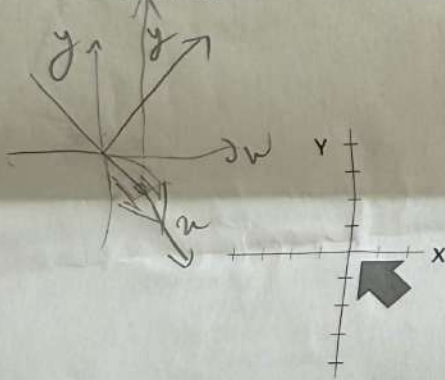
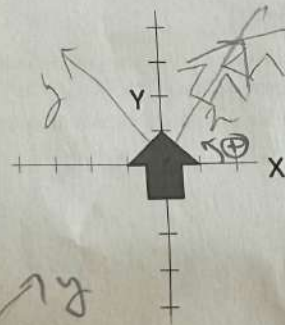


### Parte I - Transformações Geométricas, Câmara

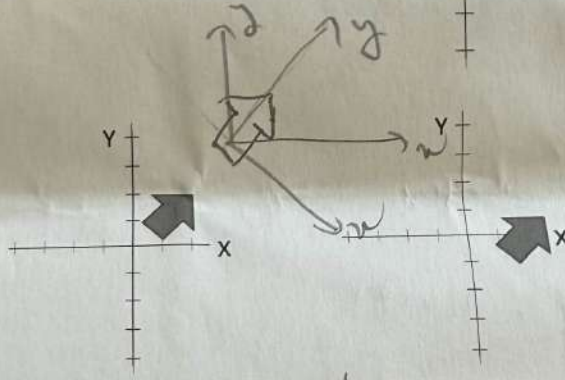
- ✓ Considere uma primitiva gráfica para desenhar um objecto com centro na origem, e a seguinte sequência de transformações geométricas a aplicar ao objecto:

```
glRotate(-45, 0.0, 0.0, 1.0);  
glTranslate(2.0, 0.0, 0.0);  
glRotate(90, 0.0, 0.0, 1.0);
```

Qual das seguintes opções corresponde ao objecto transformado?  
Justifique, indicando cada um dos passos intermédios.



a)



b)

c)

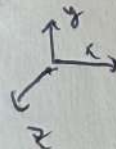
- ✓ Considere um quadrado centrado na origem cujas arestas têm 1 unidade de comprimento e a seguinte matriz de transformação geométrica 2D:

$$\begin{bmatrix} -0.707 & 0.707 & 1 \\ -0.707 & -0.707 & 2 \\ 0 & 0 & 1 \end{bmatrix}$$

Desenhe o sistema de coordenadas após a transformação, e desenhe o quadrado na sua posição antes e depois da transformação apresentada.

8. ✓ Considere o seguinte excerto de código :

```
translate(0, 0, -2);  
drawSphere (); // esfera 1  
translate(0, 0, 2);  
gluLookAt(3, 0, -3, 0, 0, -3, 0, 1, 0);  
translate (-3, 0, 0);  
drawSphere (); // esfera 2
```



Considerando somente o plano XZ do espaço global, desenhe e identifique a posição das esferas, a posição da câmara e o sistema de eixos da câmara.

4. Pretende-se colocar uma câmara na circunferência de raio unitário com centro na origem, como ilustrado na figura.

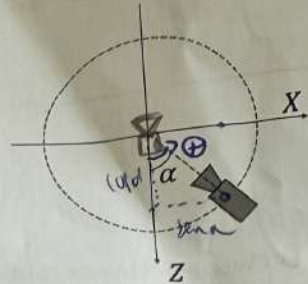
(a) Escreva os parâmetros da função `gluLookAt`, sabendo que os três primeiros parâmetros representam a posição da câmara, os três seguintes indicam o ponto para onde a câmara aponta, e os três últimos definem o vector "up".

`gluLookAt( _ , _ , _ , _ , _ , _ , _ , _ , _ );`

(b) Recorrendo somente a rotações e translações, escreva a sequência de transformações geométricas apropriadas para obter exactamente a mesma definição da câmara.

