FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA
departamento de engenharia informática

Computação Gráfica

André Perrotta (avperrotta@dei.uc.pt)
Hugo Amaro (hamaro@dei.uc.pt)

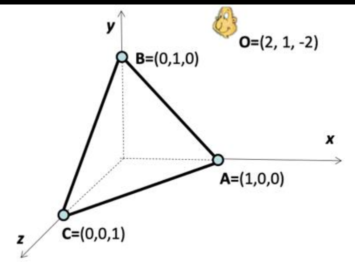
TP_11: Revisão semestre - exercícios

Geometria e Transformações

Considere um triângulo definido pelos vértices A, B e C, de coordenadas

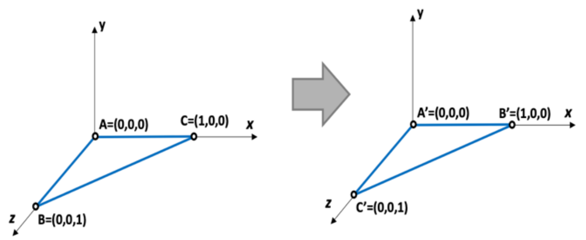
$$A=(1,0,0),\ B=(0,1,0),\ C=(0,0,1)$$

assente no plano $x+y+z=1$.
Existe um observador localizado na posição **O=(2, 1, -2).**
Use a regra da mão direita para identificar qual o lado visível (ou da frente) para o observador.
Espera-se a realização de cálculos rigorosos para o efeito.

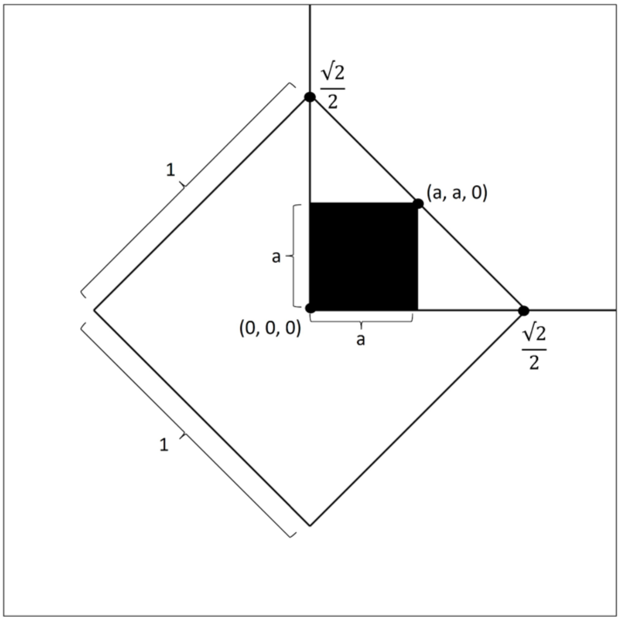


Considere o seguinte triângulo, existente no plano y=0, definido pelos vértices A=(0,0,0), B=(0,0,1) e C=(1,0,0).

Acha possível definir uma transformação que permita virar o triângulo ao contrário, como indicado na figura?
Isto é, os vértices B e C trocam de lugar, mantendo o vértice a sua posição original.



A cena a seguir foi desenhada com OpenFrameworks/OpenGl. Os valores de coordenadas e tamanhos foram adicionados em pós-processamento da imagem gerada pelo código, e estão em valores de "coordenadas mundo". A aplicação foi configurada para uma janela de 1024x1024 pixels, definida em main.cpp.



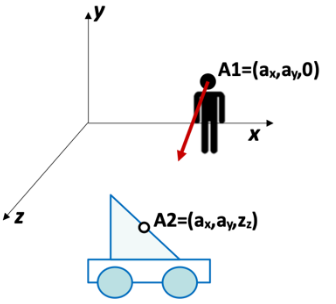
Complete o código com os valores necessários para a gerar a imagem apresentada anteriormente. Se achar pertinente, coloque "//" (comments) nas transformações que julgar desnecessárias.

```
void draw(){

  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(−1, 1, −1, 1, −2, 2);
  glMatrixMode(GL_MODELVIEW);
  lookat(0, 0, 1, 0, 0, 0, 0, 1, 0);

  glPushMatrix();
    glTranslatef(___, ___, ___);
    glRotatef(___, ___, ___, ___);
    glScalef(___, ___, ___);

    glPushMatrix();
      glColor3f(0, 0, 0);
      glTranslatef(___, ___, ___);
      glScalef(___, ___, ___);
      glPolygonMode(GL_FRONT_AND_BACK, GL_LINE);
      rect();
    glPopMatrix();

    glPushMatrix();
      glColor3f(0, 0, 0);
      glTranslatef(___, ___, ___);
      glRotatef(___, ___, ___, ___);
      glScalef(___, ___, ___);
      glPolygonMode(GL_FRONT_AND_BACK, GL_FILL);
      rect();
      glPopMatrix();

  glPopMatrix();

  glPushMatrix();
    axis();
  glPopMatrix();

}
```

Camera, projeção

Considere a existência de um observador localizado na posição A1=(ax, ay, 0).
Considere o ponto A2, de coordenadas A2=(ax, ay, az), pertencente à vela do veículo, como se mostra na figura.
O observador está a olhar para o ponto A2, encontrando-se de pé, isto é orientado para cima.

