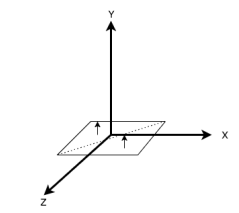
# Normais e Texturas

Nesta fase e por forma a conseguirmos uma ilustração mais real do Sistema Solar decidimos aplicar normais, importantes para a iluminação, e texturas.

### Plano

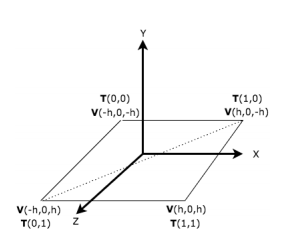
As normais do plano são bastante simples, visto que apenas têm de apontar para “cima” consoante o plano no qual são desenhados, ou seja, são todas iguais e como são desenhados no plano xOz o eixo dos y corresponde ao eixo das “alturas”, logo as normais correspondem ao vetor (0,1,0).



Relativamente aos pontos de textura do plano, estes também são bastante simples, visto que apenas temos de fazer corresponder cada vértice do plano ao vértice do espaço das texturas.

A correspondência é a seguinte:

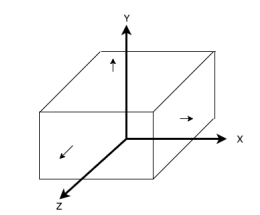
* (-l/2,0,-l/2) -> (-1,-1)
* (-l/2,0,l/2) -> (-1,1)
* (l/2,0,-l/2) -> (1,-1)
* (l/2,0,l/2) -> (1,1)



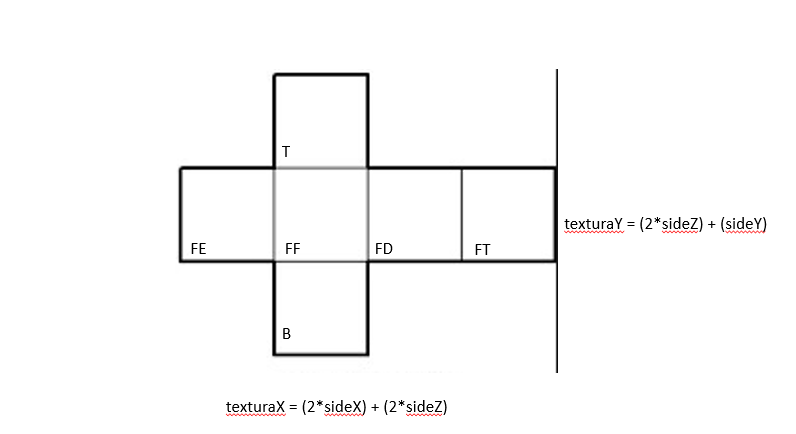
### Caixa

Relativamente às normais da caixa, estas seguem o mesma raciocínio que o plano, no entanto nesta figura temos de considerar que cada face corresponde a um plano diferente, pelo que as normais serão as mesmas na mesma face, mas diferentes para as diferentes faces. Aqui a ideia é que a normal aponte no sentido de fora da caixa, pelo que serão as seguintes:

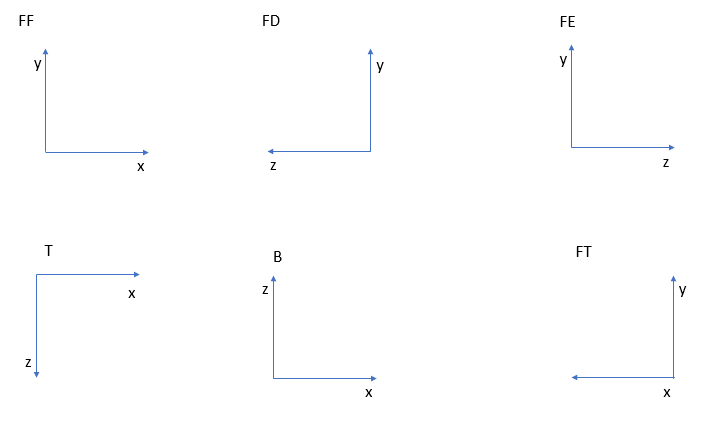
* Face Frontal - vetor (0,0,1)
* Face Traseira - vetor (0,0,-1)
* Face Direita - vetor (1,0,0)
* Face Esquerda - vetor (-1,0,0)
* Topo - vetor (0,0,1)
* Base - vetor (0,0,-1)



