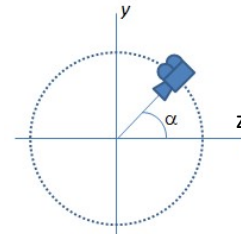


Ficha de Consolidação II

Transformações Geométricas

1. Considere que se pretende 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 um ponto para onde a câmara está a apontar, e os três últimos parâmetros definem o vector "up";

i) `gluLookAt(____, ____ , ____ , ____ , ____ , ____ , ____ , ____ , ____);`

- b) Recorrendo somente a rotações e translações, escreva a sequência de transformações geométricas apropriadas para obter exactamente a mesma definição da câmara (pode utilizar funções como *sin* e *cos*).

i) `glRotate(____, ____ , ____ , ____);`

ii) `glTranslate(____, ____ , ____);`

2. Considere o seguinte excerto de código :

```
gluLookAt( 5, 0, 5,      0, 0, 0,      0, 1, 0);  
drawEsfera(); // desenha esfera de raio 1 centrada na origem
```

De acordo com o seguinte código, assinale as afirmações verdadeiras:

- a) No espaço global a esfera é desenhada com o centro em $(0, 0, 0)$.
- b) No espaço câmara a esfera é desenhada com o centro em $(-5, 0, -5)$.
- c) No espaço câmara a esfera é desenhada com o centro no eixo Z.
3. Considere que uma câmara está definida com a seguinte instrução:
- ```
gluLookAt(p1, p2, p3, l1, l2, l3, u1, u2, u3);
```
- a) Apresente o processo de cálculo para mover a câmara para a esquerda uma unidade, mantendo a direcção do olhar, recorrendo somente à informação fornecida na instrução.

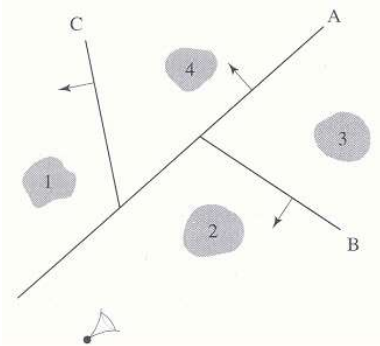
- b) Apresente o processo de cálculo para mover a câmara para cima uma unidade, mantendo a direcção do olhar, recorrendo somente à informação fornecida na instrução.
4. Considere que se pretende adicionar uma câmara no modo explorador numa aplicação em OpenGL. Apresente os cálculos, considerando coordenadas esféricas, para determinar a primeira componente da função `gluLookAt` (a posição da câmara) assumindo que a câmara está sempre a olhar para a origem. Considere um ângulo vertical  $\alpha$ , e um ângulo horizontal  $\beta$ . Ilustre graficamente os cálculos efectuados.
5. Considere que se pretende adicionar uma câmara no modo FPS numa aplicação em OpenGL. Apresente os cálculos, considerando coordenadas esféricas, para determinar a segunda componentes da função `gluLookAt` (o ponto para onde está a olhar) considerando um ângulo vertical  $\alpha$  e um ângulo horizontal  $\beta$ . Assuma que a câmara se encontra posicionada no ponto  $P(x,y,z)$ . Ilustre graficamente os vectores e pontos considerados.
6. Considere o seguinte excerto de código:

```
translate(0, 0, -3);
drawEsfera ();
translate(0, 0, 3);
gluLookAt(px, py, pz, 0, 0, -1, 0, 1, 0);
translate (0, 0, -10);
drawEsfera ();
```

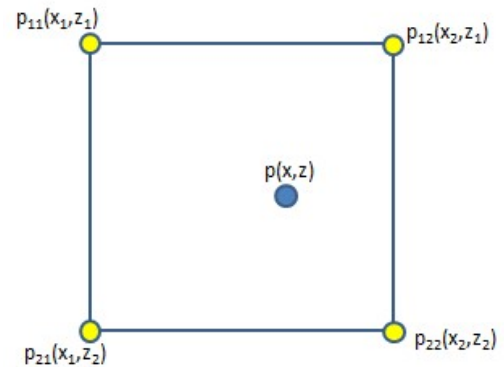
De acordo com o código acima comente as seguintes afirmações (recorra a diagramas para suportar a resposta):

- a) A posição do centro da primeira esfera no espaço câmara é  $(0,0,-3)$ .
- b) Caso  $(px,py,pz) = (0, 0, -3)$ , a posição do centro da segunda esfera no espaço câmara é  $(0, 0, -7)$ .
- c) Caso  $(px,py,pz) = (0, 0, -5)$ , a segunda esfera não é visível .
- d) Caso  $(px, py, pz) = (0, 0, 0)$ , a segunda esfera não é visível .
- e) Caso  $(px, py, pz) = (0, 0, -7)$ , as esferas ocupam a mesma posição no espaço câmara.

7. Considere a seguinte divisão do espaço utilizando uma BSP. Construa a árvore correspondente e, dada a posição da câmara indicada na figura, apresente a ordem de desenho dos objectos de forma a garantir a ordem de escrita dos pixels.



8. Considere que se pretende usar uma grelha para representar um terreno, à semelhança do que foi pedido no trabalho prático. As coordenadas dos pontos da grelha são números inteiros e a dimensão dos lados de cada quadrícula 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  $p$ .



9. Considere a biblioteca gUM que contem primitivas gráficas para cadeiras e mesas como se ilustra nas figuras. Escreva uma função em C que permita construir em OpenGL um modelo semelhante ao apresentado na figura com a cena das mesas e cadeiras. Como referência, em termos de medidas, considere que a mesa tem um raio de 1 unidade, e que as cadeiras têm os lados do tampo com 0,4 unidades.

