

N.º _____ Nome _____

Duração da prova: 75 minutos

Perguntas de escolha múltipla: cada resposta incorrecta desconta 1/3 do valor da pergunta

Parte Teórica

30% – 8 valores mín.

- a. Da comparação de uma interface WIMP (*Windows, Icons, Menus and Pointers*) com uma de linha de comandos resulta normalmente que
- i. Os paradigmas de interacção e as metáforas adoptadas são idênticos em ambos os casos
 - ii. O esforço de memorização e a carga cognitiva impostos aos utilizadores são menores na primeira do que na segunda
 - iii. A probabilidade de ocorrência de erros de interacção é maior na primeira do que na segunda
 - iv. Todas as anteriores
- b. Um sistema gráfico dotado de um *frame buffer* RGBA de 1024 x 768 x 32 bits
- i. Não está preparado para reproduzir imagens de cenas compostas por objectos transparentes
 - ii. Permite a reprodução de gráficos vectoriais
 - iii. Permite a reprodução de imagens com 512 níveis de azul
 - iv. Todas as anteriores
- c. Uma única multiplicação de matrizes permite efectuar
- i. Uma translação, seguida de uma rotação, seguida de um escalamento
 - ii. Uma projecção perspectiva, desde que a seguir se efectue uma operação de divisão perspectiva
 - iii. Uma mudança de sistema de coordenadas
 - iv. Todas as anteriores
- d. No modelo da câmara virtual adoptado pelo OpenGL, as transformações de *viewport* permitem
- i. Posicionar os objectos na cena e, se for caso disso, deformá-los
 - ii. Orientar a câmara
 - iii. Ampliar e reduzir a fotografia tirada pela câmara
 - iv. Nenhuma das anteriores

- e. A equação $x^2 + y^2 + z^2 - 1 = 0$ descreve implicitamente a superfície de
- Um cubo unitário centrado na origem
 - Um cilindro de raio e altura unitários centrado na origem
 - Uma esfera unitária centrada na origem
 - Nenhuma das anteriores
- f. Em OpenGL um objecto muito polido pode ser simulado usando um material
- Que reflecte significativamente a componente de luz difusa
 - Com uma elevada componente de emissão
 - Com um coeficiente de especularidade reduzido
 - Nenhuma das anteriores
- g. No modelo de iluminação do OpenGL o vector *halfway* é usado para calcular de uma forma simplificada
- A componente ambiente de iluminação
 - A componente difusa de iluminação
 - A componente especular de iluminação
 - Nenhuma das anteriores
- h. No mapeamento de texturas em OpenGL, o processo de filtragem designado por `GL_NEAREST_MIPMAP_LINEAR`
- Não é usado em contextos de magnificação, ou seja, quando a um pixel no ecrã corresponde apenas uma fracção de um *texel* na imagem de textura
 - Calcula uma média pesada da matriz de 2×2 *texels* que mais se aproxima do centro do pixel no *mipmap* que melhor se adequa ao contexto de minificação existente
 - Calcula uma média pesada da matriz de 2×2 *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
 - Nenhuma das anteriores

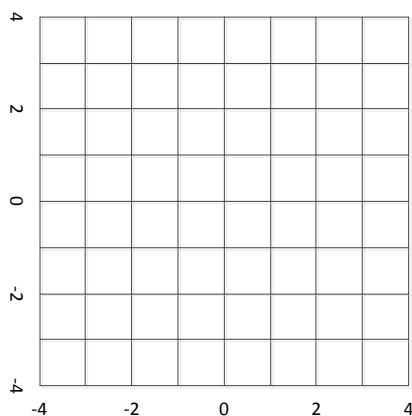
N.º _____ Nome _____

Parte Teórico-Prática

40% – 8 valores mín.

