

Sistemas Gráficos e Interação

Época Especial

2012-09-06

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

30%

- a. [2.5] Os formatos **BMP** e **JPEG** de representação de gráficos constituem exemplos de
- Representações matriciais
 - Representações vectoriais
 - Uma representação matricial e uma vectorial, respectivamente
 - Uma representação vectorial e uma matricial, respectivamente
- b. [2.5] Num sistema gráfico dotado de um *frame buffer* RGB de 1024 x 1024 píxeis, 8 bits/píxel
- Não é possível a reprodução de imagens compostas por mais do que 1 milhão de píxeis
 - Cada píxel é descrito por 8 bits para a componente vermelha, 8 bits para a verde e 8 bits para a azul
 - É possível a reprodução de imagens com 16 níveis de vermelho, 16 níveis de verde e 16 níveis de azul
 - Nenhuma das anteriores
- c. [2.5] A qual das seguintes sequências de transformações corresponde uma matriz de transformação composta igual à matriz identidade?
- `glTranslated(1.0, 2.0, 3.0); glTranslated(3.0, 2.0, 1.0);`
 - `glRotated(45.0, 0.0, 0.0, 1.0); glRotated(-45.0, 0.0, 1.0, 0.0);`
 - `glScaled(2.0, 4.0, 8.0); glScaled(0.5, 0.25, 0.125);`
 - Nenhuma das anteriores
- d. [2.5] Admita que pretende aplicar a um objecto a seguinte sequência de transformações: uma translação $T1$, seguida de uma rotação R , seguida de uma translação $T2$. Qual das seguintes matrizes traduz a referida composição de transformações?
- $T1 \times T2 \times R$
 - $T1 \times R \times T2$
 - $R \times T2 \times T1$
 - $T2 \times R \times T1$

- e. **[2.5]** Em OpenGL, para obter uma projecção em perspectiva associada a um volume de visualização assimétrico, deverá recorrer-se à instrução
- i. `glOrtho()`
 - ii. `glFrustum()`
 - iii. `gluPerspective()`
 - iv. Nenhuma das anteriores
- f. **[2.5]** Nas representações de objectos por células
- i. As grelhas dividem o espaço em cubos de igual dimensão
 - ii. As *octrees* dividem o espaço em cubos cujos lados são potências de 2
 - iii. As *BSP-trees* dividem o espaço em poliedros convexos
 - iv. Todas as anteriores
- g. **[2.5]** No modelo de iluminação do OpenGL o vector *halfway* é usado para calcular de uma forma simplificada
- i. A componente ambiente de iluminação
 - ii. As componentes ambiente e difusa de iluminação
 - iii. As componentes ambiente, difusa e especular de iluminação
 - iv. Nenhuma das anteriores
- h. **[2.5]** A técnica de *mipmapping* de mapeamento de texturas
- i. É suportada pelo OpenGL
 - ii. Permite que texturas de diferentes níveis de resolução sejam aplicadas de forma adaptativa
 - iii. Reduz os efeitos de discretização que decorrem da interpolação
 - iv. Todas as anteriores