Nestas condições determine a matriz de transformação que permite obter as coordenadas do ponto A2 (e do veículo, caso seja desejado) em função do referencial do observador.



Considere uma aplicação desenvolvida em OpenGl e configurada com uma janela quadrada (largura=altura). Mais, considere que no inicio do programa é configurado o recorte (glViewPort) para toda a tela e que seu volume de projeção é configurado através da função $glOrtho(left, right, bottom, top, near, far)$ com os seguintes valores:

$glOrtho(-1, 1, -1, 1, -2, 2)$.

(a) (1 valor): Considerando que a matriz Modelview está em suas condições iniciais (identidade). Quais valores de x, y, z podemos atribuir à um vértice de forma a garantir que o mesmo se encontra dentro do volume de projeção?

Considere agora que, para além da projeção, é também definida uma vista da cena utilizando o algoritmo UVN implementado na função $lookat(\vec{pos}, \vec{target}, \vec{up})$ com os seguintes valores:

$lookat(0, 0, 1, 0, 0, 0, 0, 1, 0)$

E, alteram-se os valores de recorte, utilizando a função $glViewport(x0, y0, width, height)$ com os seguintes valores:

$glViewport(0, \frac{h}{2}, \frac{w}{4}, \frac{h}{2})$, onde w e h referem à largura e altura da janela da aplicação.

Após estas definições é então desenhada uma linha entre os vértices $A(-\frac{1}{2}, 0, -1)$ e $B(-\frac{1}{2}, 0, 1)$.

(b) (1 valor): Qual a distância entre os vértices A e B em coordenadas mundo? (justifique)

(c) (1 valor): Qual a distância entre os vértices A e B em pixels na janela da aplicação? (justifique)

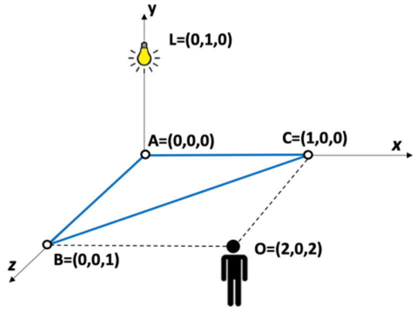
Iluminação

4.1 .

Numa determinada cena existe um triângulo, um observador e uma fonte de iluminação, como se mostra na figura seguinte

Objeto:
Triângulo amarelo existente no plano y=0, definido pelos vértices A=(0,0,0), B=(0,0,1) e C=(1,0,0). As constantes de reflexão ambiente, difusas e especulares são: ka=0; kd=ks=0.4.

Observador: Encontra-se localizado na posição O=(2,0,2)
Iluminação: Fonte de luz pontual, verde, localizada na posição L=(0,1,0).



4.1.1

Considerando o modelo de Phong caracterize (cor + intensidade) do ponto do triângulo que apresenta a máxima **componente difusa**. Note que a resposta é valida para qualquer ponto do triângulo e não apenas para os vértices A, B ou C. Justifique.

4.1.2

Considerando o modelo de Phong caracterize (cor + intensidade) do ponto do triângulo que apresenta a máxima **componente especular**. Note que a resposta é valida para qualquer ponto do triângulo e não apenas para os vértices A, B ou C. Justifique.

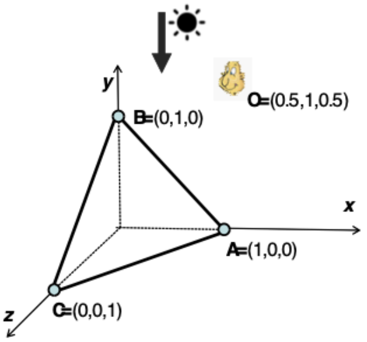
5.2

Considere novamente o triângulo apresentado na alínea 2.2, definido pelos vértices A, B e C, de coordenadas

A=(1,0,0), B=(0,1,0), C=(0,0,1)

O observador está agora localizado na posição **O=(0.5, 1, 0.5)**.
Existe uma luz direccional, apenas com **componente especular**, de cima para baixo, como se mostra na figura.

