Sistemas Gráficos e Interacção

Época Normal 2017-01-25

N.º _____ Nome _____

Duração da prova: 75 minutos
Cotação de cada pergunta: assinalada com parêntesis rectos
Perguntas de escolha múltipla: cada resposta incorrecta desconta 1/3 do valor da pergunta

Parte Teórica **30%**

a. **[2.5]** Os formatos **BMP** e **SVG (*Scalable Vector Graphics*)** de representação de gráficos constituem exemplos de

   i. Representações matriciais
   ii. Representações vectoriais
   iii. Uma representação matricial e uma vectorial, respectivamente
   iv. Uma representação vectorial e uma matricial, respectivamente

b. **[2.5]** Qual das seguintes transformações corresponde a uma reflexão no plano OXY?

   i. `glTranslated(0.0, 0.0, -1.0);`
   ii. `glRotated(180.0, 0.0, 0.0, 1.0);`
   iii. `glScaled(-1.0, -1.0, 0.0);`
   iv. Nenhuma das anteriores

c. **[2.5]** Se se efectuar uma operação de divisão perspectiva do ponto de coordenadas homogéneas $[1.0, 2.0, 3.0, 4.0]^T$ ir-se-á obter

   i. $[4.0, 8.0, 12.0, 1.0]^T$
   ii. $[0.25, 0.5, 0.75, 1.0]^T$
   iii. $[-3.0, -2.0, -1.0, 0.0]^T$
   iv. Nenhuma das anteriores

d. **[2.5]** Na representação por fronteira de um objecto (***B-rep***)

   i. É difícil exibir um ponto sobre a superfície do objecto
   ii. É fácil determinar, dado um ponto, se o mesmo está no interior, na fronteira ou no exterior do objecto
   iii. As operações booleanas são complicadas de efectuar
   iv. Nenhuma das anteriores

e. **[2.5]** Nas representações de objectos por células

   i. As grelhas dividem o espaço em cubos cujos lados são potências de 2
   ii. As *octrees* dividem o espaço em cubos de igual dimensão
   iii. As *BSP-trees* dividem o espaço em poliedros convexos
   iv. Todas as anteriores

f. **[2.5]** O modelo de iluminação do OpenGL

   i. Requer a especificação de normais
   ii. Constitui um exemplo de um modelo de iluminação local
   iii. Contempla apenas os caminhos para a luz do tipo fonte luminosa → superfície → observador
   iv. Todas as anteriores

g. **[2.5]** Para iluminar uma cena com uma fonte de luz direccional, deverá

   i. Activar o modelo de iluminação do OpenGL
   ii. Especificar para a posição da fonte de luz um conjunto de coordenadas homogéneas tal que $w = 1$
   iii. Especificar para o ângulo de *cutoff* o valor de 180^o
   iv. Todas as anteriores

h. **[2.5]** No mapeamento de texturas em OpenGL, o processo de filtragem designado por GL_NEAREST_MIPMAP_NEAREST

   i. Escolhe o *texel* que mais se aproxima do centro do pixel no *mipmap* que melhor se adequa ao contexto de minificação existente
   ii. Calcula uma média pesada da matriz de 2 x 2 *texels* que mais se aproxima do centro do pixel no *mipmap* que melhor se adequa ao contexto de minificação existente
   iii. Escolhe o *texel* que mais se aproxima do centro do pixel em cada um dos dois *mipmaps* que melhor se adequam ao contexto de minificação existente; em seguida, efectua uma interpolação linear destes dois valores
   iv. Calcula uma média pesada da matriz de 2 x 2 *texels* que mais se aproxima do centro do pixel em cada um dos dois *mipmaps* que melhor se adequam ao contexto de minificação existente; em seguida efectua uma interpolação linear destes dois valores

