

Computação Gráfica
Fase 4
Relatório de Desenvolvimento

Bárbara Faria
(A85774)

João Marques
(A84684)

José Pires
(A84552)

Tiago Lima
(A85126)

5 de junho de 2022

Resumo

No âmbito da UC de Computação Gráfica foi-nos proposto a realização de um trabalho prático. Este foi dividido em 4 partes, sendo esta a quarta e última fase. Nesta fase foi pedido que o gerador da aplicação fosse capaz de gerar coordenadas de textura e normais ao vertice. O nosso engine terá de ser capaz, de ativar as funcionalidades de luz e textura, assim como ler e aplicar as coordenadas das normais e texturas dos modelos.

Conteúdo

1	Gerador	2
1.1	Plano	2
1.2	Caixa	3
1.3	Esfera	3
1.4	Cone	4
1.5	Bezier	5
2	Motor Gráfico	6
2.1	Iluminação	6
2.1.1	Luz	6
2.1.2	Cor	7
3	Exemplos dos professores	8
4	Sistema Solar	12
5	Conclusão	13

Capítulo 1

Gerador

Uma normal é um vetor unitário cuja direção é perpendicular a uma superfície num determinado ponto. Neste capítulo será explicado como foi feita a implementação do cálculo das normais dos vértices de cada modelo.

1.1 Plano

A geração das normais em relação ao plano é feita de uma forma muito simples, como neste caso, se trata de um plano XZ, as normais dos seus vértices são sempre as mesmas, sendo $(0,1,0)$ a normal dos pontos da face voltada para cima, e $(0,-1,0)$ para os pontos da face voltada para baixo. Como é demonstrado na figura 1.1.

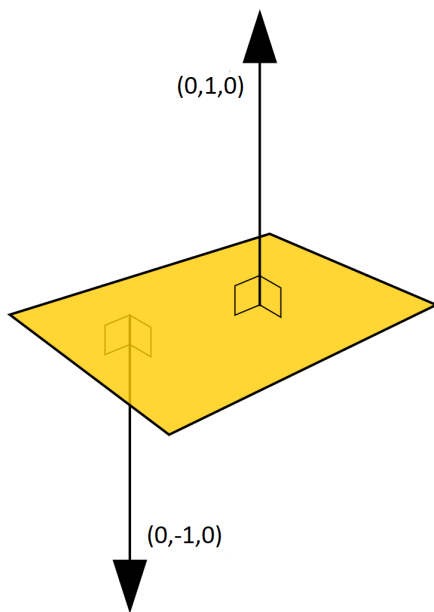


Figura 1.1: Normais do plano

1.2 Caixa

A geração das normais da caixa faz-se de uma maneira muito similar ao plano, uma vez que temos seis faces, sendo que duas são paralelas ao plano XZ, então mais uma vez as suas normais serão $(0,1,0)$ para a face voltada para cima e $(0,-1,0)$ para a face voltada para baixo; temos também duas faces paralelas ao eixo XY, então da mesma forma as normais serão $(0,0,1)$ para a face voltada para cima e $(0,0,-1)$ para a face voltada para baixo; as ultimas duas faces serão paralelas ao plano YZ e as suas normais serão, $(1,0,0)$ para a face voltada para cima e $(-1,0,0)$ para a face voltada para baixo.

