

Sistemas Gráficos e Interação

Época de Recurso

2013-02-16

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]** Na representação de um vector 2D em coordenadas homogéneas são usadas
- i. Apenas duas componentes: x e y
 - ii. Três componentes: x , y e w , em que $w \neq 0$
 - iii. Quatro componentes: x , y , z e w , em que $w = 1$
 - iv. Nenhuma das anteriores
- b. **[2.5]** A qual das seguintes sequências de transformações corresponde uma matriz de transformação composta igual à matriz identidade?
- i. `glTranslated(1.0, 2.0, 3.0); glTranslated(-1.0, -2.0, -3.0);`
 - ii. `glRotated(45.0, 0.0, 1.0, 0.0); glRotated(-45.0, 0.0, 1.0, 0.0);`
 - iii. `glScaled(2.0, 4.0, 8.0); glScaled(0.5, 0.25, 0.125);`
 - iv. Todas as anteriores
- c. **[2.5]** As transformações projectivas
- i. Obrigam a que o centro de projecção esteja sempre posicionado na origem
 - ii. São incompatíveis com o uso de coordenadas homogéneas
 - iii. Não preservam as combinações afins
 - iv. Todas as anteriores
- d. **[2.5]** A equação $(x + 1)^2 + (y - 2)^2 + (z + 3)^2 = 4$ constitui
- i. Uma descrição implícita de uma superfície esférica
 - ii. Uma parametrização de uma superfície cilíndrica
 - iii. Uma descrição implícita de uma superfície cónica
 - iv. Nenhuma das anteriores

- e. **[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 um conjunto de coordenadas tal que $w \neq 0$
 - iii. Especificar para o ângulo de *cutoff* um valor compreendido entre 0° e 90°
 - iv. Todas as anteriores
- f. **[2.5]** A contribuição dada pela componente de iluminação especular do modelo de Phong
 - i. Não depende da geometria do objecto iluminado
 - ii. Não depende da posição da fonte de luz
 - iii. Não depende da posição do observador
 - iv. Nenhuma das anteriores
- g. **[2.5]** Uma função de mapeamento de texturas
 - i. Devolve o valor de uma determinada propriedade – cor, por exemplo – correspondente a cada ponto do espaço de textura
 - ii. Devolve o ponto do objecto correspondente a cada ponto do espaço de textura
 - iii. Devolve um apontador para uma lista dos objectos aos quais foram aplicados essas texturas
 - iv. Nenhuma das anteriores
- h. **[2.5]** No domínio comportamental de desenvolvimento de interfaces com o utilizador
 - i. Não é adoptado o ponto de vista do sistema
 - ii. Estão envolvidos aspectos tais como os factores humanos e as especificações de usabilidade
 - iii. A definição de algoritmos e estruturas de dados é irrelevante face a outros aspectos
 - iv. Todas as anteriores