`glRotate( _ , _ , _ , _ );`

`glTranslate( _ , _ , _ );`



5. Considere uma câmara posicionada no ponto  $(-3, 0, -2)$ , a olhar na horizontal, numa direção que faz 45 graus com o eixo Z negativo.

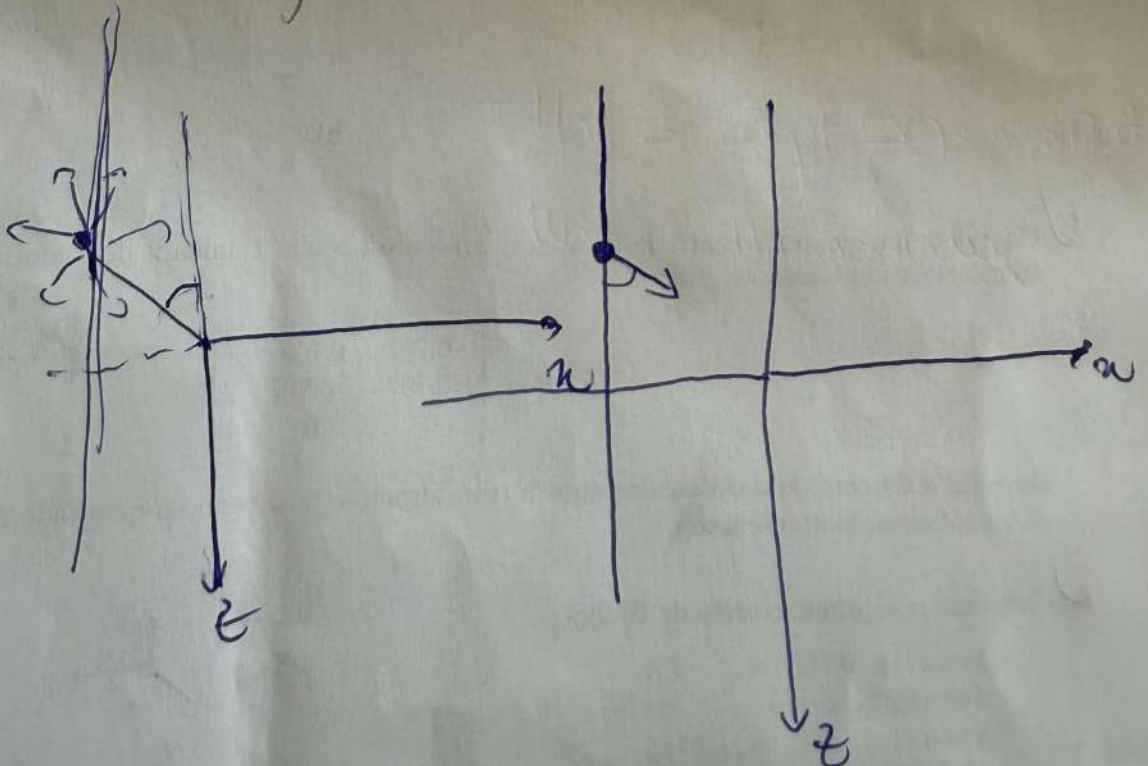
Apresente os cálculos necessários para definir a câmara usando a instrução:

`gluLookAt( p1, p2, p3, l1, l2, l3, u1, u2, u3 );`

gluLookAt( -3, 0, -2, 1, 0, 0, 0, 1, 0 );

`glRotate( 45, 1, 0, 0 );`

`glTranslate( -3, 0, -2 );`





## Parte II - Curvas; Iluminação; Texturas; View Frustum Culling

2. Utilizando o método de De Casteljau apresente o diagrama para descobrir graficamente, sem realizar cálculos, o ponto  $t = 0.25$  da curva cúbica de Bézier com os seguintes pontos de controlo (em 2D):

- $P_0 (0, 0)$
- $P_1 (0, 1)$
- $P_2 (2, 0)$
- $P_3 (2, 1)$

(0,1)

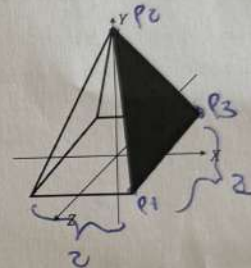
(2,1)

(0,0)

(2,0)

3. Considere duas das componentes da equação de iluminação: difusa e especular. Apresente a equação de cada componente suportada por um diagrama indicando claramente os elementos envolvidos na equação. No caso da luz especular considere a proposta Blinn-Phong.