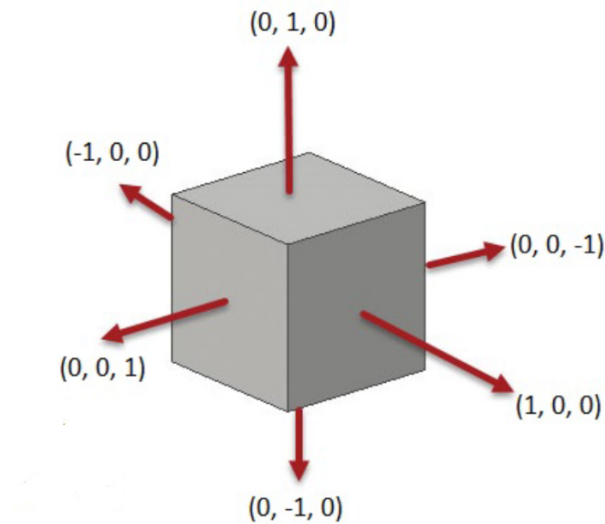


Figura 1.2: Normais do cubo

1.3 Esfera

Relativamente à esfera, as suas normais serão calculadas através da normalização dos componentes x, y e z de cada vertice, isto é, dividindo o seu valor pelo raio, ora então temos que a normal de qualquer vertice da superfície é dada por $\cos(\beta) \times \sin(\alpha), \sin(\beta), \cos(\beta) \times \cos(\alpha)$.

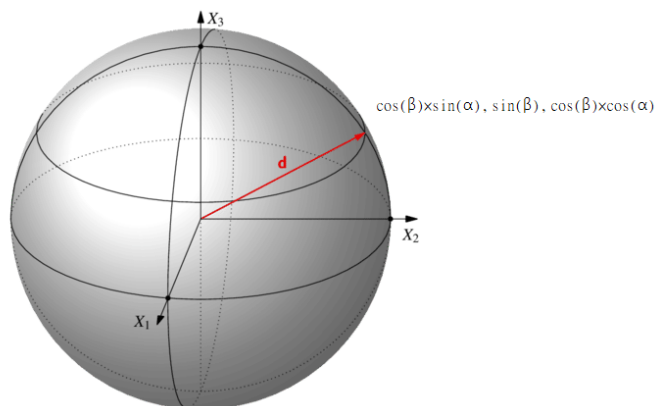


Figura 1.3: Normal da esfera

1.4 Cone

A normal correspondente a todos os vértices da base do cone será $(0,-1,0)$. Por outro lado, a normal dos vértices pertencente à superfície será dada pela fórmula,

$$\cos(ac) \times \sin(a), \sin(ac), \cos(ac) \times \cos(a)$$

, onde a será o ângulo da “slice” do vértice e ac será o ângulo do cone, este será calculado convertendo para radianos a divisão da altura do cone pelo raio da base, como ilustrado na figura 1.4

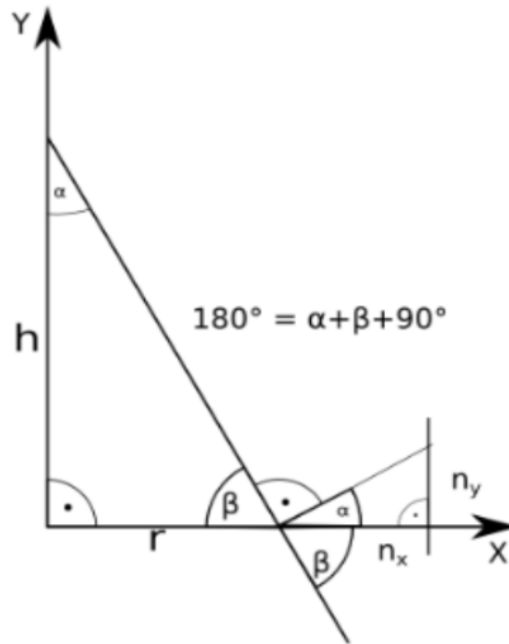


Figura 1.4: Normal da superfície do cone

1.5 Bezier

No que se trata de Patches de Bezier as normais de cada ponto são calculadas, pela normalização do produto externo entre os vetores tangentes a esse ponto. Consideremos um ponto A de coordenadas u e v então, sejam