Para calcular os pontos de textura desta figura tivemos de analisar uma imagem 2D (correspondente ao espaço de texturas) com a construção da mesma:

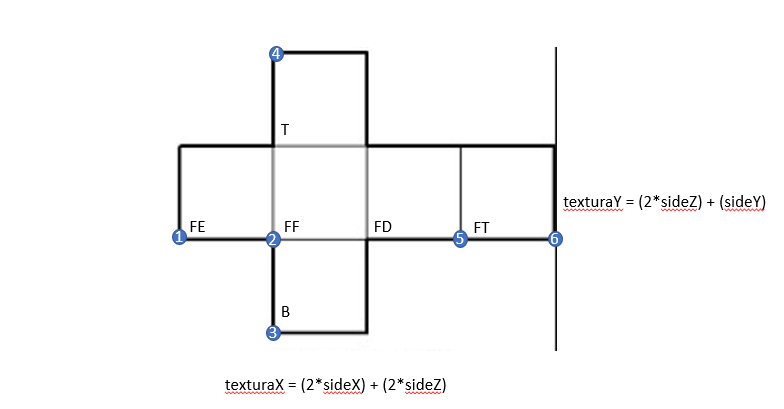


Com a análise da planificação do cubo chegamos aos seguintes resultados para o desenho de cada face e os eixos respetivos:



Como podemos ver nas imagens acima descritas, o eixo x das texturas corresponde ao valor apresentado e o eixo y também, isto apenas teve de ser realizado olhando para os eixos das faces e somando os respetivos eixos.

Por exemplo o eixo y é correspondente ao eixo “y” das faces frontal, do topo e da base, ou seja: y+z+z = y + 2\*z. O valor é o sideX(Y ou Z) pois é o valor passado para as dimensões do cubo, para que depois se posa fazer corresponder cada ponto na coordenada respetiva.



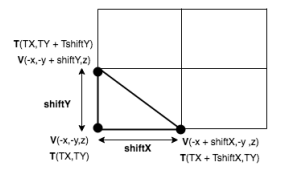
Para iniciar o processo de conversão para coordenadas de textura tivemos de analisar os pontos iniciais para cada face, apresentamos esses pontos em cima e as suas coordenadas de seguida:

1. (0 , sideZ/texturaY)
2. (sideZ/texturaX , sideZ/texturaY)
3. (sideZ/texturaX , 0)
4. (sideZ/texturaX , 1)
5. ((2\*sideZ+sideX)/texturaX , sideZ/texturaY)
6. (1 , sideZ/texturaY)

Após esta análise tivemos de pensar como seriam feitas as iterações, ou seja qual seria o valor da iteração para cada face, chagamos aos seguintes resultados:

* deltaXText = (sideX/texturaX)/ divisions -> para todas as faces envolventes ao X, visto que este apenas varia no sentido do eixo dos x da coordenadas de textura
* deltaYText = (sideY/texturaY)/ divisions -> para todas as faces envolventes ao Y, visto que este apenas varia no sentido do eixo dos y da coordenadas de textura
* deltaZText1 = (sideZ/texturaX)/ divisions -> para todas as faces envolventes ao Z que variam no eixo dos x das coordenadas de textura
* deltaZText2 = (sideZ/texturaY)/divisions -> para todas as faces envolventes ao Z que variam no eixo dos y das coordenadas de textura

De seguida apresentamos um exemplo para a face da frente, as outras são análogas (variando apenas no sentido positivo ou negativo como apresentado anteriormente):



