

Sistemas Gráficos e Interacção

Época de Recurso

2018-02-06

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

20%

- a. **[2.5]** Qual a dimensão em bytes de um *frame buffer* RGBA de 1024 x 1024 x 16 bits?
- i. 0.5 Megabyte
 - ii. 1 Megabyte
 - ☒ iii. 2 Megabyte
 - iv. Nenhuma das anteriores
- b. **[2.5]** Qual das seguintes matrizes representa em coordenadas homogêneas o ponto (2, -3, 4)?
- i. $[2.0, -3.0, 4.0, 0.0]^T$
 - ii. $[4.0, -6.0, 8.0, 1.0]^T$
 - ☒ iii. $[8.0, -12.0, 16.0, 4.0]^T$
 - iv. Nenhuma das anteriores
- c. **[2.5]** Qual das seguintes transformações compostas é rígida?
- i. `glTranslated(1.0, 2.0, 3.0); glScaled(-1.0, -2.0, -3.0);`
 - ii. `glRotated(0.0, 0.0, 0.0, 1.0); glScaled(0.0, 0.0, -1.0);`
 - ☒ iii. `glScaled(1.0, 2.0, 4.0); glScaled(1.0, 0.5, 0.25);`
 - iv. Nenhuma das anteriores
- d. **[2.5]** Considere o objecto delimitado pela superfície descrita pela equação $x^2 + y^2 + z^2 - 1 = 0$. O ponto de coordenadas (0.4, 0.6, 0.8) encontra-se
- i. No interior do objecto
 - ii. Na fronteira do objecto
 - ☒ iii. No exterior do objecto
 - iv. Nenhuma das anteriores

- e. **[2.5]** Qual a representação associada ao objecto referido na alínea anterior?
- i. Paramétrica
 - ii. Implícita
 - iii. CSG
 - iv. Nenhuma das anteriores
- f. **[2.5]** Uma forma de determinar o vector normal a um polígono planar consiste em
- i. Calcular o produto escalar dos vectores definidos por duas arestas do polígono e dividir o resultado obtido pelo somatório dos comprimentos dos vectores
 - ii. Projectar o polígono nos planos OYZ, OZX e OXY e calcular as áreas dos polígonos resultantes; as componentes da normal serão proporcionais a estes valores
 - iii. As respostas i. e ii.
 - iv. Nenhuma das anteriores
- g. **[2.5]** Em OpenGL um objecto muito polido pode ser simulado usando um material
- i. Que reflecte significativamente a componente de luz difusa
 - ii. Com uma elevada componente de emissão
 - iii. Com um coeficiente de especularidade elevado
 - iv. Nenhuma das anteriores
- h. **[2.5]** No mapeamento de texturas em OpenGL, o processo de filtragem designado por `GL_NEAREST_MIPMAP_LINEAR`
- 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 Interação

Época de Recurso

2018-02-06

N.º _____ Nome _____

Parte Teórico-Prática

30%

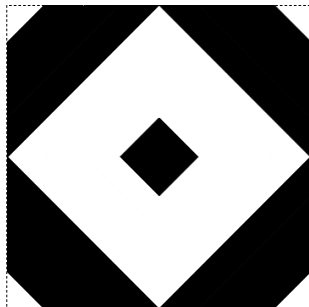
Resolução: No próprio enunciado

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

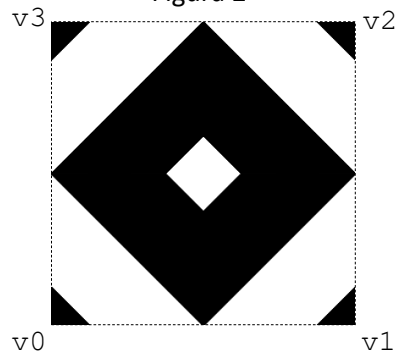
- a. **[3.0]** Aplique a textura apresentada na Figura 1 a um quadrado, de modo a ficar com o aspecto apresentado na Figura 2.