Sistemas Gráficos e Interacção

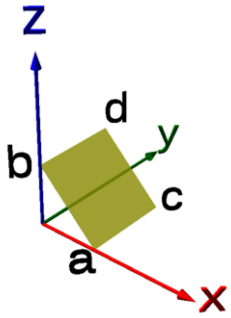
Época Normal 2017-01-25

Parte Teórico-Prática **40%**

Resolução: Em folhas próprias e separadas (A4). Não se esqueça de indicar, em cada folha, o número de estudante e o nome completo
Perguntas de escolha múltipla: cada resposta incorrecta desconta 1/3 do valor da pergunta
Nota: Em todas as perguntas, a menos que algo seja dito em contrário, assuma a posição da câmara por omissão

a. **[3.0]** Complete o código para modelar o quadrado apresentado na figura e defina a **normal unitária** correctamente, assumindo que a orientação por omissão dos polígonos no *OpenGL* não foi alterada.

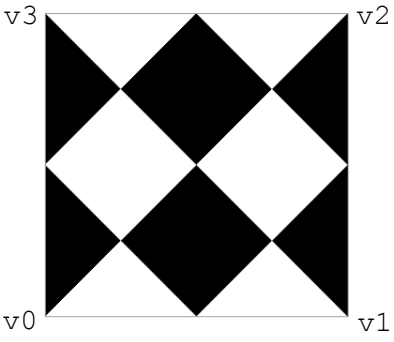
Os *arrays* `a`, `b`, `c` e `d` já estão definidos; os segmentos $\overline{ab}$ e $\overline{cd}$ fazem 45° com o plano oxy; e os segmentos $\overline{ac}$ e $\overline{bd}$ são paralelos ao eixo dos yy.




```
glBegin(GL_QUADS); // ou GL_POLYGON
  glNormal3f(cos(M_PI / 4.0), 0.0, cos(M_PI / 4.0));
  // ou sqrt(2.0) / 2.0
  glVertex3fv(a); // ou c, d, b, a
  glVertex3fv(c); // ou d, b, a, c
  glVertex3fv(d); // ou b, a, c, d
  glVertex3fv(b);
glEnd(); // se o quadrado fosse desenhado no sentido oposto (retrógrado ou horário), o sentido da normal também seria o oposto
```

b. **[3.0]** Aplique a textura apresentada na figura 1 a um quadrado, de modo a ficar com o aspecto apresentado na figura 2.

1)



2)

```
...
    glTexCoord2f(0.5, 0.5); // como alternativa
    glVertex3fv(v0);        // usar outros valores
    glTexCoord2f(2.5, 0.5); // desfasados de 2 unidades
    glVertex3fv(v1);        // e com parte decimal .5
    glTexCoord2f(2.5, 2.5); // ou então trocar s com t
    glVertex3fv(v2);        // e usar valores
    glTexCoord2f(0.5, 2.5); // com parte decimal .0
    glVertex3fv(v3);        // (isto é, rodar a imagem)
...
```

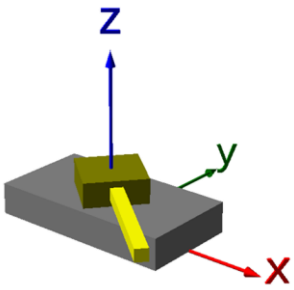
c. **[2.0]** Indique os parâmetros de configuração de texturas que é necessário realizar para o código acima funcionar correctamente.

Activar o modo de repetição do padrão de textura nos eixos s e t (ou seja, configurar `GL_TEXTURE_WRAP_S` e `GL_TEXTURE_WRAP_T` como `GL_REPEAT`).

Nota: mude para uma nova folha de respostas

d. **[3.0]** Pretende-se implementar a câmara de um tanque. A câmara está colocada por cima da torre e aponta sempre na direcção do canhão (paralela ao plano oxy).

