenhar e alguns argumentos que o caracterizam (variáveis consoante o sólido) gera os pontos necessários para o seu desenho através de triângulos. Esses pontos são guardados num ficheiro ordenados pela ordem correta para os triângulos serem desenhados como pretendido.

Implementação do Plano

Para a implementação do plano o gerador recebe 4 argumentos: o comprimento, a largura, o número de camadas horizontais e o número de camadas verticais.

O plano é desenhado a partir do seu canto superior esquerdo, cujas coordenadas são facilmente determinadas através do seu comprimento e largura. A partir desse ponto é possível descobrir todos os outros da seguinte forma: o ponto seguinte terá um deslocamento no eixo do x em relação ao anterior, sendo esse deslocamento dado pela largura dividida pelo número de camadas horizontais. O ponto seguinte terá um deslocamento no eixo do z em relação ao inicial. Da mesma forma esse deslocamento é dado pelo comprimento dividido pelo número de camadas verticais. Assim temos já o primeiro triângulo formado. O segundo terá estes últimos dois pontos e ainda um outro que é dado pelo ponto inicial com um deslocamento tanto vertical como horizontal calculado da mesma forma que os anteriores. Temos assim dois triângulos formados. A partir daí descobrem-se todos os outros iterando este algoritmo por toda a largura e comprimento do plano. Segue-se um excerto de código que demonstra como foi realizado o cálculo do ponto inicial e dos deslocamentos (saltos):

Resultado

Segue-se um exemplo de um plano gerado desta forma com comprimento 4, largura 2, 5 camadas horizontais e 5 camadas verticais.

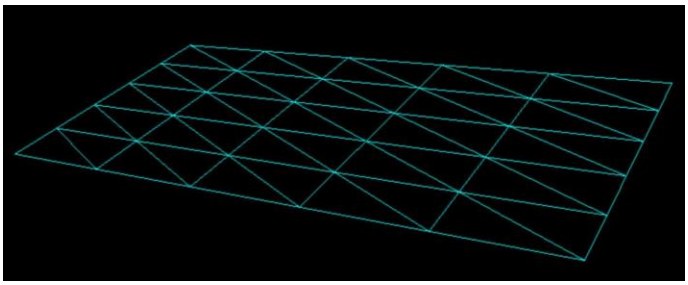

Figura 1 - Representação do plano

Implementação do Paralelepípedo

Para a implementação do paralelepípedo o gerador recebe 6 argumentos: comprimento, largura, altura, número de camadas verticais, número de camadas horizontais e número de camadas longitudinais.

As bases do paralelepípedo são geradas da mesma forma que o plano, apenas com uma translação no eixo do y de tantas unidades quanta a altura do sólido a dividir por dois (para ficar centrado), uma translação no sentido negativo e outra no sentido positivo.

Para os planos traseiro e frontal utilizamos um deslocamento no eixo do y que corresponde à razão entre a altura e as camadas horizontais e o deslocamento no eixo do z que já tínhamos definido anteriormente para as bases. Os pontos foram calculados da mesma forma que os das bases mas utilizando estes deslocamentos. Nestes pontos o valor de x nunca se altera.

Nos pontos das bases laterais é o valor de z que nunca se altera e como explicado anteriormente são utilizados os deslocamentos no eixo do x e y para descobrir as coordenadas de todos os pontos.

Resultado

Segue-se um exemplo de um paralelepípedo gerado desta forma com comprimento 4, largura 3, altura 2, 10 camadas verticais, 10 camadas horizontais e 10 camadas longitudinais.

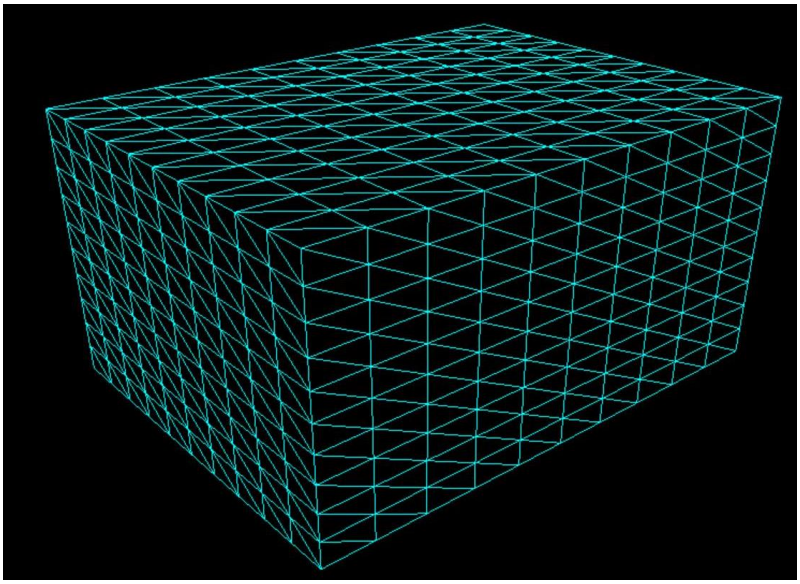

Figura 2 - Representação do paralelepípedo

Implementação do Cone

Para o desenho do cone são utilizados 4 argumentos: o raio, a altura, as camadas verticais e as camadas horizontais.

