

## Sistemas Gráficos e Interacção

Época de Recurso

2014-02-12

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]** A linguagem PostScript de descrição de páginas confere a um dispositivo que a interprete a aparência de um dispositivo
- i. Matricial
  - ii. Tensorial
  - iii. Vectorial
  - iv. Nenhuma das anteriores
- b. **[2.5]** A impressão de um documento descrito na linguagem PostScript numa vulgar impressora de jacto de tinta
- i. Não é possível
  - ii. É possível mas requer uma operação prévia de rasterização
  - iii. É possível mas requer uma operação prévia de reconhecimento de padrões
  - iv. Nenhuma das anteriores
- c. **[2.5]** Na representação de um ponto 3D em coordenadas homogéneas são usadas
- i. Apenas três componentes:  $x$ ,  $y$  e  $z$
  - ii. Quatro componentes:  $x$ ,  $y$ ,  $z$  e  $w$ , em que  $w = 0$
  - iii. Quatro componentes:  $x$ ,  $y$ ,  $z$  e  $w$ , em que  $w \neq 0$
  - iv. Nenhuma das anteriores
- d. **[2.5]** A transformação que resulta da composição das transformações `glTranslated(1.0, 2.0, 3.0); glRotated(45.0, 0.0, 0.0, 1.0); glScaled(3.0, 2.0, 1.0);`
- i. É a transformação identidade
  - ii. É uma transformação rígida
  - iii. É uma transformação perspectiva
  - iv. Nenhuma das anteriores

- e. **[2.5]** Na representação de sólidos por *octrees*
  - i. Há ambiguidade, pois a uma mesma representação podem corresponder vários modelos
  - ii. O espaço é dividido em cubos de igual dimensão
  - iii. O espaço é dividido em cubos cujos lados são potências de base 2
  - iv. Nenhuma das anteriores
- f. **[2.5]** A contribuição dada pela componente especular do modelo de iluminação do OpenGL
  - i. Não é compatível com as fontes de luz direccionais
  - ii. É característica das superfícies polidas
  - iii. Depende do ângulo de incidência da luz e não depende da posição do observador
  - iv. Nenhuma das anteriores
- g. **[2.5]** A contribuição dada pela componente de emissão do modelo de iluminação do OpenGL
  - 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 o fenómeno da fluorescência
  - iv. Nenhuma das anteriores
- h. **[2.5]** A técnica de *mipmapping* de mapeamento de texturas
  - i. Não é suportada pelo OpenGL
  - ii. Permite que texturas de diferentes níveis de resolução sejam aplicadas de forma adaptativa
  - iii. Não é compatível com as parametrizações esféricas
  - iv. Nenhuma das anteriores

## Sistemas Gráficos e Interacção

Época de Recurso

2014-02-12

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 e o semieixo positivo dos ZZ como representando a cota positiva

- a. **[2.0]** A arquitectura genérica de um programa gráfico prevê quatro componentes. Explique o relacionamento entre a componente de lógica e o *renderer*.

---

---

---

---

---

---

- b. **[3.0]** De que informação necessita para poder ter uma câmara polar na sua cena?

---

---

```
typedef struct camara_t  
{
```

---

---

---

---

---

---

```
} camara_t;
```



## Sistemas Gráficos e Interação

Época de Recurso

2014-02-12

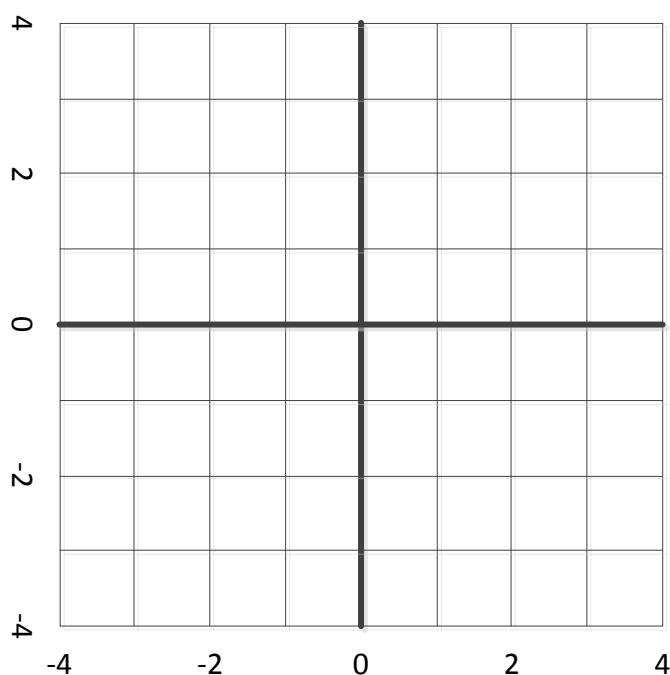
N.º \_\_\_\_\_ Nome \_\_\_\_\_

c. **[2.0]** Se pretender iluminar a sua cena com uma luz fixa – tal como a proveniente de um candeeiro de iluminação pública – e uma luz móvel – uma lanterna portátil, por exemplo – irá necessitar de