### Esfera

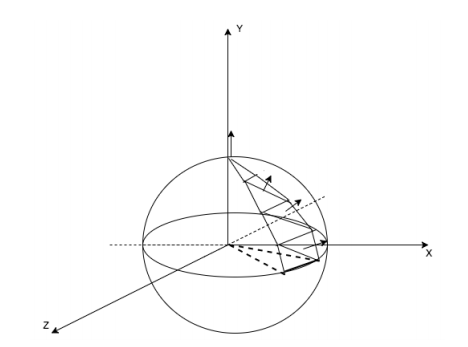
Para calcular a normal da esfera apenas temos de verificar que esta tem de apontar no sentido oposto ao sentido do centro da mesma, logo o vetor correspondente à normal da mesma será:

(sin(beta \* j) \* sin(alfa\*i), cos(beta \* j) , sin(beta \* j) \* cos(alfa\*i))

Na iteração (i,j) e com:

* alfa = 2\*pi/fatias
* beta = pi / camadas.

Ou seja é muito parecido às coordenadas do ponto, apenas terá de ser normalizado (divido pelo raio).



Para calcular os pontos de textura da esfera inicialmente tivemos algumas dificuldades, no entanto depois percebemos que o ponto de textura apenas corresponderia à iteração em que estávamos, seria então:

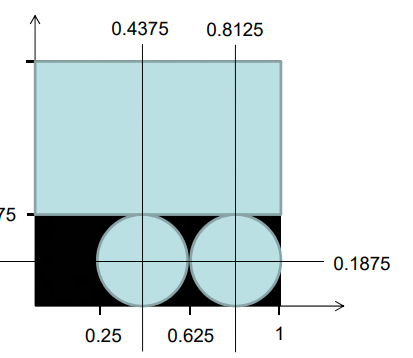
(i/fatias , 1- (j/camadas) ).

### Cilindro

O cálculo das normais para o cilindro pode ser feito por 3 partes distintas: a base, o topo e o corpo lateral. Na base e no topo é parecido à base e topo da caixa, na lateral apenas temos de garantir que o vetor normal “aponta na direção oposta ao cilindro”. De seguida apresentam-se os respetivos vetores:

* Topo - (0,0,1)
* Base - (0,0,-1)
* Corpo Lateral - (sin(alfa\*i), 0, cos(alfa\*i)) - sendo alfa = 2\*pi/fatias e i a iteração correspondente.

Para mapear os pontos para o espaço de textura utilizamos a construção do cilindro como mostramos de seguida:



A primeira esfera corresponde ao topo, pelo que a coordenada de textura correspondente no topo será:

(0.4375+0.1875\*cos(alfa\*i), 0.1875+0.1875\*sin(alfa\*i))

A segunda esfera corresponde à base, pelo que a sua coordenada de textura correspondente será:

(0.8125+0.1875\*cos(alfa\*i), 0.1875+0.1875\*sin(alfa\*i))

Relativamente ao corpo lateral corresponde à parte de cima, para este apenas temos de verificar que:

* a coordenada x de textura será a iteração atual a dividir pelo número de iterações
* a coordenada y será o valor da altura atual a dividir pela altura total multiplicado pelo delta da altura no espaço de texturas (0.625) e somando 0.375

ou seja:

(i/fatias , (i\*deltaAltura/altura)\*0.625+0.375)

Onde deltaAltura = altura / camadas (no entanto nós apenas consideramos 2 camadas).

