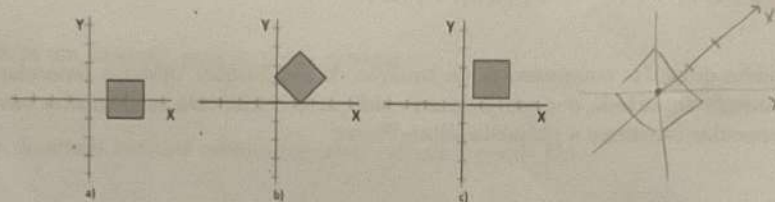
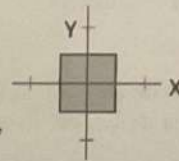


Parte I - Transformações Geométricas, Câmara

1. Considere uma primitiva gráfica para desenhar um cubo com centro na origem e lado com dimensão de 2 unidades, e a seguinte sequência de transformações geométricas a aplicar ao cubo:

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

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



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

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

Desenhe o cubo na sua posição antes e depois da transformação geométrica apresentada.

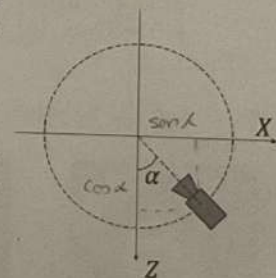
3. Pretende-se colocar uma câmara na circunferência de raio unitário com centro na origem, como ilustrado na figura.

- (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".

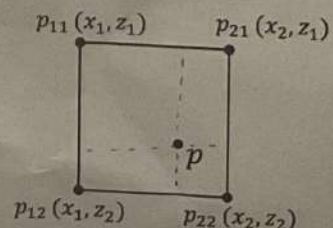
```
gluLookAt( , , , , , , , , , );
```

- (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 que se pretende usar uma grelha para representar um terreno, à semelhança do que foi pedido nas aulas práticas. As coordenadas dos pontos da grelha são números inteiros e a dimensão dos lados de cada quadricla da grelha é uma unidade. Para obter a altura dos pontos da grelha é disponibilizada a função $h(p_{ij})$, sendo p_{ij} um ponto da grelha. Com base na figura, indique como proceder matematicamente para calcular a altura do ponto p utilizando interpolação bilinear.



5. Considere o seguinte excerto de código :

```
glTranslatef(0, 2, 0);
glutSolidSphere(1, 16, 16); // esfera 1 (raio 1, 16 slices e 16 stacks)
glTranslatef(0, -2, 0);
gluLookAt( 5, 0, 5, 0, 0, 0, 0, 1, 0);
glTranslatef(3, 0, 3);
glutSolidSphere(1, 16, 16); // esfera 2
...
```

Altere os parâmetros da função `gluLookAt` de forma a que ambas as esferas fiquem na mesma posição no espaço global.

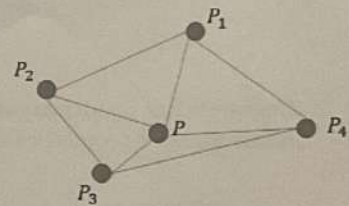
Parte II - Curvas de Bezier; Iluminação; Texturas; View Frustum Culling

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):

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

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 Blinn-Phong.

8. Considere um modelo representado por um conjunto de triângulos numa grelha irregular. Apresente um método para calcular um vector que possa ser considerado uma aproximação ao vector normal para o 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();
```

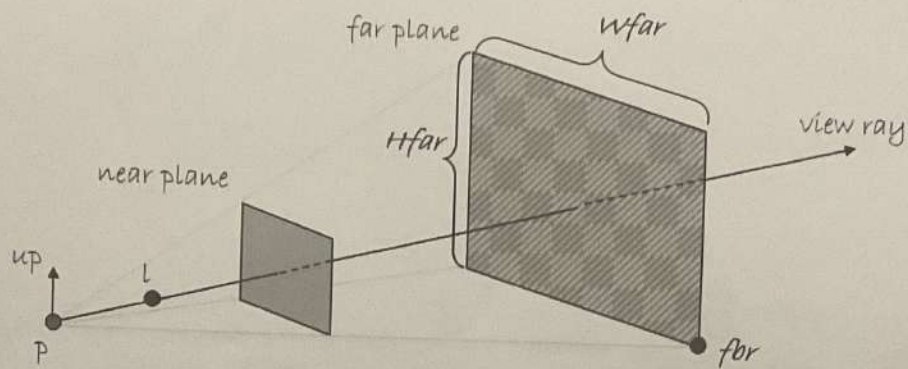
No código que se segue, defina as coordenadas de textura em falta de modo a obter como resultado a imagem direita.



```
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:

- os valores $Wfar$ e $Hfar$
- assumindo que os valores da alínea anterior estão calculados, calcule o ponto fbr