

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. Considere uma situação em que o utilizador de um sistema de manipulação directa pretende efectuar uma tarefa que, tendo em conta o contexto de utilização em vigor naquele momento, é ilegal. Qual o comportamento mais adequado da interface face a esta situação?
- ☒ i. O objecto de interacção que desencadeia a tarefa deverá estar desactivado, de modo a impedir o accionamento do mesmo por parte do utilizador
  - ii. O objecto de interacção que desencadeia a tarefa deverá permanecer activado e, caso o utilizador o accione, deverá surgir uma mensagem de erro a explicar a razão pela qual a tarefa não pode ser efectuada
  - iii. O objecto de interacção que desencadeia a tarefa deverá permanecer activado e, caso o utilizador o accione, nada acontecerá
  - iv. Nenhuma das anteriores
- b. Um sistema gráfico com um *frame buffer* RGB de 24 bits tem capacidade para reproduzir imagens com
- i. Uma resolução de  $2^6 \times 2^4$  píxeis
  - ☒ ii.  $2^8$  níveis de azul
  - iii.  $2^{32}$  cores
  - iv. Nenhuma das anteriores
- c. Uma transformação de inclinação (*shearing*) constitui um exemplo de uma transformação
- ☒ i. Linear
  - ii. Rígida
  - iii. Projectiva
  - iv. Todas as anteriores
- d. Numa projecção paralela
- i. As dimensões aparentes dos objectos diminuem com o aumento da distância à câmara
  - ☒ ii. As dimensões aparentes dos objectos não dependem da distância à câmara
  - iii. O volume de visualização tem a forma de um tronco de pirâmide
  - iv. Nenhuma das anteriores
- e. A parametrização da superfície que delimita um sólido
- i. Não é usada na representação de sólidos por fronteira (B-Rep)
  - ii. É válida mesmo quando a superfície se auto-intersecta
  - ☒ iii. Estabelece um sistema de coordenadas sobre a superfície herdado de um sistema de coordenadas no plano
  - iv. Todas as anteriores

- f. A contribuição dada pela componente de iluminação especular do modelo de Phong
- i. Não é compatível com as fontes de luz direccionais
  - ii. É característica dos materiais baços ou foscos
  - iii. Depende do ângulo de incidência da fonte de luz, mas não depende da posição do observador
  - iv. Nenhuma das anteriores
- g. De que forma ou formas permite o mecanismo de mapeamento de texturas do OpenGL aplicar uma textura à superfície de um objecto?
- i. Substituindo a cor da superfície pela dos téxeis
  - ii. Modulando a cor da superfície com a dos téxeis
  - iii. Misturando a cor da superfície com uma cor predefinida
  - iv. Todas as anteriores
- h. O método *Z-Buffer* de determinação de visibilidade
- i. Opera no espaço do objecto
  - ii. Não é susceptível de ser implementado em hardware
  - iii. Mantém para cada pixel um valor de profundidade
  - iv. Nenhuma das anteriores

N.º \_\_\_\_\_ Nome \_\_\_\_\_

Parte Teórico-Prática

40% – 10 valores mín.

**Notas:**

Em todas as perguntas, a menos que algo seja dito em contrário, assuma a orientação dos eixos e a posição da câmara por omissão do OpenGL.

*Fórmula paramétrica da circunferência:*

$$x = r \cdot \cos(t)$$

$$y = r \cdot \sin(t)$$

$$\cos(60^\circ) = 0,5$$

$$\sin(60^\circ) = 0,87$$

- a. Suponha que a sua cena tem uma única luz, definida unicamente com `glLightfv(GL_LIGHT0, GL_DIFFUSE, ld)`, em que `ld` é o vector `{1, 0.5, 0.8, 1}`, e um objecto com material definido unicamente com `glMaterial(GL_FRONT_AND_BACK, GL_DIFFUSE, md)`, em que `md` é o vector `{0.5, 1, 0.5, 1}`. Qual a cor (em termos de RGB) resultante que se verá no ecrã?