- i. Configurar luz ambiente e uma fonte de luz posicional
- ii. Configurar luz ambiente e uma fonte de luz direcciona
- iii. Configurar duas fontes de luz posicional
- iv. Configurar duas fontes de luz direcciona
- v. Configurar uma fonte de luz posicional e outra direcciona

d. **[2.0]** Esboce o desenho que resulta do seguinte extracto de código

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





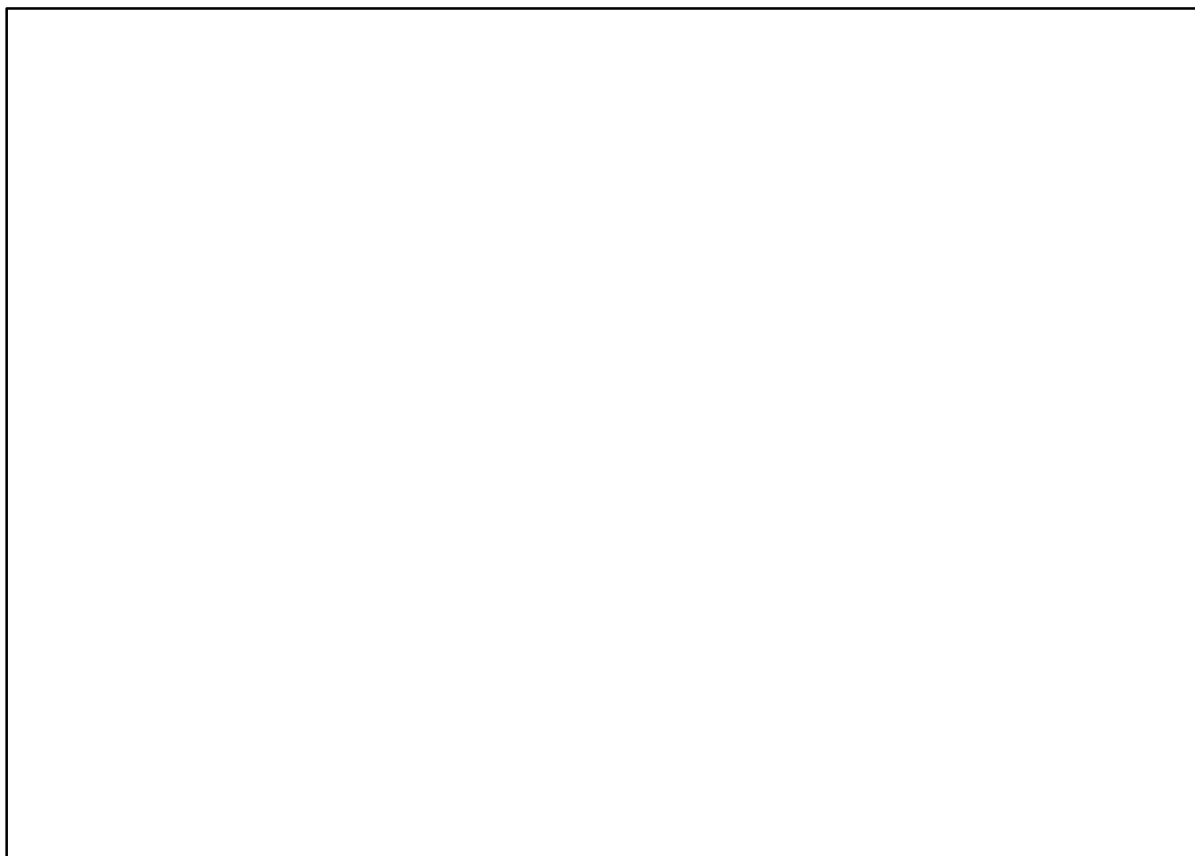
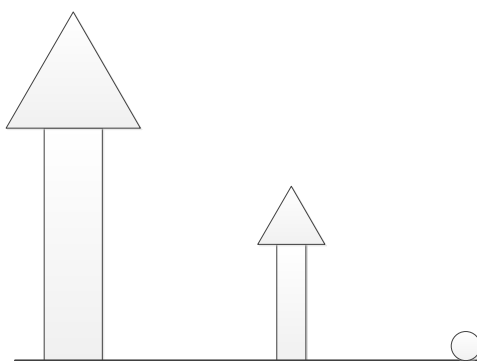
## Sistemas Gráficos e Interação

Época de Recurso

2014-02-12

N.º \_\_\_\_\_ Nome \_\_\_\_\_

- e. **[5.0]** Construa a árvore de cena do objecto representado na figura. Não se esqueça de indicar as transformações e os respectivos parâmetros. Assuma a existência das funções `box()` e `triang()`, as quais desenhavam respectivamente o tronco e a copa da árvore grande, centrados na origem. A árvore pequena tem 50% da dimensão da árvore grande. O círculo é desenhado pela função `bola()`, centrado na origem. A distância entre os objectos é dada pela constante `DIST_OBJ`. As dimensões da árvore grande são dadas pelas constantes `LARGURA_TRONCO`, `ALTURA_TRONCO`, `ALTURA_COPA`. O raio do círculo é dado pela constante `RAIO_BOLA`.







## Sistemas Gráficos e Interacção

Época de Recurso

2014-02-12

N.º \_\_\_\_\_ Nome \_\_\_\_\_

f. **[6.0]** Escreva o código que permite efectuar a animação da cena descrita na alínea anterior que a seguir se descreve:

- A bola começa a rolar para a esquerda;
- A bola pára quando colidir com a árvore pequena;
- Quando atingida pela bola, a árvore pequena começa a cair para a esquerda.

---

---

---

---

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

```
    glutTimerFunc(1000, Timer, 0);
```

---

---

---

---

---

---

---

---

---

---

```
    // redesenhar o ecrã
```

```
    glutPostRedisplay();
```

```
}
```