Para desenhar a base iteramos sobre o número de camadas verticais, utilizando a razão entre $2 * \pi$ e o número de camadas verticais para incrementar um ângulo α, desde que este toma o valor de 0 até completar o círculo. A base foi desenhada com um valor de y correspondente a – $(altura\ do\ cone/2)$ para que o cone ficasse centrado em relação à origem. Os pontos foram descobertos utilizando o sistema de coordenadas polares.

Para a lateral do cone utilizamos dois ciclos. O primeiro itera sobre as camadas horizontais e o segundo, à semelhança do utilizado para a construção da base, itera sobre as camadas verticais. Através do cálculo da razão entre o raio e as camadas horizontais foi possível descobrir quanto é que o raio iria diminuir em cada iteração e utilizar essa informação para as coordenadas dos pontos. Através da razão entre a altura do cone e o número de camadas horizontais foi possível determinar a altura a que iriam estar os pontos da camada seguinte do cone. Desta forma, diminuindo o raio a cada iteração e aumentando o valor do y foi possível construir o cone com sucesso.

Resultado

Segue-se um exemplo de um cone gerado desta forma com raio 1, altura 5, 30 camadas verticais e 30 camadas horizontais.

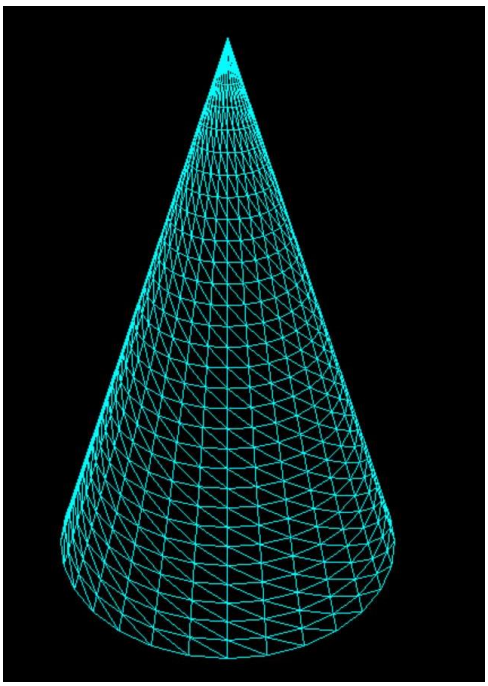

Figura 3 - Representação do cone

Implementação da Esfera

Para a implementação da esfera o gerador recebe três parâmetros: o raio, o número de camadas horizontais e o número de camadas verticais. Para desenhar esta primitiva utilizaram-se coordenadas esféricas, coordenadas estas que trabalham com dois ângulos, o θ cujo domínio pertence ao intervalo [0,2π] e o ângulo φ, pertencente ao intervalo [0,π]. Foram definidos quatro pontos, pontos esses onde os ângulos θ e φ variam de acordo com o salto, sendo esse salto a razão entre 2* π e o numero de camadas horizontais, no diz respeito ao ângulo θ e a razão entre π e o número de camadas verticais no que diz respeito ao ângulo φ.

Resultado

Segue-se um exemplo de uma esfera gerada desta forma com raio 1, 30 camadas verticais e 30 camadas horizontais.

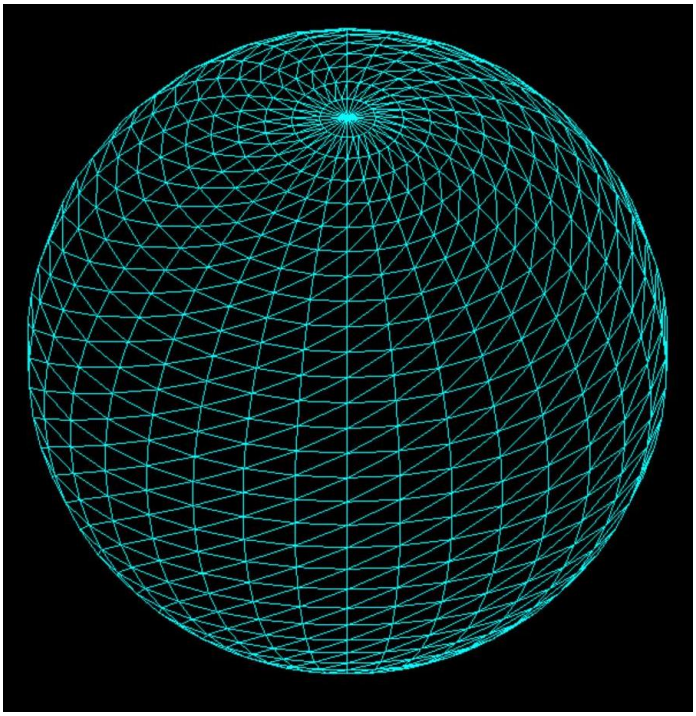

Figura 4 - Representação da esfera

Implementação do Cilindro

A implementação do cilindro foi feita de forma semelhante ao cone, com a exceção de que o raio não sofre nenhuma diminuição ao longo das iterações e em vez de uma só base tem duas bases simétricas em relação ao eixo do y.