$$R = 0.5$$

$$G = 0.5$$

$$B = 0.4$$

- b. Preencha o seguinte vector para que defina uma luz posicional na posição (4,3,1).

```
float lpos[] = {4.0, 3.0, 1.0, 1.0};  
glLightfv(GL_LIGHT0, GL_POSITION, lpos);
```

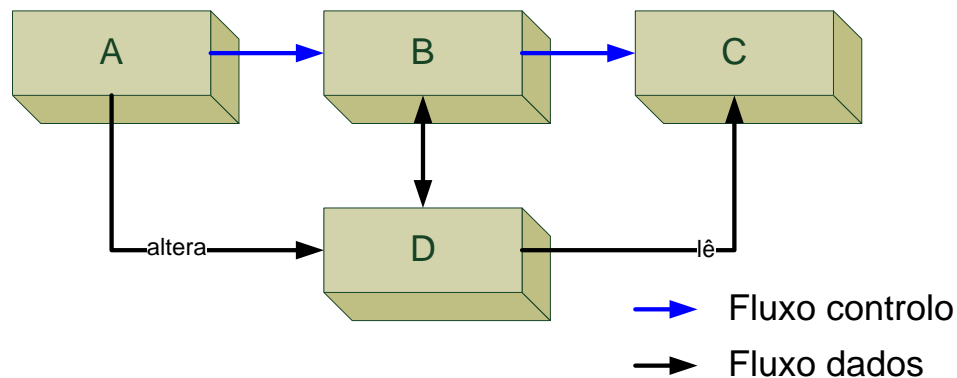
- c. Pretende ter uma vista de topo da cena centrada no seu objecto. A posição deste é dada pelas variáveis `obj.x`, `obj.y`, `obj.z`. Como deve configurar a câmara usando `gluLookAt`?

```
gluLookAt( obj.x, obj.y, obj.z + ALTURA CAMARA,  
           obj.x, obj.y, obj.z,  
           0.0, 1.0, 0.0);
```



N.º \_\_\_\_\_ Nome \_\_\_\_\_

- d. Identifique cada um dos quatros blocos fundamentais de um programa gráfico (ver figura) e indique sucintamente para que servem.



A = Input

Trata de receber os eventos de *input* do sistema (teclas, rato, etc.). Também se podem considerar os temporizadores como eventos de *input*. Eventualmente altera o modelo.

B = Lógica

Implementa as regras de “movimentação” dos objectos no “mundo”. Normalmente, em resposta aos eventos de *input*, irá alterar os valores das variáveis do modelo. Deve manter o ritmo da “história” independente do desempenho do hardware.

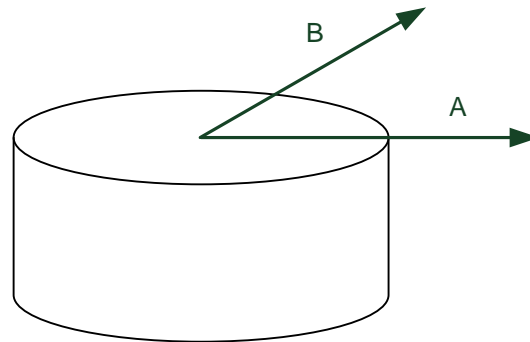
C = Renderer

Trata de desenhar a cena de acordo com os valores actuais das variáveis do modelo. Deve obter o máximo de ciclos de refrescamento que for possível.

D = Modelo

Estruturas de dados e variáveis que representam a cena, isto é, o “mundo” que se pretende desenhar.

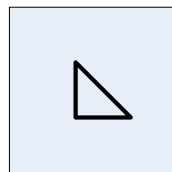
- e. Calcule as normais unitárias A e B do seguinte objecto, sabendo que a “circunferência” do topo do cilindro tem 2 unidades de raio e foi desenhada com 12 segmentos, sendo que a normal A corresponde ao vértice com 0° e a normal B ao vértice com 60°.



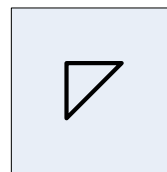
$$A = (1, 0, 0)$$

$$B = (0.5, 0, -0.87)$$

- f. Suponha que pretende texturizar um quadrado por forma a ficar como na figura (a), usando a imagem de textura (b). Quais as coordenadas de textura que deve usar?



