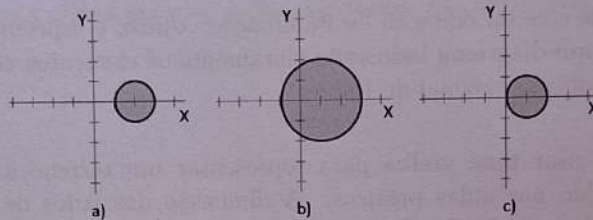
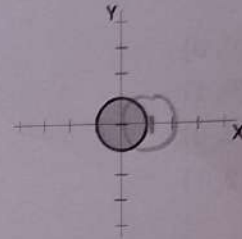


## Parte I - Transformações Geométricas, Câmera

1. Considere uma primitiva gráfica para desenhar uma esfera com centro na origem e raio unitário, e a seguinte sequência de transformações geométricas aplicadas à esfera:

```
glScale(0.5, 0.5, 0.5);  
glTranslate(2.0, 0.0, 0.0);  
glScale(2.0, 2.0, 2.0);
```

Qual das seguintes opções corresponde à esfera transformada? Justifique apresentando os passos intermédios.



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

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

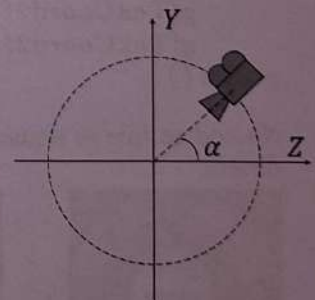
3. Pretende-se colocar uma câmara na circunferência de raio unitário com centro na origem, como ilustrado na figura.

- (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( _ , _ , _ );
```



4. Considere o seguinte excerto de código :

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

5. Considere que uma câmara está definida com a seguinte instrução:

```
gluLookAt( p1, p2, p3, 11, 12, 13, u1, u2, u3 );
```

Apresente o processo de cálculo para mover a câmara para cima uma unidade (considerando o referencial da câmara), mantendo a direcção do olhar, recorrendo somente à informação fornecida na instrução.

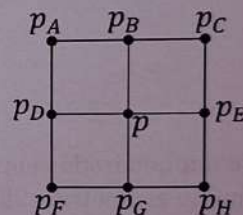
## Parte II - Curvas de Bezier; Iluminação; Texturas; View Frustum Culling

6. Utilizando o método de De Casteljau apresente o diagrama para descobrir graficamente o ponto quando  $t = 0.25$  de uma curva cúbica de Bézier. Considere os seguintes pontos de controlo (em 2D):

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

7. 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 original de Phong.

8. Considere que se pretende usar uma grelha para representar um terreno à semelhança do que foi pedido nas aulas práticas. A dimensão dos lados de cada quadrícula da grelha é uma unidade. Para se poder calcular a iluminação dos pontos da grelha é necessário calcular a normal em cada ponto. Com base na figura, representativa de uma quadrícula do terreno, indique como proceder matematicamente para calcular a normal do ponto  $p$ .



9. 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();
```

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

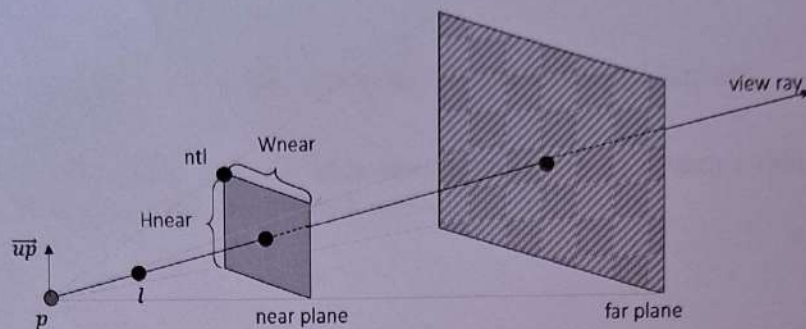


```
glBegin(GL_QUADS);
    glTexCoord2f(, ); glVertex3f(-1.0f, -1.0f, 0.0f);
    glTexCoord2f(, ); glVertex3f( 1.0f, -1.0f, 0.0f);
    glTexCoord2f(, ); glVertex3f( 1.0f,  1.0f, 0.0f);
    glTexCoord2f(, ); glVertex3f(-1.0f,  1.0f, 0.0f);
glEnd();
```



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:

- (a) os valores  $W_{near}$  e  $H_{near}$
- (b) assumindo que os valores da alínea anterior estão calculados, calcule o ponto  $fbr$