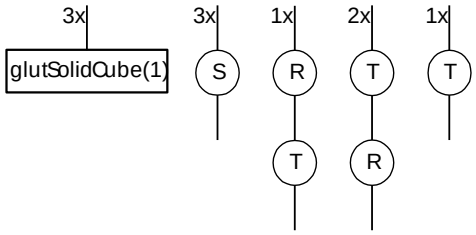
A posição do tanque é definida por `modelo.pos.x`, `modelo.pos.y` e `modelo.pos.z`; o ângulo do tanque por `modelo.tanque.direccao`; o ângulo da torre em relação à base por `modelo.tanque.angTorre`; e a altura da câmara em relação à base do tanque pela constante `ALTURA_CAM`.



Nota: Quando os ângulos são de 0°, a base, a torre e o canhão são desenhados alinhados com o eixo dos xx (o canhão aponta para a parte positiva do eixo).

```
gluLookAt(modelo.pos.x, modelo.pos.y, modelo.pos.z + ALTURA_CAM,
  modelo.pos.x + cos(modelo.tanque.direccao + modelo.tanque.angTorre),
  modelo.pos.y + sin(modelo.tanque.direccao + modelo.tanque.angTorre),
  modelo.pos.z + ALTURA_CAM,
  0.0, 0.0, 1.0);
```

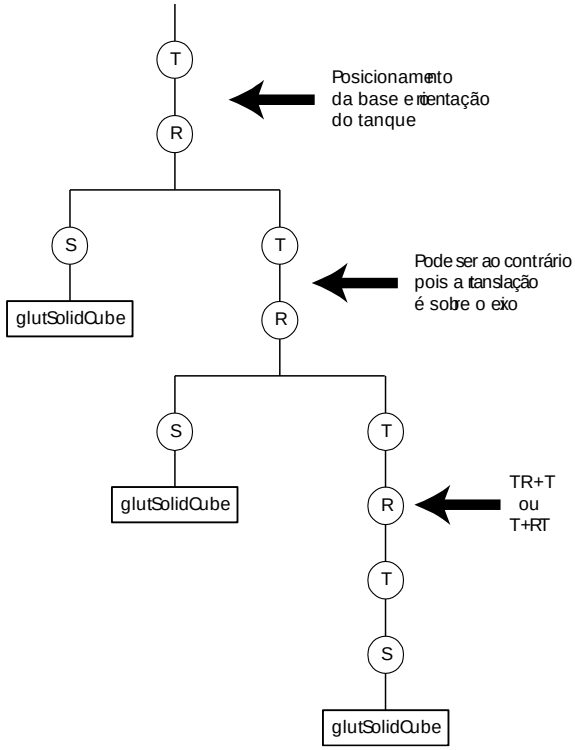
e. **[3.0]** Pretende-se implementar a árvore de desenho do tanque anterior, usando a função `glutSolidCube(1.0)` para desenhar cada um dos paralelepípedos. Crie a árvore de cena a partir dos blocos apresentados na figura.



Nota: O centro da torre está alinhado com o centro da base, e o canhão roda no ponto de ligação com a torre.

Sistemas Gráficos e Interacção

Época Normal 2017-01-25



Nota: mude para uma nova folha de respostas

f. **[2.0]** Considere um cubo com material azul (0.25, 0.25, 1.0) iluminado por uma única fonte de luz amarela (1.0, 1.0, 0.0). Qual será a cor (em termos das suas componentes primárias RGB) resultante?

R = 0.25 G = 0.25 B = 0.0

g. **[2.0]** Um objecto é iluminado por duas fontes de luz: uma emite componente especular vermelha; a outra emite componente especular verde. Indique como terá de configurar o material para que o reflexo especular fique com a cor da luz emitida.

Para a reflexão especular apresentar a cor da fonte de luz, a componente de reflexão especular do material tem que ser branca (independentemente da cor do material).

h. **[2.0]** Suponha que pretende saber qual o objecto que está a ser desenhado por baixo do cursor do rato. Indique qual o ***callback* GLUT e a técnica do OpenGL** que deve utilizar para conseguir identificar o objecto.

O *callback* a utilizar seria o `glutPassiveMotionFunc()` (ou o `glutMouseFunc()`, caso estivesse a pressionar um botão do rato) e a técnica do OpenGL seria o *picking* (modo de selecção).