Resultado

Segue-se um exemplo de um cilindro gerado desta forma com raio 1, altura 3, 30 camadas verticais e 30 camadas horizontais.

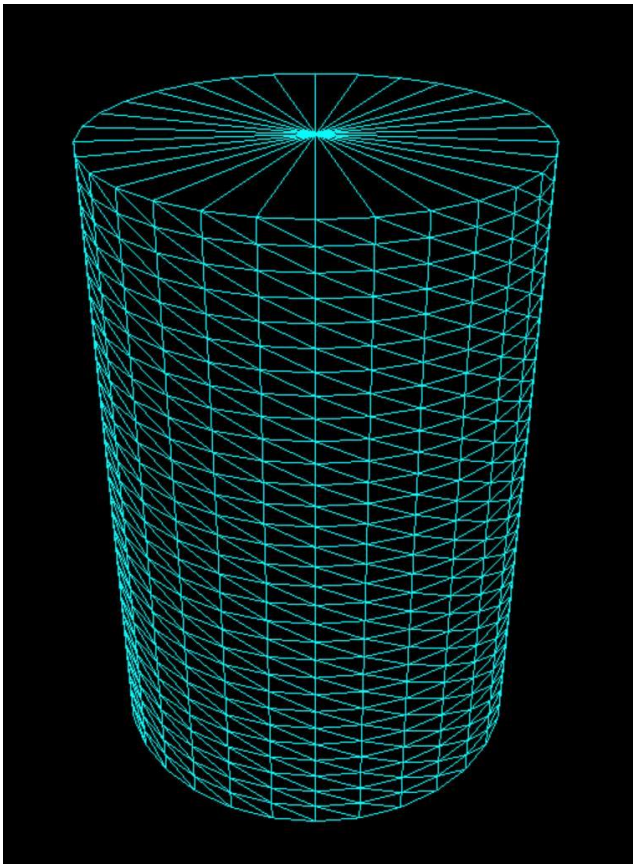

Figura 5 - Representação do cilindro

Motor 3D

O motor 3D foi criado para ler de um ficheiro XML com a informação com a informação dos modelos a desenhar e através dele desenhar o sólido correspondente.

Para o desenvolvimento do motor baseamo-nos no esqueleto que nos foi fornecido para utilização nas aulas práticas.

Acrescentamos rotações da câmara através das teclas especiais e do rato e também translações do sólido através das teclas normais para melhor visualização do trabalho realizado. Temos ainda disponível um menu quando é efetuado um clique no botão direito do rato que permite mudar o modo de apresentação dos sólidos (linhas, sólido ou pontos). Nesta parte existe uma limitação: a rotação através do rato só funciona como é suposto se ainda não tiver sido pressionada nenhuma das teclas normais (QWEASD) pois depois de terem sido pressionadas alteram os valores das variáveis que também são utilizadas para efetuar a rotação com o rato que deixa de funcionar como suposto.

Para guardar os pontos lidos foi criada uma estrutura **Ponto** constituída por três *doubles* para guardar cada valor das coordenadas e um vetor **Pontos** para guardar vários elementos da estrutura anterior. Foi feita uma função ***readFile*** para fazer a leitura e *parsing* do ficheiro e guardar os pontos lidos nas estruturas mencionadas anteriormente.

O desenho das primitivas é feito iterando sobre o vetor **Pontos** e através do **GL_TRIANGLES** e do **glVertex3f** são desenhados todos os triângulos cujos pontos estavam guardados no ficheiro lido. Os pontos gravados no ficheiro já se encontram devidamente ordenados de forma a desenhar o sólido pretendido logo isso não é algo com que tivéssemos que nos preocupar nesta fase da construção do motor.

Conclusão

A realização desta primeira fase do projeto foi muito útil para a consolidação dos conhecimentos previamente lecionados nas aulas teóricas e práticas da Unidade Curricular de Computação Gráfica.

Aprendemos a lidar melhor com as rotações da câmara através do teclado e também com o rato, tendo só tido uma pequena limitação que pretendemos corrigir numa fase futura.

Enfrentamos algumas dificuldades principalmente no que toca a visualizar os pontos no espaço através de coordenadas polares e esféricas e a perceber como as mesmas funcionam, contudo conseguimos superar esses obstáculos e realizar todas as formas sugeridas e ainda uma adicional: o cilindro.

No geral fazemos um balanço bastante positivo do trabalho conseguido.