$$U = [u^3, u^2, u, 1]$$

$$V = [v^3, v^2, v, 1]$$

calculámos as derivadas em u e v da seguinte maneira

$$\text{deri}U = [3u^2, 2u, 1, 0] \times M \times P \times M^T \times V^T$$

$$\text{deri}V = U \times M \times P \times M^T \times [3v^2, 2v, 1, 0]$$

Onde M representa a matriz de Bezier e P os dezasseis pontos de controlo

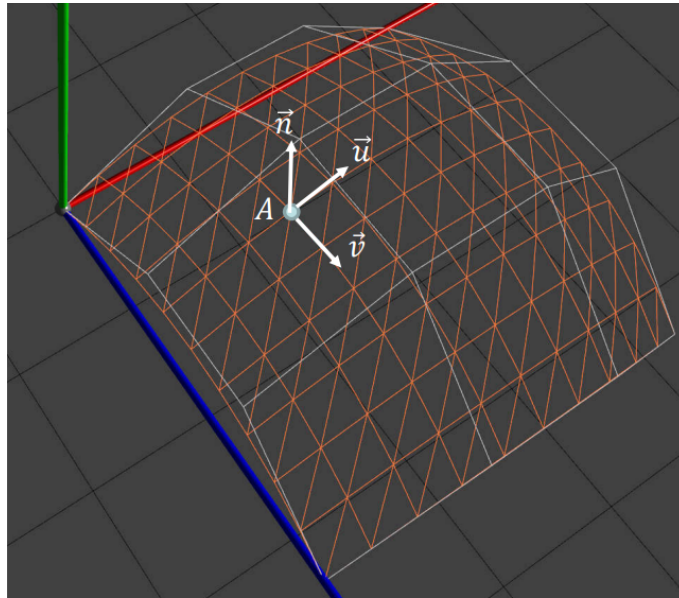


Figura 1.5: Normal de Bezier

Capítulo 2

Motor Gráfico

Comparando com a última fase, o nosso motor gráfico está agora preparado para desenhar as sombras das nossas primitivas. A sombra de uma primitiva é desenhada automaticamente a partir das suas normais, mas para tal, será necessário que exista pelo menos uma fonte de luz de forma a visualizarem-se as sombras.

2.1 Iluminação

2.1.1 Luz

Uma luz pode ser um de três tipos: point, directional e spotlight. Quando uma luz é do tipo point, significa que os raios de luz são emitidos em todas as direções a partir de um único ponto, uma do tipo directional significa que os raios de luz serão todos paralelos uns aos outros e se caso seja do tipo spotlight, os raios de luz serão emitidos a partir de um ponto, mas, em vez de irradiarem em todas as direções como no point, estão restritos a uma forma tipo cone.

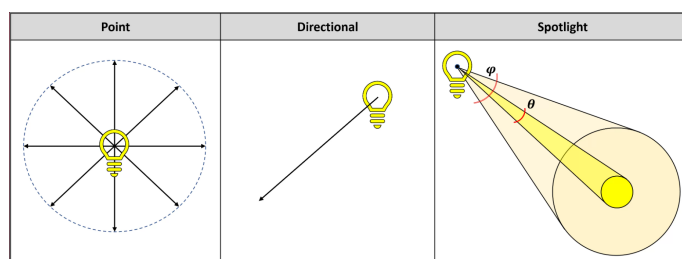


Figura 2.1: Tipos de luz

Após a leitura do ficheiro XML o nosso motor gráfico irá se encarregar de guardar o tipo e as coordenadas da luz numa classe que denominamos Luz. Para conseguirmos fazer uso de múltiplas luzes na mesma cena, mantemos armazenado globalmente o valor do número de luzes a ativar (incrementado a cada instância criada da classe Luz), então iterando sobre esse valor e fazendo uso da função, `glLightfv` do OpenGL, ativamos `Light0`, `Light1`... (máximo 7).

2.1.2 Cor

As cores de cada modelo têm 5 componentes diferentes: diffuse, ambient, specular, emissive e o shininess. Na reflexão difusa, a intensidade recebida é refletida uniformemente em todas as direções. A iluminação ambiente simula as interações entre os objetos e a luz, iluminando todos por igual. A reflexão especular verifica, ao iluminar materiais brilhantes, uma mancha mais clara cuja posição depende da posição do observador, sendo que o coeficiente da especularidade determina a dimensão da mancha. Semelhante ao que foi feito anteriormente é criada uma classe Cor onde o nosso engine vai-se encarregar de guardar os valores lidos do XML e fazendo uso da função, `glMaterialfv` do OpenGL, atribuímos os materiais respetivamente.

