

Época Normal

02-02-2012

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]** Um gráfico descrito no formato **BMP (Bitmap)**
- i. Constitui um exemplo de uma representação matricial
  - ii. Pode ser livremente rodado ou escalado sem perda significativa de precisão
  - iii. Pode ser convertido numa representação vectorial com base numa operação de rasterização
  - iv. Todas as anteriores
- b. **[2.5]** Num sistema gráfico dotado de um *frame buffer* RGB de 512 x 512 píxeis, 8 bits/píxel
- i. É permitida a reprodução de imagens compostas por mais do que 1 milhão de píxeis
  - ii. Cada píxel é descrito por 3 bits para a componente vermelha, 3 bits para a verde e 2 bits para a azul, num total de 8 bits
  - iii. É permitida a reprodução de imagens com 256 níveis de vermelho, 256 níveis de verde e 256 níveis de azul
  - iv. Nenhuma das anteriores
- c. **[2.5]** Se se pretender reflectir um objecto no plano  $z = 0$  deve-se efectuar a seguinte transformação
- i. `glTranslated(0.0, 0.0, -1.0);`
  - ii. `glRotated(180.0, 0.0, 0.0, 1.0);`
  - iii. `glScaled(1.0, 1.0, -1.0);`
  - iv. Todas as anteriores
- d. **[2.5]** Considere duas transformações genéricas, representadas pelas matrizes  $T1$  e  $T2$ . A transformação associada à matriz  $T2 \times T1$  traduz a composição das transformações
- i.  $T1$  seguida de  $T2$
  - ii.  $T2$  seguida de  $T1$
  - iii. A ordem das transformações é irrelevante
  - iv. Nenhuma das anteriores

- e. **[2.5]** Para obter a transformação **perspectiva** de um ponto 3D
- i. Basta efectuar uma multiplicação por uma determinada matriz de  $3 \times 3$
  - ii. Basta efectuar uma multiplicação por uma determinada matriz de  $4 \times 4$
  - iii. É necessário efectuar uma operação de divisão perspectiva (divisão por  $w$ )
  - iv. Nenhuma das anteriores
- f. **[2.5]** Em modelação, a técnica de codificação conhecida por **Winged-Edge**
- i. Constitui um exemplo da representação por fronteira (*B-rep*)
  - ii. Armazena informação numa estrutura associada às arestas
  - iii. Permite determinar em tempo constante todos os 9 tipos de adjacência de vértices, arestas e faces
  - iv. Todas as anteriores
- g. **[2.5]** A contribuição dada pela componente de **emissão** do modelo de Phong
- i. Modela o efeito da reflexão da luz por outros objectos do ambiente
  - ii. Simula a reflexão da luz por objectos constituídos por materiais baços ou foscos
  - iii. Simula a reflexão por objectos altamente polidos
  - iv. Nenhuma das anteriores
- h. **[2.5]** Uma função de mapeamento de textura
- i. Devolve, para cada ponto do espaço de textura, o ponto correspondente do objecto
  - ii. Devolve, para cada ponto do espaço de textura, a cor (ou outra propriedade) que se pretende atribuir ao ponto correspondente do objecto
  - iii. Não pode basear-se na descrição paramétrica da superfície do objecto ao qual a textura está a ser aplicada
  - iv. Todas as anteriores

Época Normal

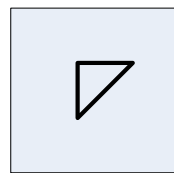
02-02-2012

N.º \_\_\_\_\_ Nome \_\_\_\_\_

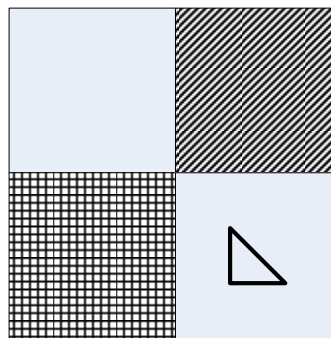
**Parte Teórico-Prática**
**40%**

**Nota:** Em todas as perguntas, a menos que algo seja dito em contrário, assuma a posição da câmara por omissão e o semieixo positivo dos ZZ como representando a cota positiva.