### Torus

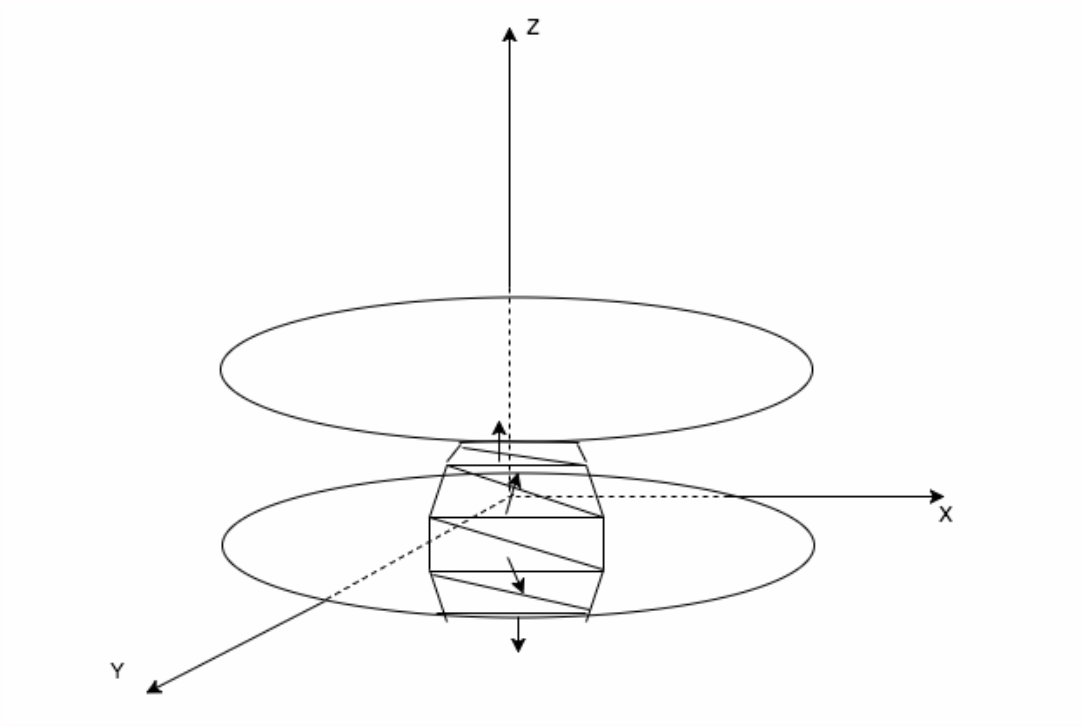
Para obter os vetores normais do torus foi preciso observar o processo de construção do mesmo uma vez que dá-nos as orientações dos vetores normais de cada vértice no momento do desenho deste. Para ober os vetores das normais utilizamos então a seguinte formula:

Normal( cos(A)\*cos(L), sin(A)\*cos(L), sin(L) ) - A

onde,

A – desvio do anel

L – desvio de cada lado que forma um anel



Para as coordenadas de texturas utilizamos um mapeamento simples. Dividimos a imagem da nossa textura em varias tiras correspondentes ao numero de anéis do torus. Posteriormente dividimos cada tira pelo numero de camadas do torus e atribuimos cada rectangulo resultante à camada correspondente.

A formula utilizada foi a seguinte:

Textura( i/slice , j/sides)

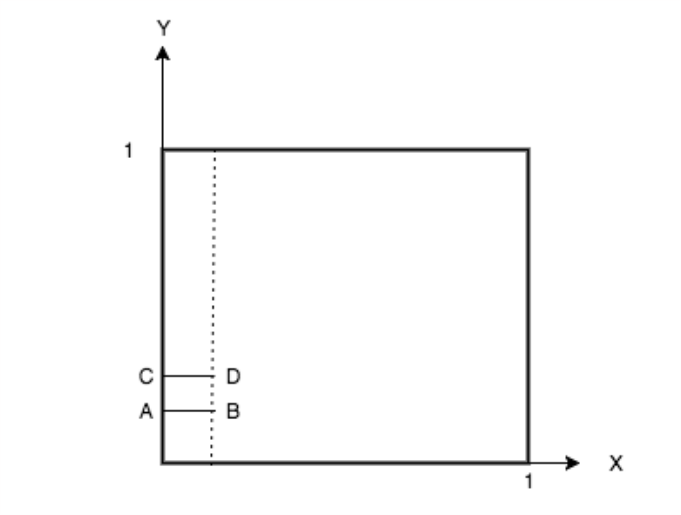
onde,

i – corresponde ao número do anel

j – corresponde ao número do lado

slice – número total de anéis

sides – número total de lados



### Cilindro

Assim como no cilindro dividimos o cone em 2 partes para o calculo das normais.

Assim sendo:

**Base:** vetor (0,0,-1)(assim como a base do cilindro);

**Corpo:** vetor (sin(alfa),L/camada, cos(alfa))-L