- a. Esboce o desenho que resulta do seguinte extracto de código.

```
glBegin(GL_TRIANGLE_STRIP);
    glVertex2f(1.0, 1.0);
    glVertex2f(2.0, 1.0);
    glVertex2f(2.0, 2.0);
    glVertex2f(3.0, 2.0);
glEnd();
```



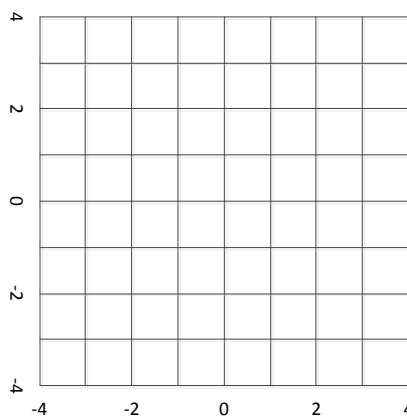
- b. Considere uma cena desenhada usando o plano OXZ como solo. Sabendo que a posição do personagem é representada por $\{obj.x, obj.y, obj.z\}$, configure a câmara de modo a obter uma vista de topo da cena centrada no personagem.

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


N.º _____ Nome _____

- c. Esboce o desenho que resulta do seguinte extracto de código.

```
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
glRotatef(90.0, 0.0, 0.0, 1.0);
glTranslatef(-2.0, 1.0, 0.0);
glRectf(0.0, 0.0, 1.0, 1.0);
```

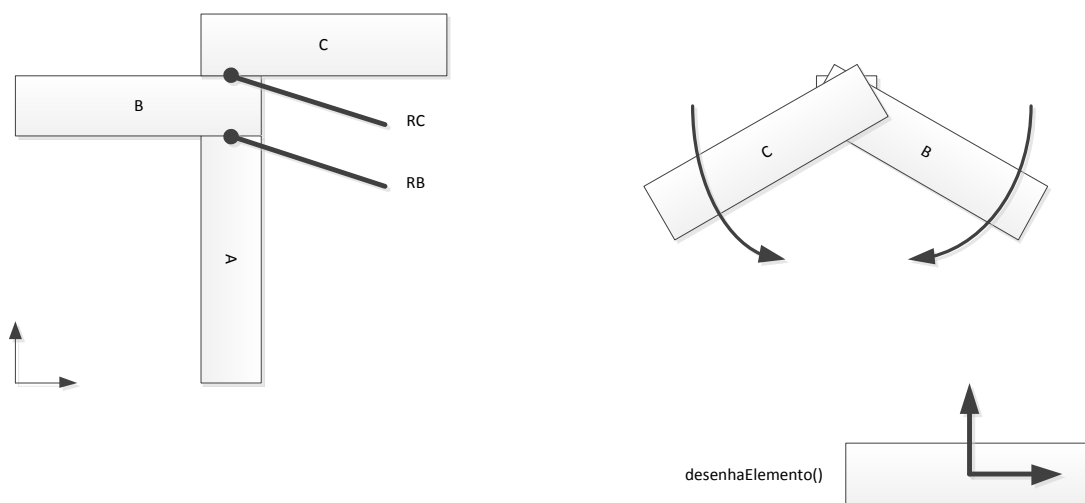


- d. Considere uma cena composta por um cubo de material $\{0, 0.25, 0\}$ e uma esfera de material $\{1, 0, 0.5\}$. A cena é iluminada unicamente por uma fonte de luz direccional cuja direcção é dada por $\{1, 1, 1\}$ e componente difusa por $\{0, 1, 1\}$. Que objectos serão visíveis?
- Apenas o cubo
 - Apenas a esfera
 - Tanto o cubo como a esfera
 - Nenhum deles
- e. Suponha que pretende aplicar uma textura a um polígono com uma imagem repetida duas vezes na horizontal. Complete o seguinte extracto de código.