Figura 1



```
glTexCoord2f(0.5, 0.5);
glVertex3fv(v0);
glTexCoord2f(1.5, 0.5);
glVertex3fv(v1);
glTexCoord2f(1.5, 1.5);
glVertex3fv(v2);
glTexCoord2f(0.5, 1.5);
glVertex3fv(v3);
```

Figura 2



```
// ou qualquer outro mapeamento em que
// as coordenadas terminem em 0.5 e
// tenham dimensão 1.0 por 1.0
```

- b. **[3.0]** Pretende-se modelar um terreno a partir de uma matriz que contém o valor da cota em vários pontos do terreno. Que tipo de polígonos e qual das primitivas de desenho do OpenGL serão mais indicados para fazer a modelação?

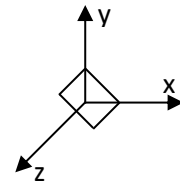
Tipo de polígonos: triângulo

Primitiva de desenho: GL_TRIANGLE_STRIP

- c. **[3.0]** Considerando as definições por omissão do OpenGL, pretende-se definir a normal para o quadrilátero desenhado pelo seguinte extracto de código. Qual a normal unitária perpendicular ao quadrilátero?

Nota: Pode usar funções trigonométricas com ângulos expressos em graus ou em radianos.

```
glBegin(GL_QUADS);
    glNormal3f(-sin(45.0), -sin(45.0), 0.0);
    // ou (-cos(45.0), -cos(45.0), 0.0)
    // ou (-sqrt(2.0) / 2.0, -sqrt(2.0) / 2.0, 0.0)
    glVertex3f(1, 0, 0);
    glVertex3f(1, 0, 1);
    glVertex3f(0, 1, 1);
    glVertex3f(0, 1, 0);
glEnd();
```



- d. **[3.0]** Pretende-se simular uma câmara montada num helicóptero a olhar directamente para baixo. O *up* da câmara está alinhado com a direcção em que o helicóptero segue. A posição do helicóptero é dada por `modelo.x`, `modelo.y` e `modelo.z`, e a direcção em que o helicóptero está a seguir é dada por `modelo.dir`. Complete a instrução seguinte de modo a obter o resultado pretendido, considerando como eixo vertical o eixo dos Z (positivo para cima).

```
gluLookAt(modelo.x, modelo.y, modelo.z,
           modelo.x, modelo.y, modelo.z - 1.0,
           cos(modelo.dir), sin(modelo.dir), 0.0);
```

- e. Considere o objecto representado na Figura 3 e a existência da função `caixa()` que desenha um cubo com 1 unidade de lado, centrado na origem.

As dimensões dos elementos são L_x , A_x e P_x , em que x designa o nome do elemento.

Considere que:

- O elemento A se desloca linearmente sobre o plano XZ;
- O elemento B roda em torno do ponto médio do topo do elemento A;
- O elemento C roda em torno do ponto indicado em relação ao elemento B;
- O elemento D roda em torno do **seu centro**, em relação ao elemento C.