Sistemas Gráficos e Interação

Época de Recurso

2013-02-16

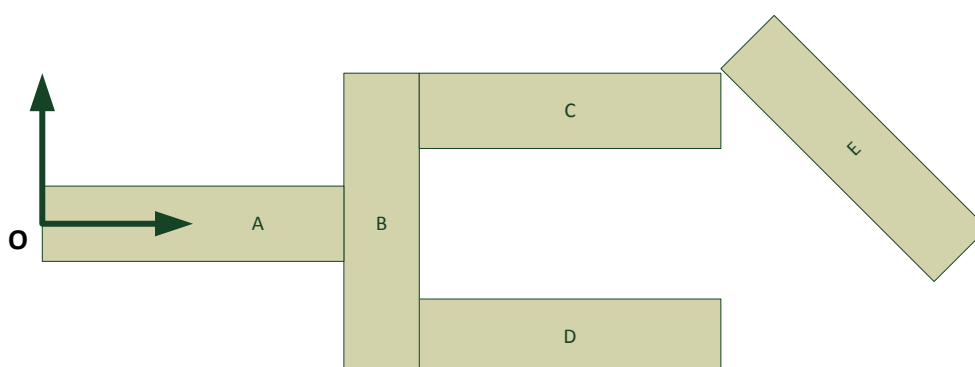
N.º _____ Nome _____

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. **[4.0]** Construa a árvore de cena do objecto representado na figura, sabendo que dispõe apenas de uma função `box()` que desenha um rectângulo com 4 unidades de largura e 1 unidade de altura, **centrado na origem**. Tenha em consideração que o ponto **O** na figura representa a origem do sistema de eixos coordenados.



Sistemas Gráficos e Interacção

Época de Recurso

2013-02-16

N.º _____ Nome _____

- b. **[3.0]** Pretende-se controlar o elemento **E** da alínea anterior com as teclas ‘a’ e ‘z’, respectivamente para “abrir” (o elemento **E** fica na horizontal, paralelo a **C**) e “fechar” (o elemento **E** fica na vertical, perpendicular a **C**) o objecto. Assuma que a função `DesenhaObjecto()` desenha correctamente o elemento **E**, tendo em conta os valores correntes das variáveis de modelo. Complete o código que se segue onde entender necessário.

```
struct Modelo {
```

```
    ...
```

```
} modelo;
```

```
void Display()
```

```
{
```

```
    glClear(GL_COLOR_BUFFER_BIT);
```

```
    glLoadIdentity();
```

```
    DesenhaObjecto();
```

```
    glFlush();
```

```
}
```

```
void Key(unsigned char key, int x, int y)
```

```
{
```



```
}
```

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

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

```
    // redesenhar o ecrã
```

```
    glutPostRedisplay();
```

```
}
```

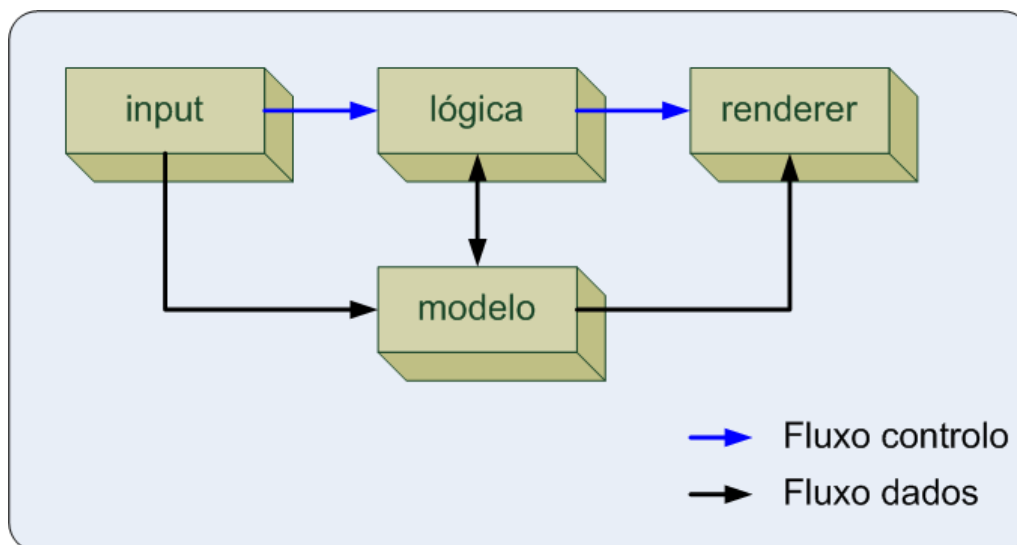
Sistemas Gráficos e Interação

Época de Recurso

2013-02-16

N.º _____ Nome _____

c. Considere a arquitectura simplificada de um programa gráfico representada na figura.



i. **[3.0]** Explique sucintamente a interação dos blocos *input*, *lógica* e *renderer* com o bloco *modelo*.

ii. **[3.0]** Indique que *callbacks* do GLUT estão relacionadas com cada bloco.