(a)

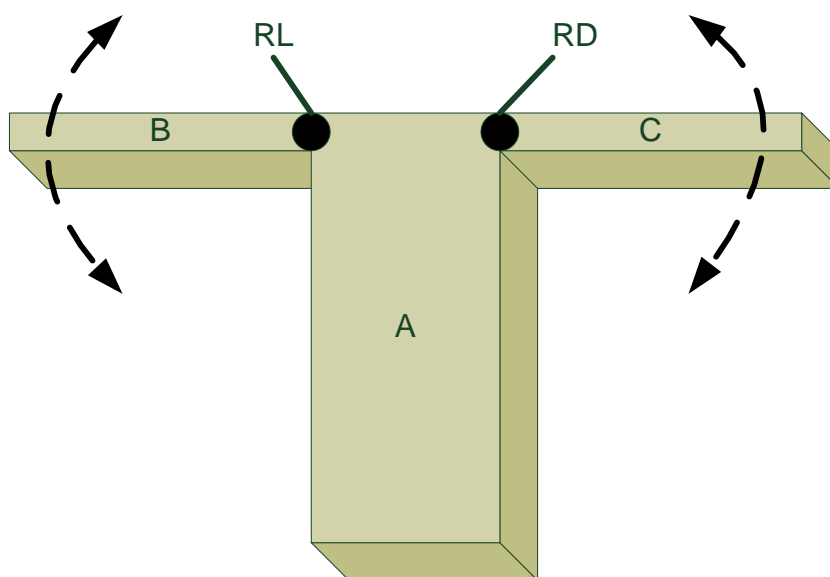


(b)

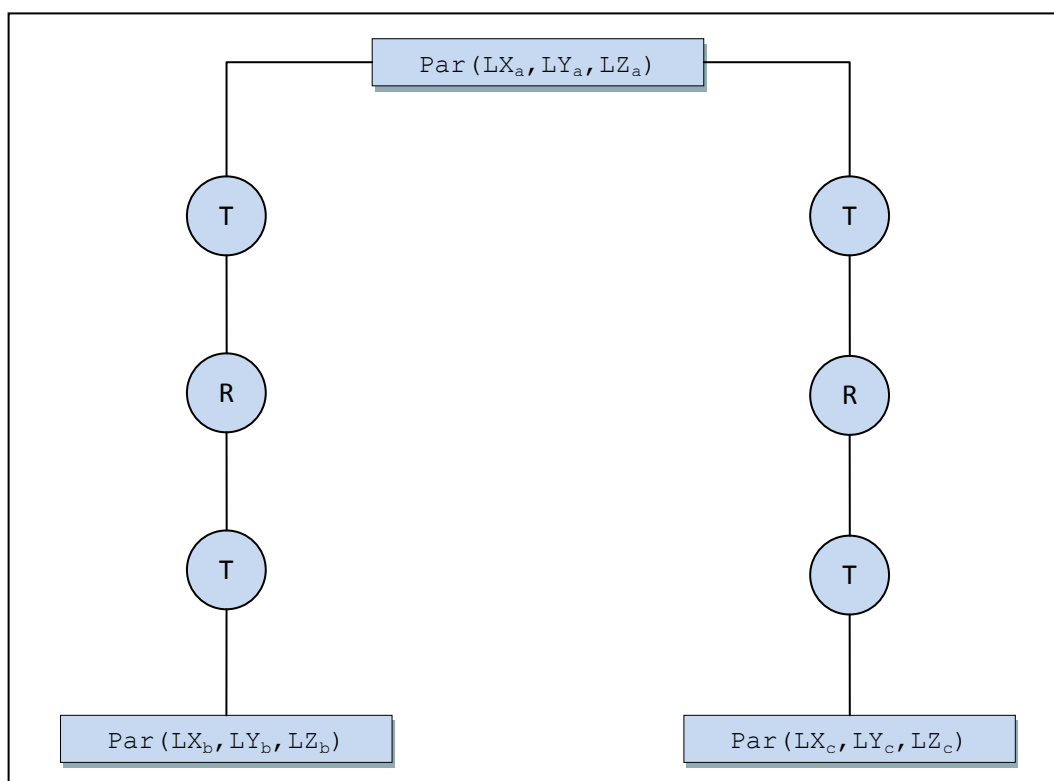
```
glBegin(GL_QUADS);
    glTexCoord2f(0.0, 1.0);
    glVertex3f(-2.0, -1.0, 0.0);
    glTexCoord2f(1.0, 1.0);
    glVertex3f(-2.0, 1.0, 0.0);
    glTexCoord2f(1.0, 0.0);
    glVertex3f(0.0, 1.0, 0.0);
    glTexCoord2f(0.0, 0.0);
    glVertex3f(0.0, -1.0, 0.0);
glEnd();
```

