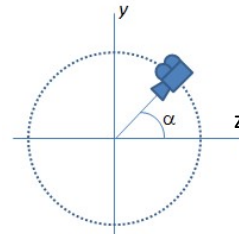


Ficha de Consolidação II

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(0, sin(-a) cos(-a), 0, 0, 0, 0, 0, 1, 0);`

- 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(-a, 1, 0, 0);`

ii) `glTranslate(0, 0, 1);`

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).

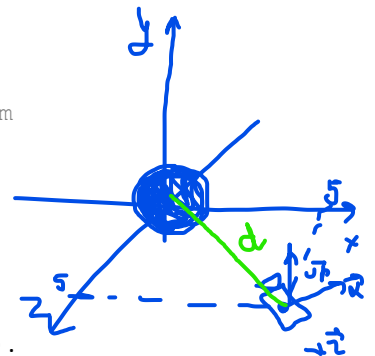
Verdade

- b) No espaço câmara a esfera é desenhada com o centro em (-5, 0, -5).

Falso. O espaço câmara corresponde ao espaço do ponto de vista do referencial da câmara, logo $d^2 = 5^2 + 5^2 \Leftrightarrow d = \sqrt{50}$. Assim, o centro da esfera no espaço câmara é (0,0,-sqrt(50)).

- c) No espaço câmara a esfera é desenhada com o centro no eixo Z.

Verdade. No espaço câmara a esfera é desenhada com centro no eixo dos Z (0,0,-sqrt(50)).