Capítulo 3

Exemplos dos professores

Este capítulo resume-se a demonstrar um modelo do sistema solar, assim como, os exemplos fornecidos pelos professores, gerados pelo nosso programa

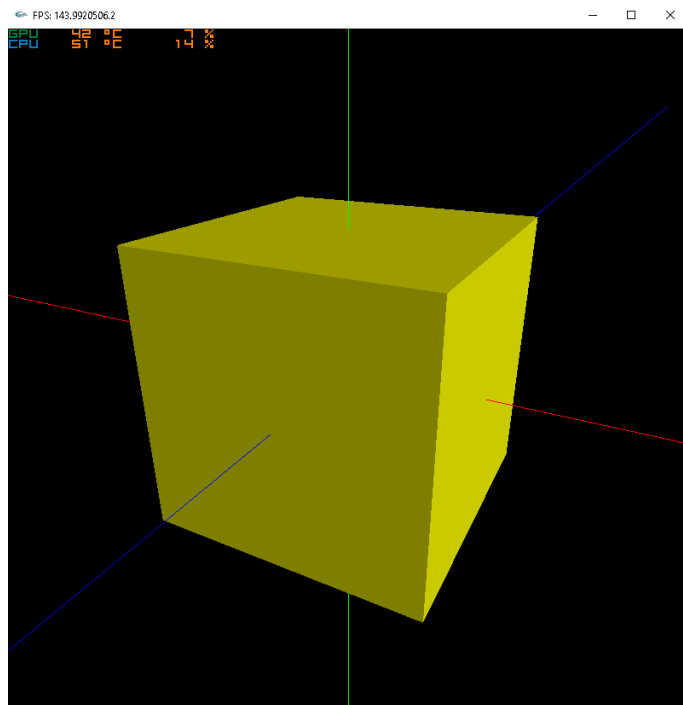


Figura 3.1: teste 1

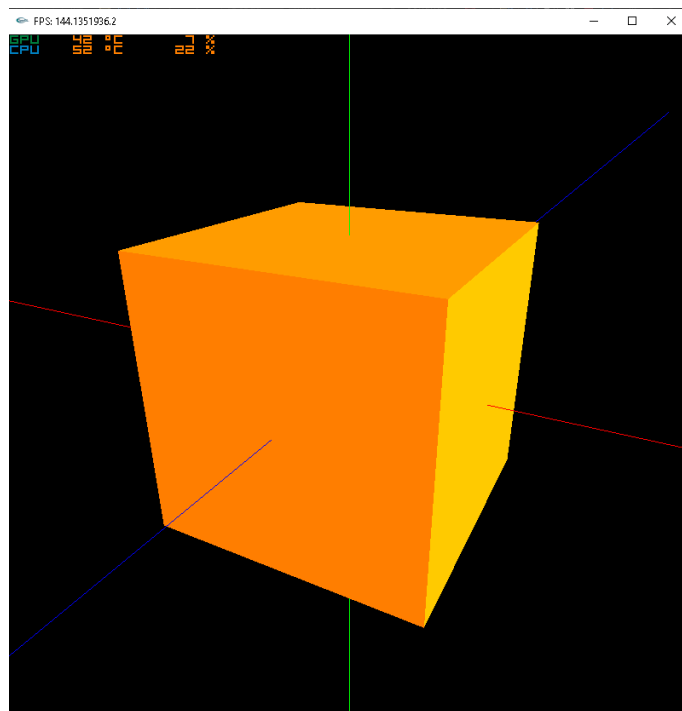


Figura 3.2: teste 2

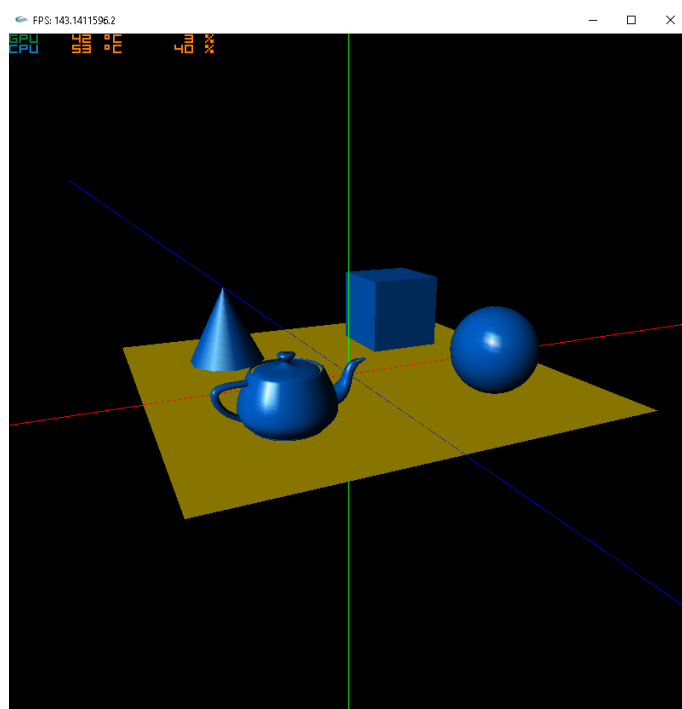


Figura 3.3: teste 3

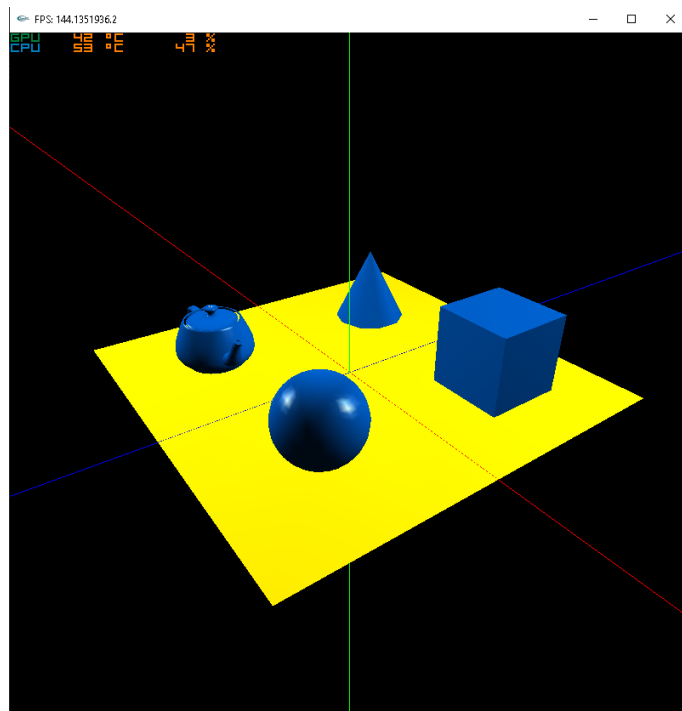


Figura 3.4: teste 4

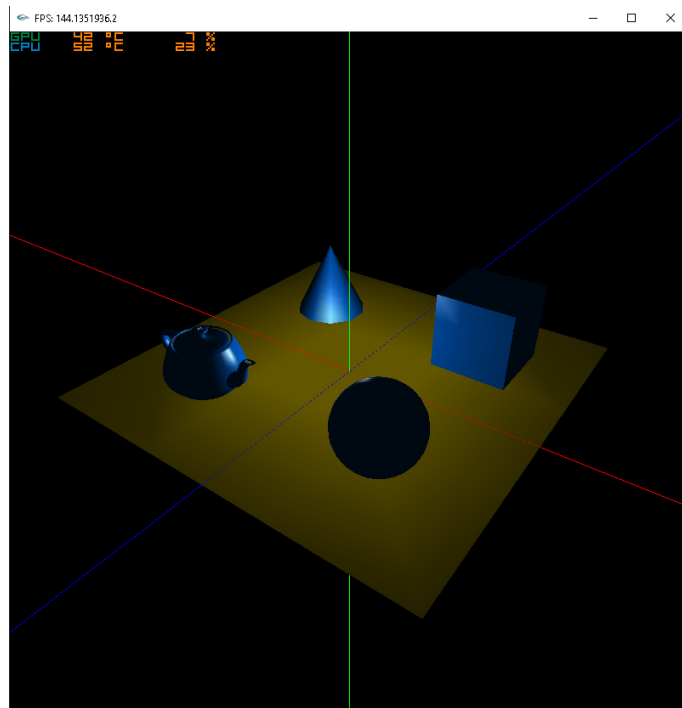