Sistemas Gráficos e Interacção

Época Especial

2012-09-06

N.º _____ Nome _____

Parte Teórico-Prática

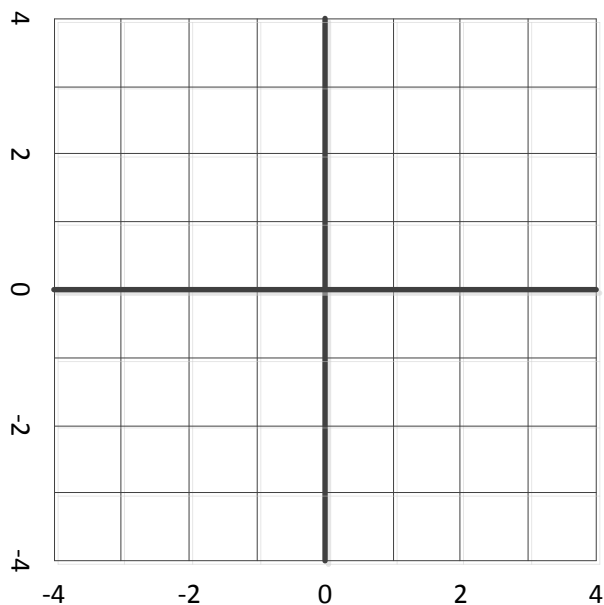
40%

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. **[1.5]** Admita que pretende definir uma vista na **terceira pessoa** (atrás do personagem, acompanhando o seu movimento). Qual dos seguintes raciocínios é correcto?
- i. Com base na orientação do personagem e na sua posição (x, y) , calcular a posição da câmara adicionando a x e a y a distância da câmara
 - ii. Apontar a câmara para o personagem e calcular a sua posição com base na orientação oposta à do personagem.
 - iii. Colocar a câmara numa posição x, y, z fixa, a apontar para o personagem
 - iv. Calcular a posição da câmara subtraindo à posição do personagem a distância da câmara
 - v. Colocar a câmara na mesma posição do personagem, a apontar para o ponto para o qual o personagem está a olhar
- b. **[3.0]** Qual a diferença entre uma luz posicional e uma luz direcciona? Indique como, em OpenGL, distingue uma da outra.

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

```
glBegin(GL_LINE_LOOP);
    glVertex2d(-1.0, -1.0);
    glVertex2d(2.0, 1.0);
    glVertex2d(2.0, 2.0);
    glVertex2d(-3.0, 1.0);
glEnd();
```



d. **[1.5]** Admita que, num programa escrito em OpenGL, as alturas estão representadas no semieixo positivo dos XX, o solo no plano YZ e a câmara se encontra na posição por omissão. Qual das seguintes afirmações é correcta?

- i. A câmara encontra-se posicionada ao nível do solo
- ii. A câmara apresenta uma vista de topo
- iii. Apenas os objectos “afundados” no solo são visíveis
- iv. A câmara apresenta uma vista de baixo
- v. Nada é visível, pois é sempre necessário efectuar `gluLookAt()` antes de se visualizar a cena