- a. **[2.0]** Suponha que pretende aplicar uma textura a um quadrado por forma a ficar como na figura a), usando a imagem de textura b). Complete o seguinte extracto de código.



(a)



(b)

```
glBegin(GL_QUADS);
  glTexCoord2f(_____, _____);
  glVertex3f(-2.0, -1.0, 0.0);
  glTexCoord2f(_____, _____);
  glVertex3f(-2.0, 1.0, 0.0);
  glTexCoord2f(_____, _____);
  glVertex3f(0.0, 1.0, 0.0);
  glTexCoord2f(_____, _____);
  glVertex3f(0.0, -1.0, 0.0);
glEnd();
```

- b. [2.0] Suponha que pretende uma vista de topo da cena que mostre a totalidade do seu mundo (centrado em 0.0, 0.0, 0.0). Sabendo que a distância da câmara é representada por *CAMERA\_DISTANCE*, como deverá configurá-la?

```
gluLookAt ( _____, _____, _____,
            _____, _____, _____,
            _____, _____, _____ ) ;
```

- c. [2.0] Considere uma cena composta por uma única fonte de luz, definida unicamente pela instrução `glLightfv(GL_LIGHT0, GL_AMBIENT, lamb)`, em que `lamb` é o *array* {1.0, 0.5, 0.8, 1.0}. Se o resultado final pretendido no ecrã for R = 0.0, G = 0.5, B = 0.2, qual o valor de `mamb` que deverá ser passado à instrução `glMaterial(GL_FRONT_AND_BACK, GL_AMBIENT, mamb)`?

