

Parte Teórica

50% – 60 minutos

N.º \_\_\_\_\_ Nome \_\_\_\_\_

1. As representações **vectoriais** de gráficos
  - a) Caracterizam-se por uma complexidade de processamento  $O(n.\text{º de píxeis})$
  - b) Consistem em colecções de objectos geométricos tais como pontos, segmentos de linhas rectas e polígonos
  - c) Implicam uma perda significativa de precisão na realização de operações como a rotação e a escala
  - d) Podem ser convertidas em representações matriciais com base em técnicas de reconhecimento de padrões
2. Se a um ponto com coordenadas  $(x_1, y_1, z_1)$  subtrairmos um ponto com coordenadas  $(x_0, y_0, z_0)$  obteremos
  - a) Um ponto com coordenadas  $(x_1 - x_0, y_1 - y_0, z_1 - z_0)$
  - b) Um escalar com o valor  $(x_1 - x_0) * (y_1 - y_0) * (z_1 - z_0)$
  - c) Um vector com componentes  $(x_1 - x_0, y_1 - y_0, z_1 - z_0)$
  - d) Uma matriz quadrada 3 x 3
3. Se a um determinado objecto aplicarmos uma translação genérica seguida de uma rotação genérica obteremos
  - a) Um objecto de dimensões idênticas às do objecto original
  - b) Um objecto deformado, pois a transformação que resulta da composição de translações e rotações não é rígida
  - c) Um objecto centrado na origem
  - d) O mesmo efeito que obteríamos se aplicássemos as mesmas transformações pela ordem inversa
4. Considere o objecto  $A'B'C'$  que resulta da projecção de um triângulo genérico definido pelos vértices  $A, B$  e  $C$ . Se o cálculo da orientação daquele objecto resultar num valor positivo, podemos concluir que
  - a) O triângulo está orientado de perfil para o observador
  - b) O triângulo está orientado com a face de trás voltada para o observador
  - c) O triângulo está orientado com a face da frente voltada para o observador
  - d) Algo está errado, pois a orientação de três pontos complanares não pode assumir valores positivos

5. Na representação de sólidos por **fronteira** (B-Rep – *Boundary Representation*)

- a) Há ambiguidade, pois a uma mesma representação podem corresponder vários modelos
- b) O modelo é representado através da superfície que o delimita
- c) O modelo é representado por uma árvore em que os nós internos designam operações de conjuntos ou transformações lineares afim e as folhas denotam objectos primitivos
- d) O modelo é representado com base na divisão do espaço em cubos cujos lados são potências de base 2

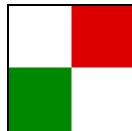
6. No modelo de Phong, a componente **emissiva** de iluminação

- a) Depende do co-seno do ângulo de incidência da luz
- b) Depende das posições do observador, do objecto e da fonte de luz
- c) Simula a fluorescência dos materiais constituintes dos objectos
- d) É atenuada pela distância entre a fonte de luz e o objecto iluminado

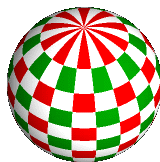
7. Em OpenGL, um ângulo de *cutoff* de 180° configura

- a) Uma fonte de luz posicional rodada de 180°
- b) Uma fonte de luz direccionada paralela à linha de visão da câmara
- c) Uma fonte de luz omnidireccional
- d) Um projector de luz

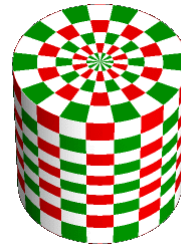
8. Se usarmos uma parametrização esférica na aplicação a um cilindro da textura representada na figura, obteremos o seguinte resultado



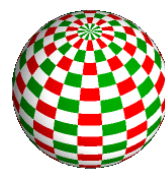
a)



b)



c)



d)

9. As mensagens de erro

- a) Devem ser fraseadas de forma positiva e construtiva
- b) Devem atribuir a culpa do erro ao utilizador
- c) Devem conter antropomorfismos
- d) Todas as anteriores

Parte Teorico-Prática

50% – 60 minutos

N.º \_\_\_\_\_ Nome \_\_\_\_\_

10. Para simular o efeito de iluminação do Sol ao longo do dia numa cena de exterior (ex., mesa de piquenique), devemos usar uma luz móvel posicional ou direccional?

---

---

11. Como o OpenGL tem o conceito de cor actual, após importar um modelo externo muitas vezes a restante cena é desenhada com cores diferentes das desejadas. Que instrução do OpenGL pode ser utilizada para evitar este problema?