N.º \_\_\_\_\_ Nome \_\_\_\_\_

- g. Pretende-se construir o objecto da figura seguinte recorrendo à função de desenho `desenhaParalelo(LX, LY, LZ)`, a qual desenha um paralelepípedo centrado na origem do sistema “local” de eixos, com dimensões LX, LY e LZ, ao longo dos eixos x, y e z respectivamente. Pretende-se que o objecto tenha os elementos B e C articulados nos pontos RL e RD, como se mostra na figura.



- i. Desenhe a árvore de cena correspondente.



- ii. Escreva a sequência de instruções OpenGL necessárias para desenhar o objecto, supondo a existência de duas variáveis, `modelo.rl` e `modelo.rd`, que indicam o ângulo de rotação dos elementos B e C respectivamente.

```
desenhaParalelo(LXa, LYa, LZa);
```

---

```
glPushMatrix();
```

---

```
glTranslatef(-LXa / 2.0, LYa / 2.0 - LYb / 2.0, 0.0);
```

---

```
glRotatef(modelo.rl, 0.0, 0.0, 1.0);
```

---

```
glTranslatef(-LXb / 2.0, 0.0, 0.0);
```

---

```
desenhaParalelo(LXb, LYb, LZb);
```

---

```
glPopMatrix();
```

---

```
glPushMatrix();
```

---

```
glTranslatef(LXa / 2.0, LYa / 2.0 - LYc / 2.0, 0.0);
```

---

```
glRotatef(modelo.rd, 0.0, 0.0, 1.0);
```

---

```
glTranslatef(LXc / 2.0, 0.0, 0.0);
```

---

```
desenhaParalelo(LXc, LYc, LZc);
```

---

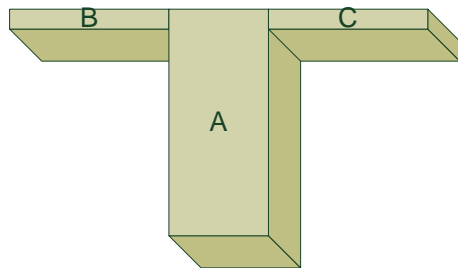
```
glPopMatrix();
```

---

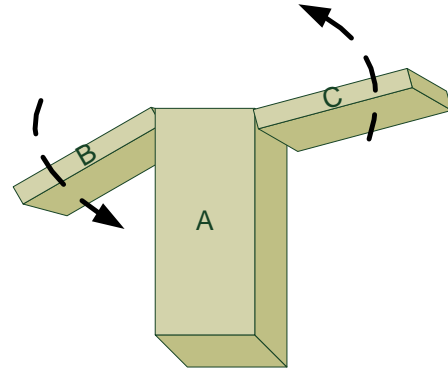


N.º \_\_\_\_\_ Nome \_\_\_\_\_

- iii. Escreva o código que deve colocar no *timer* para animar o objecto anterior da forma indicada na figura seguinte, partindo da posição inicial e terminando quando cada um dos elementos tiver rodado um total de 90°.



Posição inicial



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

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

```
    if(modelo.rl < 90)
```

```
        modelo.rl += INC_RL; // INC_RL positivo
```

```
    if(modelo.rd < 90)
```

```
        modelo.rd += INC_RD; // INC_RD positivo
```

```
    // redesenhar o ecrã
```

```
    glutPostRedisplay();
```

```
}
```