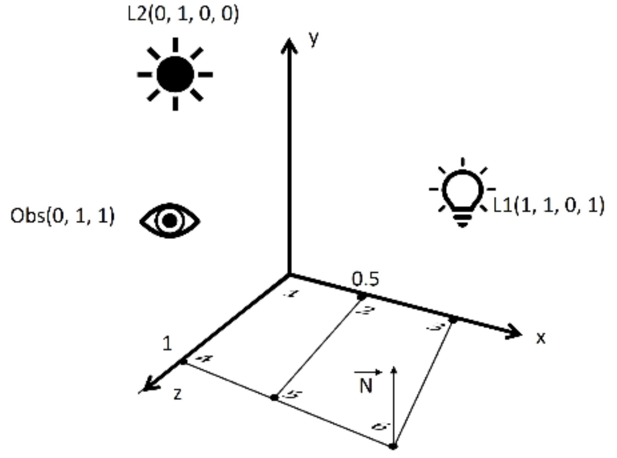
Determine o ponto mais brilhante do triângulo.
Esperam-se cálculos rigorosos na resolução deste exercício



A imagem abaixo mostra uma cena realizada em OpenGl formada por um malha de retângulos de resolução (2, 1), duas fontes de luz, L_1 e L_2, e um observador situado na posição $obs(0,1,1)$. A malha foi construida de forma a atribuir materiais com cores determinadas para cada vértice conforme é mostrado na tabela. Os coeficientes de reflexão ambiente k_A, difusa k_D e especular k_S do material são iguais a 1 ($k_A = k_D = k_S = 1$) e o coeficiente de especularidade ns também vale 1 ($ns = 1$). A normal $\vec{N}(0,1,0)$ é a mesma em todos os vértices. As fontes de luz estão configuradas conforme especificado abaixo.

$$L_{1_{pos}} = (1,1,0,1)$$
$$L_{1_{amb}}(R,G,B) = (0,0,0)$$
$$L_{1_{dif}}(R,G,B) = (0,0,0)$$
$$L_{1_{spec}}(R,G,B) = (0,0,1)$$

$$L_{2_{pos}} = (0,1,0,0)$$
$$L_{2_{amb}}(R,G,B) = (0,0,0)$$
$$L_{2_{dif}}(R,G,B) = (1,0,0)$$
$$L_{2_{spec}}(R,G,B) = (0,0,0)$$



i	R	G	B
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0

(a) (2 valor): Qual é o vértice com maior intensidade de luz? (justifique)

(b) (2 valor): Qual é o vértice com menor intensidade de luz? (justifique)

(c) (2 valor): Qual a cor e intensidade de luz, em valores R, G, B), no vértice 1? (justifique)

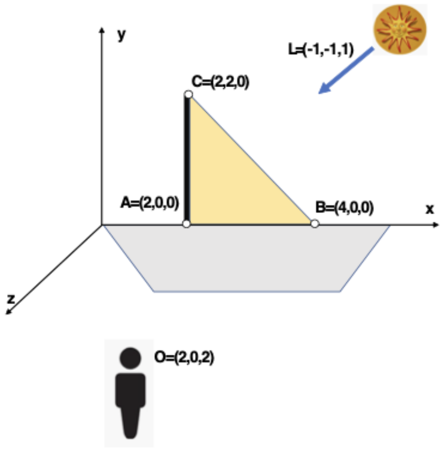
Textura

2 Grupo 2

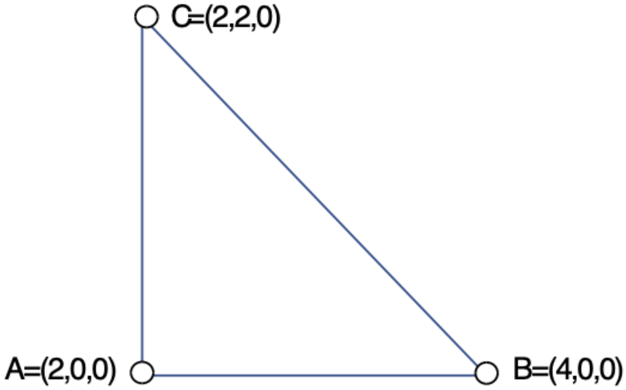
Considere um barco, como se mostra na figura ao lado.
A vela é um triângulo, definido pelos vértices A=(2,0,0); B=(4,0,0); C=(2,2,0).
Considere a imagem da bandeira de Portugal (textura), e o código openGL relativo à aplicação da textura à vela o barco.


```
glBegin(GL_TRIANGLES);
 glTexCoord2f(0, 0); glVertex3f(2, 0, 0);
 glTexCoord2f(2, 0); glVertex3f(2, 2, 0);
 glTexCoord2f(0, 2); glVertex3f(4, 0, 0);
glEnd();
```

Assumindo que a textura não se repete, faça um esboço de como a vela do barco será visivel quando lhe é aplicada a textura.



Resposta:



2.1

4

Considere a face de um paralelepípedo, definido pelos vértices (A,B,C,D), uma textura (o gato da figura 1) e o resultado da aplicação desta textura à face do paralelepípedo. Sabe-se ainda que as dimensões da textura estão normalizadas.
Dado o seguinte código em openGL que implementa o mapeamento pretendido, defina os valores dos parâmetros em falta: **sA,tA, sB, tB, sC, tC, sD, tD.**

```
glBegin(GL_POLYGON);
	glTexCoord2f(sA, tA);	glVertex3f(A);
	glTexCoord2f(sB, tB);	glVertex3f(B);
	glTexCoord2f(sC, tC);	glVertex3f(C);
	glTexCoord2f(sD, tD);	glVertex3f(D);
glEnd();
```

Figura 1