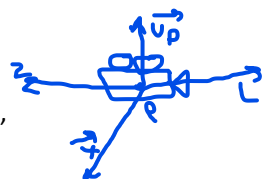


3. Considere que uma câmara está definida com a seguinte instrução:

```
gluLookAt( p1, p2, p3, 11, 12, 13, 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.

$d = L - P / ||L - P||$
 $newPos = (up * d) * 1 + P$
 $newLook = newPos + d$



- 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.

$$d = L - P / \|L - P\|$$

$$x = up * d$$

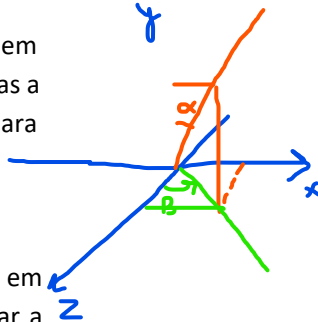
$$newPos = (x * d) * 1 + P$$

$$newLook = newPos + d$$

`gluLookAt(newPos, newLook, up)`

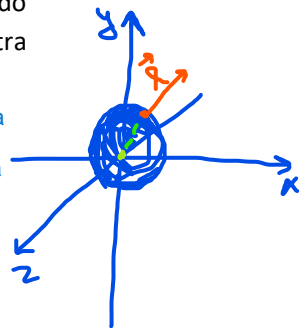
4. Considere que se pretende adicionar uma câmara no modo explorador numa aplicação em OpenGL. Apresente os cálculos, considerando coordenadas esféricas, para determinar as a primeira componente da função `gluLookAt` (a posição da câmara) assumindo que a câmara está sempre a olhar para a origem. Considere um ângulo vertical α , e um ângulo horizontal β . Ilustre graficamente os cálculos efectuados.

Seja r o raio da esfera, $l_1 = \cos(\alpha) * r$.
 $\rightarrow z = \cos \beta * \sin \alpha * r$
 $\rightarrow y = \cos \alpha * r$
 $\rightarrow x = \sin \beta * \sin \alpha * r$



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 α e um ângulo horizontal β . Assuma que a câmara se encontra posicionada no ponto $P(x, y, z)$. Ilustre graficamente os vectores e pontos considerados.

Seja r o raio da esfera. $\rightarrow lz = \cos \beta * \sin \alpha * r + \cos \beta * \sin \alpha$
 $\rightarrow ly = \cos \alpha * r + \cos \alpha$
 $\rightarrow lx = \sin \beta * \sin \alpha * r + \sin \beta * \sin \alpha$

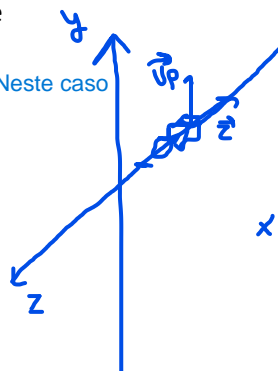


6. Considere o seguinte excerto de código:

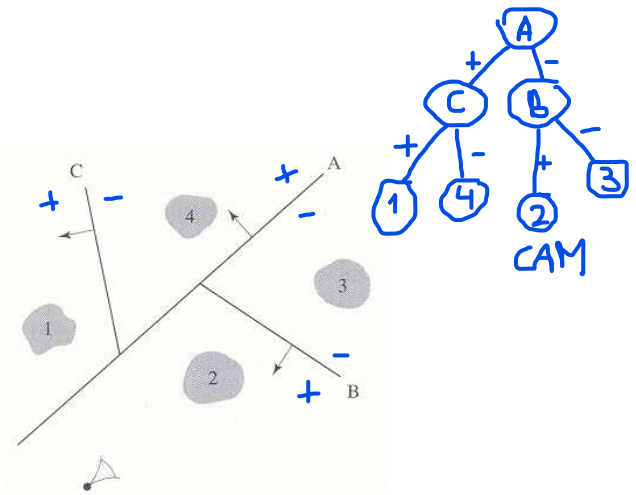
```
translate(0, 0, -3);
drawEsfera ();
translate(0, 0, 3);
gluLookAt(px, py, pz, 0, 0, -1, 0, 1, 0);
translate (0, 0, -10);
drawEsfera ();
```

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 $(0, 0, -3)$.
 Verdade. É feita uma translação $(0, 0, -3)$ sobre o referencial da câmara e é desenhada uma esfera sendo o referencial posicionado novamente onde se encontrava.
- b) Caso $(px, py, pz) = (0, 0, -3)$, a posição do centro da segunda esfera no espaço câmara é $(0, 0, -7)$.
 Falso. Como a câmara olha para o lado negativo do eixo Z, a esfera teria um centro com $z > 0$. Neste caso teria centro em $(0, 0, 7)$.
- c) Caso $(px, py, pz) = (0, 0, -5)$, a segunda esfera não é visível.
 Verdade. A esfera encontra-se atrás da câmara.
- d) Caso $(px, py, pz) = (0, 0, 0)$, a segunda esfera não é visível.
 Verdade. Como a câmara está a olhar para $(0, 0, -1)$ e na posição $(0, 0, 0)$, a esfera 1 em $(0, 0, -3)$ irá estar a tapar a esfera 2 $(0, 0, -10)$.
- e) Caso $(px, py, pz) = (0, 0, -7)$, as esferas ocupam a mesma posição no espaço câmara.
 Falso. No espaço câmara a esfera 1 encontra-se em $(0, 0, -3)$ e a 2 em $(0, 0, 3)$.



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.



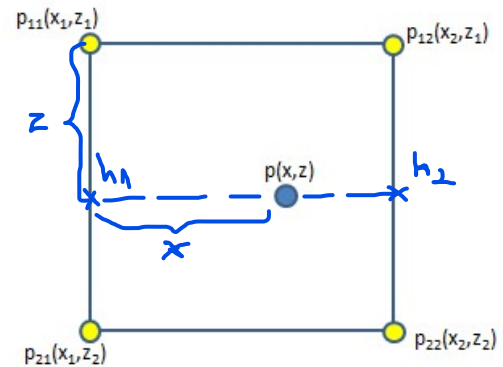
1º-2
2º-3
3º-1
4º-4

$$h1 = (1-z) * p(x1,z1) + z * p(x1,z2)$$

$$h2 = (1-z) * p(x2,z1) + z * p(x2,z2)$$

$$hp = (1-x) * h1 + x * h2$$

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.