---

---

12. Supondo que possui um método **desenhaParalelepipedo**( $L$ ,  $A$ ,  $P$ ) que desenha um paralelepípedo com largura  $L$ , altura  $A$  e profundidade  $P$  (figura 1), com base na origem do sistema de eixos, escreva a sequência de instruções necessárias para desenharmos o objecto da figura 2.



Figura 1

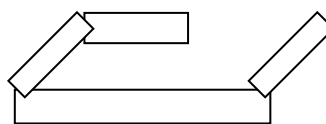


Figura 2

---

---

---

---

---

---

---

---

---

---



Parte Teorico-Prática

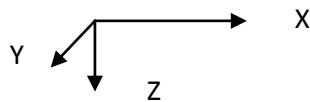
50% – 60 minutos

N.º \_\_\_\_\_ Nome \_\_\_\_\_

13. Quando movemos a câmara e uma parte do ecrã fica preto com um corte na cena desenhada, significa que

- a) Os objectos estão mal desenhados
- b) A projecção está mal definida
- c) A câmara está mal localizada
- d) O sistema de coordenadas está trocado

14. Caso queira que o seu sistema de eixos fique orientado como indicado na figura, deve usar:



- a) `gluLookAt(0, 0, -1, 0, 0, 0, 0, -1, 0)`
- b) `gluLookAt(0, 0, 1, 0, 0, 0, 0, -1, 0)`
- c) `gluLookAt(0, 1, 0, 0, 0, 0, 0, 0, -1)`
- d) `gluLookAt(0, 1, 0, 0, 0, 0, 0, 0, 1)`

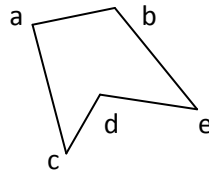
15. A utilização de *display lists* **não** resolve um dos seguintes problemas

- a) Performance
- b) Evitar recálculo de vértices
- c) Objectos com transformações dependentes de variáveis
- d) Utilização de múltiplas texturas

16. Um objecto que se mova no espaço bidimensional XY pode ser caracterizado pela posição (x, y) e direcção (ângulo em radianos). Se pretendermos que a câmara se encontre sempre atrás do objecto devemos definir a posição do objecto como o centro da câmara e utilizar para eye, qual dos seguintes valores?

- a)  $\text{eye.x} = \text{obj.x} - r * \cos(\text{obj.dir} + M\_PI);$   
 $\text{eye.y} = \text{obj.y} - r * \sin(\text{obj.dir} + M\_PI);$
- b)  $\text{eye.x} = \text{obj.x} + r * \cos(\text{obj.dir} + M\_PI);$   
 $\text{eye.y} = \text{obj.y} - r * \sin(\text{obj.dir} + M\_PI);$
- c)  $\text{eye.x} = \text{obj.x} + r * \cos(\text{obj.dir} + M\_PI);$   
 $\text{eye.y} = \text{obj.y} + r * \sin(\text{obj.dir} + M\_PI);$
- d)  $\text{eye.x} = \text{obj.x} - r * \cos(\text{obj.dir} + M\_PI);$   
 $\text{eye.y} = \text{obj.y} + r * \sin(\text{obj.dir} + M\_PI);$

17. Indique a sequência de código que permitiria desenhar o seguinte polígono



- a) `glBegin(GL_LINES);`  
`glVertex3fv(a);`  
`glVertex3fv(b);`  
`glVertex3fv(c);`  
`glVertex3fv(d);`  
`glVertex3fv(e);`  
`glEnd();`
- b) `glBegin(GL_POLYGON);`  
`glVertex3fv(a);`  
`glVertex3fv(c);`  
`glVertex3fv(e);`  
`glVertex3fv(d);`  
`glVertex3fv(c);`  
`glEnd();`
- c) `glBegin(GL_TRIANGLE_FAN);`  
`glVertex3fv(a);`  
`glVertex3fv(b);`  
`glVertex3fv(e);`  
`glVertex3fv(d);`  
`glVertex3fv(c);`  
`glEnd();`
- d) `glBegin(GL_LINE_STRIP);`  
`glVertex3fv(a);`  
`glVertex3fv(b);`  
`glVertex3fv(e);`  
`glVertex3fv(d);`  
`glVertex3fv(c);`  
`glEnd();`

18. Qual a normal (não unitária) do polígono definido pelos vértices (1, 0, 0), (0, 1, 0), (0, 1, 1) e (1, 0, 1)?

- a) (1, 0, 0)
- b) (1, 1, 1)
- c) (1, 1, 0)
- d) (0, 1, 1)