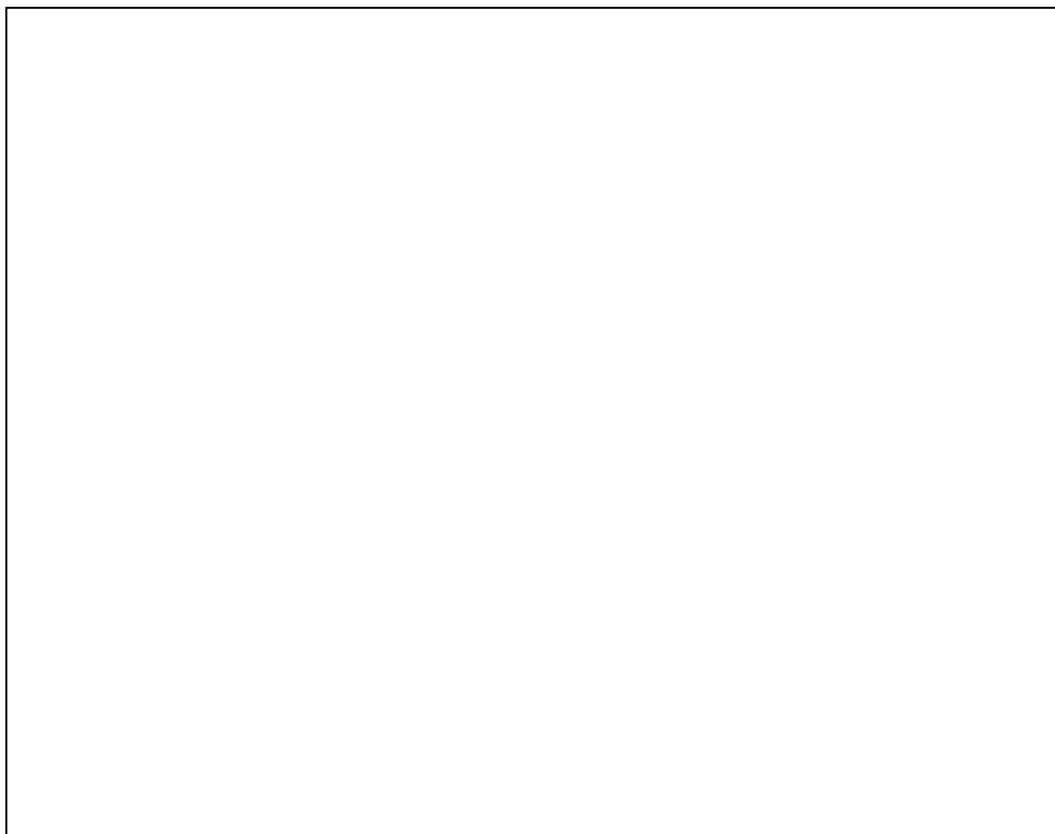
```
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, _____);
glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, _____);
...
glBegin(GL_QUADS);
    glTexCoord2f(_____, _____);
    glVertex3f(0.0, 0.0, 0.0);
    glTexCoord2f(_____, _____);
    glVertex3f(0.0, 1.0, 0.0);
    glTexCoord2f(_____, _____);
    glVertex3f(1.0, 1.0, 0.0);
    glTexCoord2f(_____, _____);
    glVertex3f(1.0, 0.0, 0.0);
glEnd();
```


N.º _____ Nome _____

- f. Pretende-se construir o objecto da figura seguinte recorrendo à função `desenhaElemento()` que desenha um paralelepípedo centrado na origem de dimensão $L \times A \times P$. Os três elementos possuem as mesmas dimensões. Pretende-se ainda que o objecto tenha os elementos B e C articulados nos pontos RB e RC como se mostra na figura.



- i. Desenhe a árvore de cena correspondente.



- [illegible]

N.º _____ Nome _____

- iii. Escreva o código que deve colocar no *timer* para animar o objecto anterior de modo a que o elemento B rode duas vezes mais depressa do que o elemento C.

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

```
    glutTimerFunc(estado.delay, Timer, 0);
```

```
    // redesenhar o ecrã
```

```
    glutPostRedisplay();
```

```
}
```