Figura 3.5: teste 5

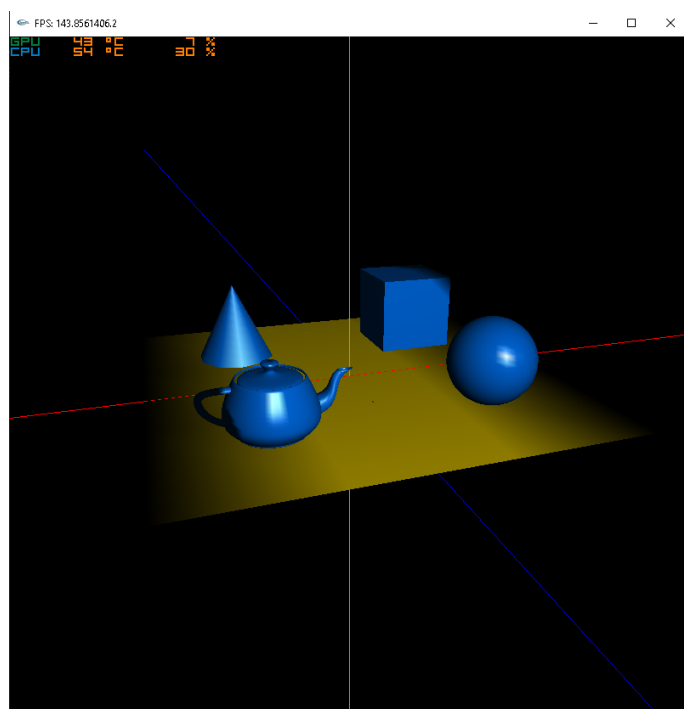


Figura 3.6: teste 6

Capítulo 4

Sistema Solar

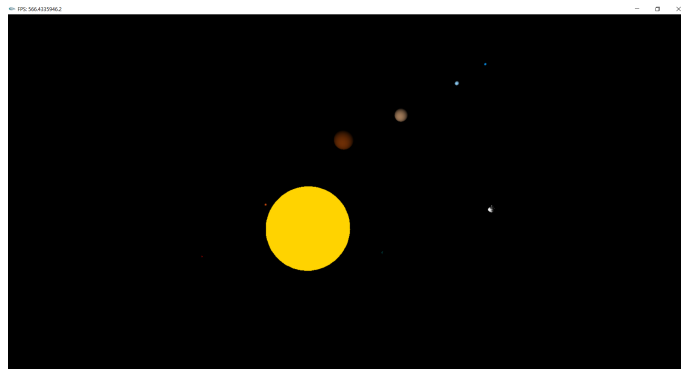


Figura 4.1: Modelo iluminado do Sistema Solar

Capítulo 5

Conclusão

Nesta última fase, conseguimos aplicar com sucesso as funcionalidades de iluminação à nossa aplicação. O nosso generator é capaz de criar normais, e a nossa engine ativar essas mesmas definições a partir da leitura dos model files. Informações sobre luzes difusa, especular, emissiva e ambiente são capazes de ser definidas através do ficheiro XML, bem como o brilho e as fontes de onde a luz é proveniente.

Por motivos de logística, não nos foi possível o desenvolvimento da parte relativa às texturas. Tendo esta entrega coincidido com uma altura crítica do semestre, o grupo não teve oportunidade de explorar essa parte do trabalho, tendo ficado aquém dos objetivos a serem cumpridos.