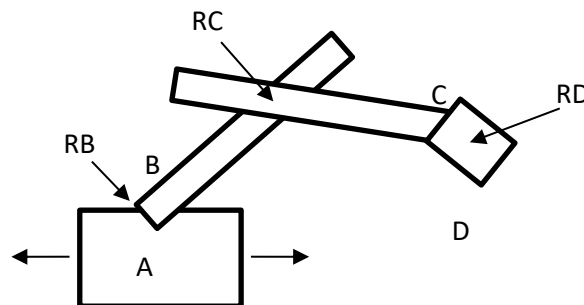


Figura 3

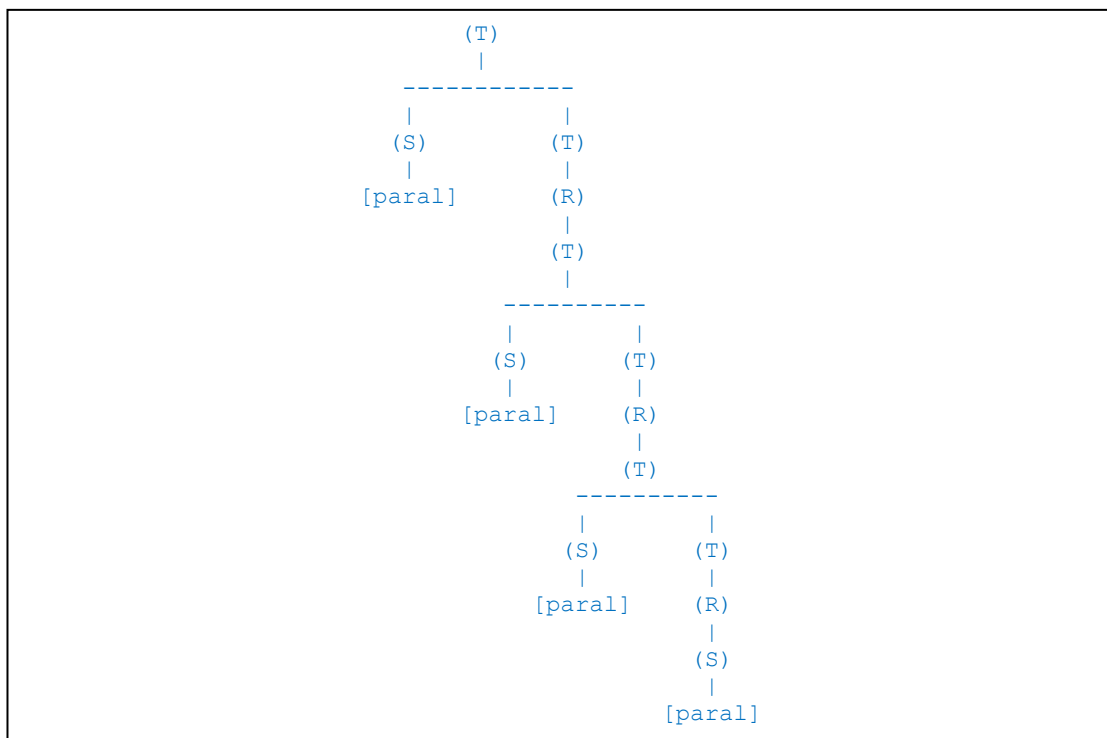
- i. **[4.0]** Construa a árvore de cena do objecto apresentado na Figura 3. Não se esqueça de colocar as transformações que garantam o movimento dos elementos A, B, C e D.

Sistemas Gráficos e Interação

Época de Recurso

2018-02-06

N.º _____ Nome _____



- ii. **[4.0]** Pretende-se controlar a rotação dos elementos **B** e **C** com as teclas do cursor (GLUT_KEY_UP e GLUT_KEY_DOWN). Quando o ângulo do elemento **B** aumenta, o ângulo do elemento **C** diminui na mesma quantidade. Complete o código que se segue para implementar o controlo do movimento pretendido, directamente no *callback* `glutSpecialFunc()`. Considere que os limites de rotação do elemento **B** são 0°(horizontal) e 90°(vertical). Use as constantes e variáveis que entender relevantes.

```

void SpecialKey(int key, int x, int y)
{
    switch(key) {
        case GLUT_KEY_UP:
            if(modelo.rotacaoB < 90.0) {
                modelo.rotacaoB += INC_ROTACAO;
                modelo.rotacaoC -= INC_ROTACAO;
            }
            break;
        case GLUT_KEY_DOWN:
            if(modelo.rotacaoB > 0.0) {
                modelo.rotacaoB -= INC_ROTACAO;
                modelo.rotacaoC += INC_ROTACAO;
            }
            break;
    }
}

```