

1. Considere a matriz A , obtida após uma sequência de transformações geométricas. Indique a sequência incorrecta para gerar a matriz A a partir da matriz identidade, e apresente a matriz resultante da respectiva opção.

$$A = \begin{bmatrix} 4 & 0 & 0 & 4 \\ 0 & 4 & 0 & 4 \\ 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- (a) `glTranslatef(1,1,1);`
`glScalef(4,4,4);`
- (b) `glTranslatef(4,4,4);`
`glScalef(4,4,4);`
- (c) `glScalef(4,4,4);`
`glTranslatef(1,1,1);`

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

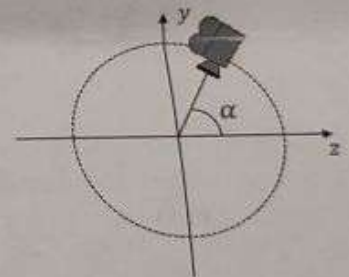
- (a) Escreva os parâmetros da função `gluLookAt`, sabendo que os três primeiros parâmetros representam a posição da câmara, os três seguintes indicam o ponto para onde a câmara aponta, e os três últimos definem o vector "up".

`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(_ , _ , _);`



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

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

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

4. 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.

5. Considere uma curva de Bézier de grau 3 em 2D com os seguintes pontos de controle: $P_0 = (0, 0)$, $P_1 = (0, 1)$, $P_2 = (2, 0)$, $P_3 = (2, 1)$.

Apresente o procedimento **geométrico** para o cálculo de $P(t)$ considerando $t = 0.25$. Desenhe uma aproximação da curva obtida.

6. 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();
```

7. Por forma a tornar eficiente o algoritmo de view frustum culling é necessário implementar algum mecanismo de agrupamento de triângulos. Descreva o processo de partição espacial 3D baseado em k-D trees.
8. Considerando a seguinte divisão do espaço, construa a árvore de partição binária (BSP) que a representa e indique qual a ordem de desenho das primitivas de forma a **minimizar a escrita de pixels**.