4. Considere uma pirâmide quadrangular cuja base tem área 4 e altura 3. Apresente a expressão para o cálculo da normal da face sombreada na figura.



1,8 : 2

2,7 : 10,7

10,4

5. Considere a imagem representativa da mira técnica utilizada pela RTP em 1956 aplicada como uma textura a um quad (polígono com 4 vértices). Um exemplo da definição das coordenadas de textura, tendo como resultado a imagem esquerda, pode ser representado com o seguinte código:

```
glBegin(GL_QUADS);
glTexCoord2f(0.0, 0.0); glVertex3f(-1.0f, -1.0f, 0.0f);
glTexCoord2f(1.0, 0.0); glVertex3f( 1.0f, -1.0f, 0.0f);
glTexCoord2f(1.0, 1.0); glVertex3f( 1.0f,  1.0f, 0.0f);
glTexCoord2f(0.0, 1.0); glVertex3f(-1.0f,  1.0f, 0.0f);
glEnd();
```

3,5 : 2

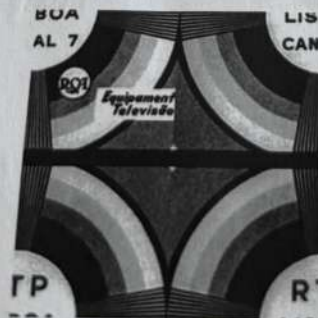
1,5 + 0,15

1,75

1,75

3,5

No código que se segue, defina as coordenadas de textura em falta de modo a obter como resultado a imagem direita.



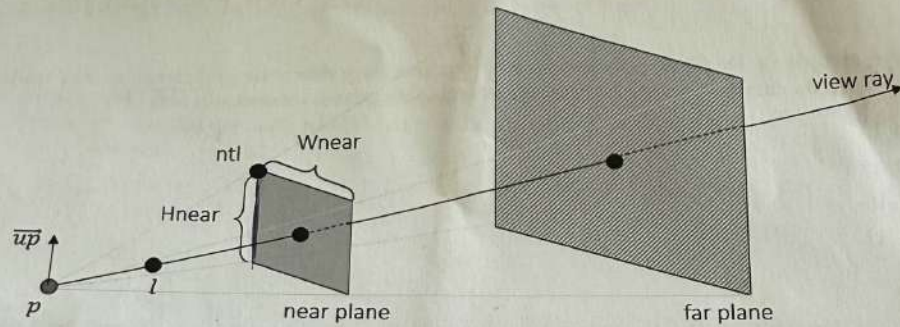
```
glBegin(GL_QUADS);
glTexCoord2f(, ); glVertex3f(-1.0f, -1.0f, 0.0f);
glTexCoord2f(, ); glVertex3f( 1.0f, -1.0f, 0.0f);
glTexCoord2f(, ); glVertex3f( 1.0f,  1.0f, 0.0f);
glTexCoord2f(, ); glVertex3f(-1.0f,  1.0f, 0.0f);
glEnd();
```

RTP RTP  
Lisboa Lisboa  
RTP RTP RTP  
Lisboa Lisboa Lisboa



10. Assuma as seguintes instruções OpenGL:

```
gluPerspective(fov, ratio, nearDist, farDist);  
gluLookAt(px, py, pz, lx, ly, lz, upx, upy, upz)
```



Considerando a figura que ilustra um frustum, indique como calcular:

- os valores  $W_{near}$  e  $H_{near}$
- assumindo que os valores da alínea anterior estão calculados, calcule o ponto  $ntl$

$$H_{far} = 2 \tan\left(\frac{fov}{2}\right) \times farDist$$

$$W_{far} = H_{far} \times ratio$$

$$H_{near} = 2 \tan\left(\frac{fov}{2}\right) \times nearDist$$

$$W_{near} = H_{near} \times ratio$$