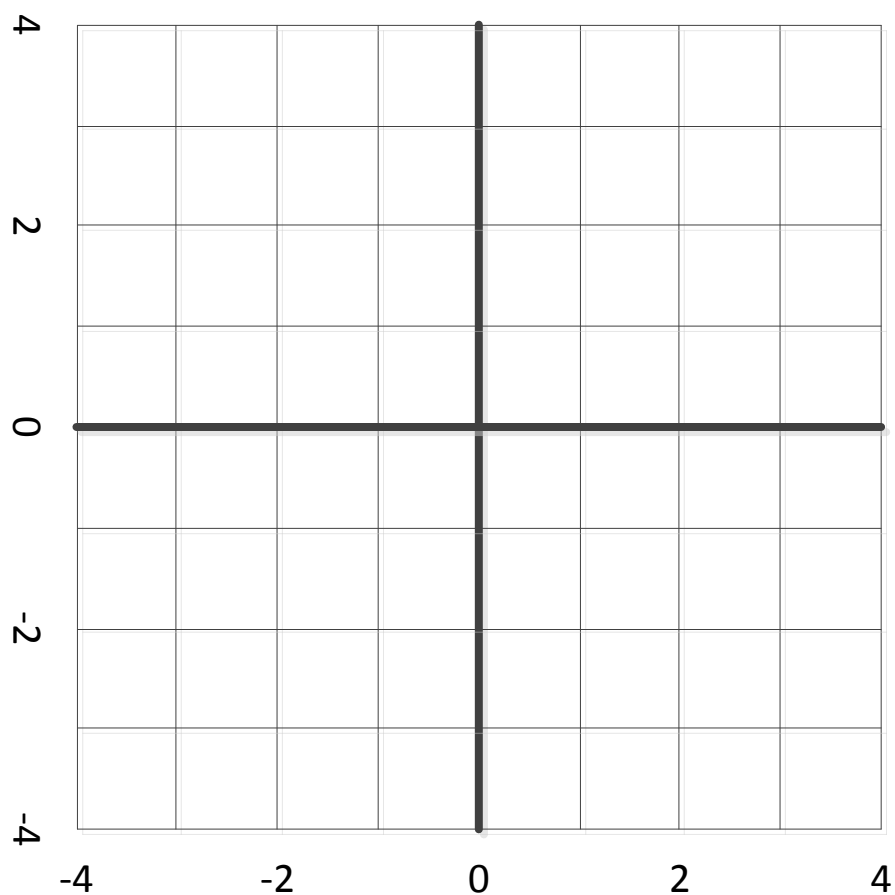
Input: _____

Lógica: _____

Renderer: _____

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

```
glBegin(GL_QUADS);  
  glVertex2d(-1.0, -1.0);  
  glVertex2d(2.0, 1.0);  
  glVertex2d(0.0, 1.0);  
  glVertex2d(-1.0, 0.0);  
  glVertex2d(-2.0, 1.0);  
glEnd();
```



Sistemas Gráficos e Interacção

Época de Recurso

2013-02-16

N.º _____ Nome _____

- e. **[4.0]** Admita que se pretende definir uma vista de cima, centrada num personagem, de modo que, quando este se desloca na direcção dos 0° , deverá observar-se o efeito da cena a deslocar-se da direita para a esquerda. Assuma a existência das variáveis de modelo `obj.x`, `obj.y` e `obj.dir`. Quais os cálculos a efectuar para posicionar a câmara na cena com a função `gluLookAt()`?

`float eyex =` _____

`float eyey =` _____

`float eyez =` _____

`float centerx =` _____

`float centery =` _____

`float centerz =` _____

`float upx =` _____

`float upy =` _____

`float upz =` _____

`gluLookAt(eyex, eyey, eyez, centerx, centery, centerz, upx, upy,
upz);`