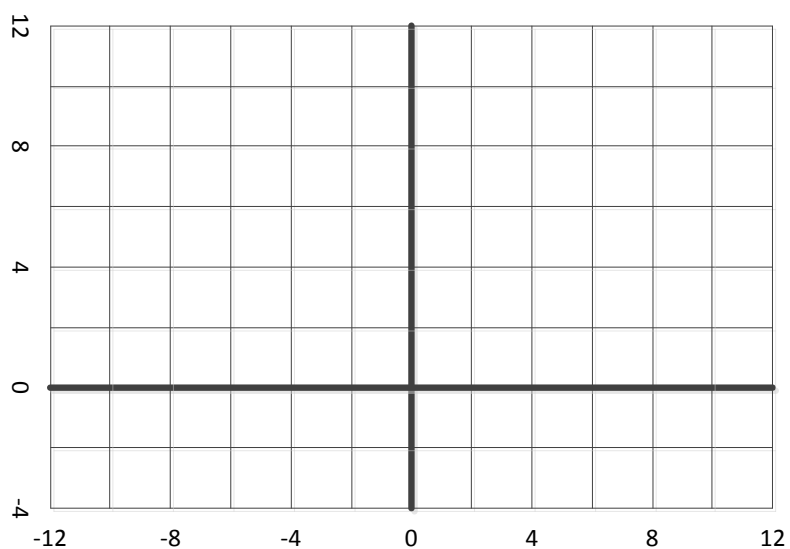
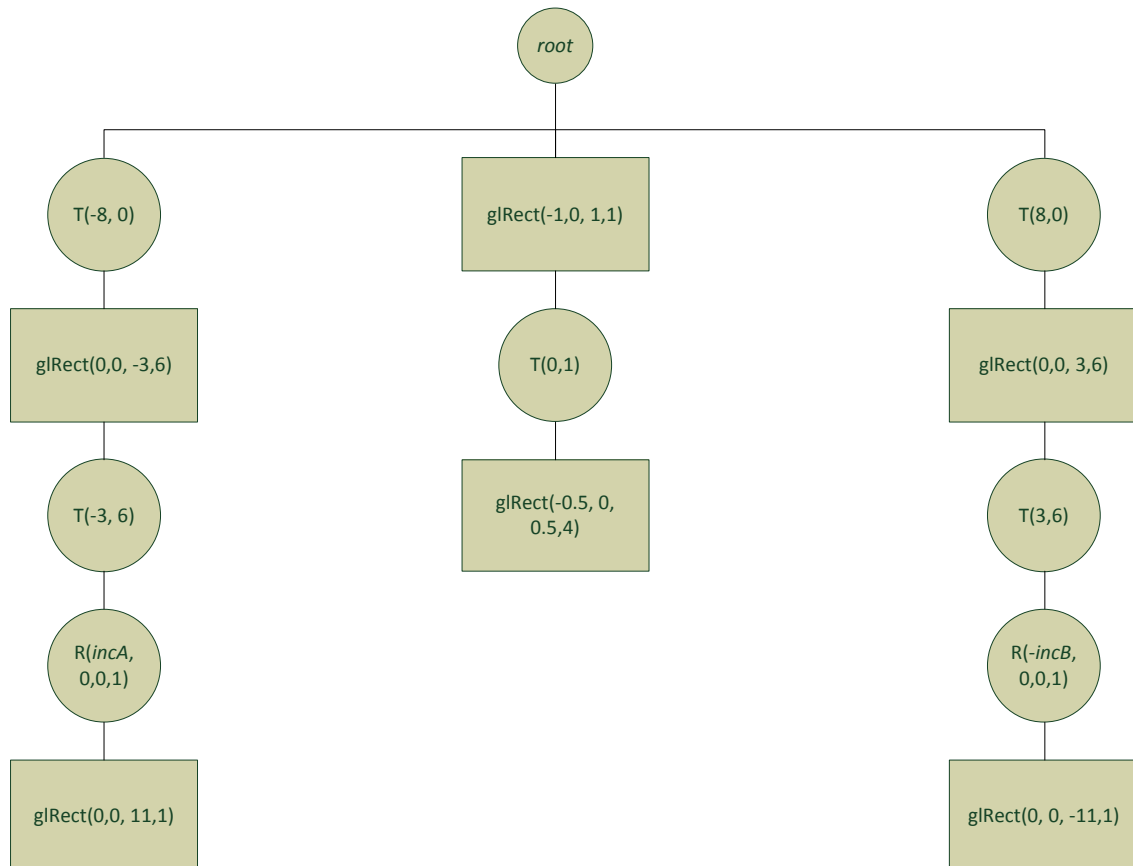
```
GLfloat mamb = { _____, _____, _____, _____ } ;
```

Época Normal

02-02-2012

N.º \_\_\_\_\_ Nome \_\_\_\_\_

- d. [6.0] Assuma que uma determinada cena é desenhada de acordo com a árvore de cena ilustrada na figura. Reproduza a cena, assumindo que  $incA = 0.0$  e  $incB = -5.0$ .





02-02-2012

N.º \_\_\_\_\_ Nome \_\_\_\_\_

- e. **[2.0]** Preencha o seguinte *array* de modo a definir uma luz com uma direcção de  $45^\circ$  em relação ao plano XY e coplanar com o plano XZ.

```
float lpos[] = {_____, _____, _____, _____};
glLigthfv(GL_LIGTH0, GL_POSITION, lpos);
```

- f. **[3.0]** Considere uma cena composta por um dado objecto. Assumindo a existência de uma variável denominada `modelo.inclinação`, escreva o código que permite controlar com o teclado a inclinação desse objecto.

```
void Keyboard(unsigned char key, int x, int y)
{
```

[illegible]

```
// redesenhar o ecran
```

```
glutPostRedisplay();
```

}

g. [3.0] Esboce o desenho que resulta do seguinte extracto de código.

```
glBegin(GL_LINES);  
    glVertex2f(-1.0, -1.0);  
    glVertex2f(2.0, 1.0);  
    glVertex2f(2.0, 2.0);  
    glVertex2f(-3.0, 2.0);  
    glVertex2f(4.0, 2.0);  
glEnd();
```