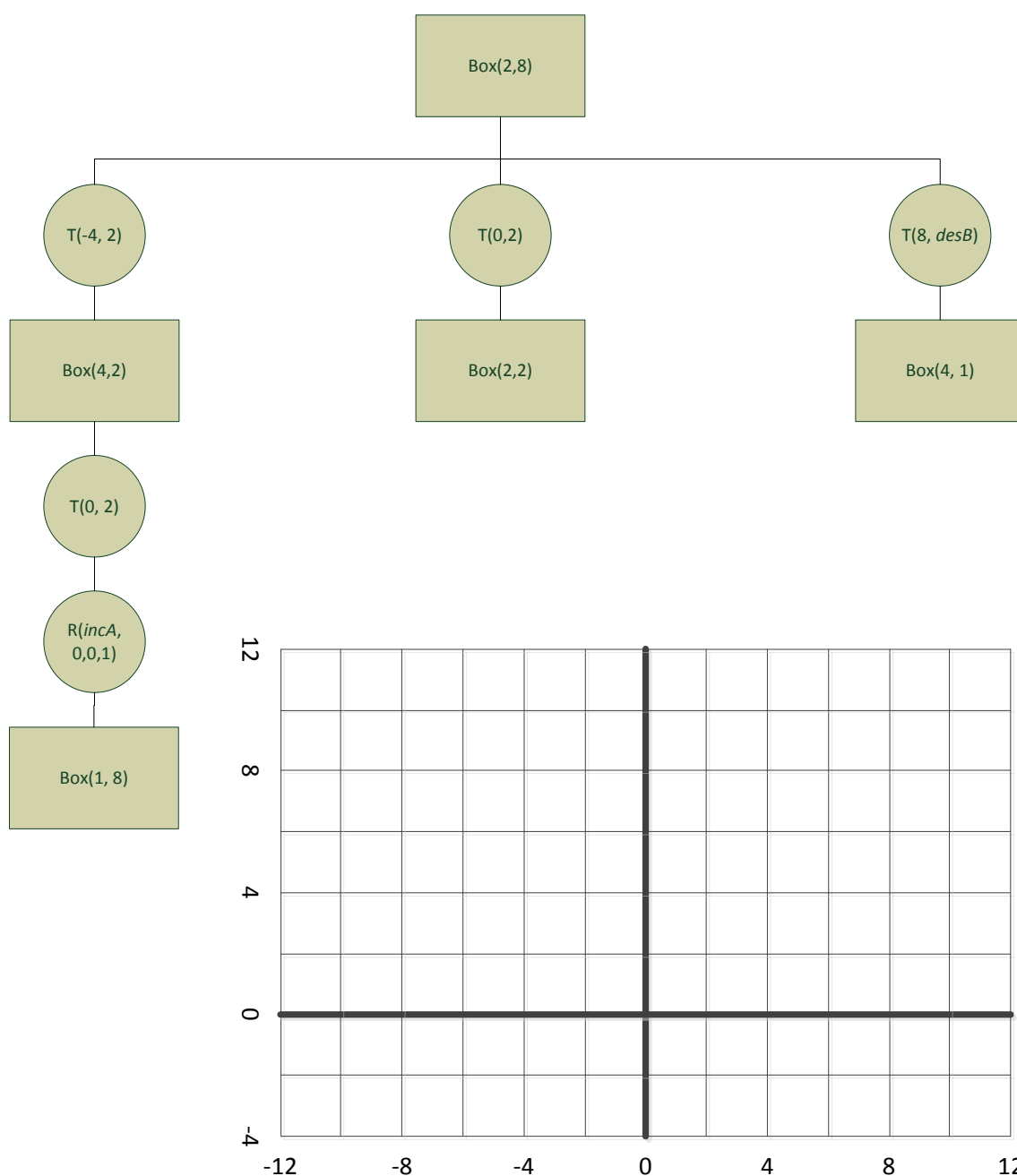
Sistemas Gráficos e Interação

Época Especial

2012-09-06

N.º _____ Nome _____

- e. **[5.0]** Assuma que uma determinada cena é desenhada de acordo com a árvore de cena ilustrada na figura. A função $\text{Box}(A, C)$ desenha um retângulo de altura A e comprimento C , a partir do canto inferior esquerdo. Reproduza a cena no plano XY, considerando as alturas no semieixo positivo dos YY, e assumindo que $\text{incA} = 5.0$ e $\text{desB} = 1.0$.



Sistemas Gráficos e Interação

Época Especial

2012-09-06

N.º _____ Nome _____

- f. **[3.0]** Considere uma cena composta por um único objecto, desenhado na origem. Pretende-se realizar uma animação que permita circundar o objecto ao longo do plano do solo (o plano YZ). Escreva o código que permite efectuar a referida animação, definindo as variáveis de modelo que entender necessárias.

```
struct modelo_t {  
{
```

```
    _____  
    _____  
} modelo;
```

```
void Display()  
{
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    glLoadIdentity();
```

```
    _____  
    _____  
    _____  
    DesenhaObjecto();
```

```
    glFlush();
```

```
}
```

```
void Timer(int value)  
{
```

```
    glutTimerFunc(100, Timer, 0);
```

```
    _____  
    _____
```

```
// redesenhar o ecrã
```

```
glutPostRedisplay();
```

```
}
```

g. **[3.0]** A diferença entre o modo *selection* e o modo *feedback* é a seguinte:

- i. No modo *selection* especificamos os objectos que pretendemos seleccionar com o rato; no modo *feedback* especificamos os objectos que pretendemos imprimir
- ii. O modo *selection* indica os objectos que intersectam o volume de visualização; o modo *feedback* indica os objectos que não intersectam o volume de visualização
- iii. O modo *selection* devolve os nomes dos objectos que intersectam o volume de visualização; o modo *feedback* devolve a geometria dos objectos que intersectam o volume de visualização
- iv. O modo *selection* obriga a desenhar (*render*) os objectos duas vezes no ecrã para obter a informação; o modo *feedback*, não