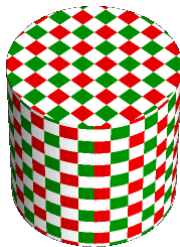
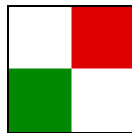


1. As representações matriciais de gráficos
  - a) Consistem em colecções de objectos geométricos tais como pontos, segmentos de linhas rectas e polígonos
  - b) Permitem que operações como a rotação e a escala sejam efectuadas quase sem perda de precisão
  - c) Caracterizam-se por uma complexidade de processamento  $O(n.º \text{ de píxeis})$
  - d) Podem ser convertidas em representações vectoriais com base em técnicas de rasterização (*scan conversion*)
2. Se a um ponto com coordenadas  $(x, y, z)$  adicionarmos um vector com componentes  $(v_x, v_y, v_z)$  obteremos
  - a) Um ponto com coordenadas  $(x + v_x, y + v_y, z + v_z)$
  - b) Um vector com componentes  $(x * v_x, y * v_y, z * v_z)$
  - c) Um ponto com coordenadas  $(x * v_x, y * v_y, z * v_z)$
  - d) Um vector com componentes  $(x + v_x, y + v_y, z + v_z)$
3. Em que circunstâncias é que a composição de uma translação com uma rotação não depende da ordem de aplicação destas transformações?
  - a) Nunca
  - b) Quando o vector que caracteriza a translação for paralelo ao eixo da rotação
  - c) Quando o vector que caracteriza a translação for perpendicular ao eixo da rotação
  - d) Sempre que usarmos coordenadas homogéneas
4. Na metáfora da câmara virtual adoptada pelo OpenGL, as transformações de visualização correspondem a
  - a) Revelar as fotografias tiradas pela câmara
  - b) Ajustar a lente/objectiva da câmara
  - c) Posicionar/orientar a câmara
  - d) Posicionar/orientar/deformar os objectos da cena
5. Na representação de sólidos com base em operações de conjuntos (CSG – *Constructive Solid Geometry*)
  - a) O modelo é representado através da superfície que o delimita
  - b) 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
  - c) O modelo é representado com base na divisão do espaço em cubos cujos lados são potências de base 2
  - d) Há ambiguidade, pois a uma mesma representação podem corresponder vários modelos

6. No modelo de Phong, a componente de iluminação ambiente

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

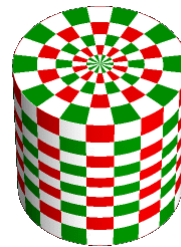
7. Se usarmos uma parametrização cilíndrica na aplicação a um cilindro da textura representada na figura, obteremos o seguinte resultado



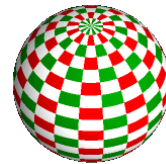
a)



b)



c)



d)

8. O *design* de uma interface WIMP

- a) Deve contemplar o uso de teclas aceleradoras, não obstante o mesmo implicar um esforço considerável de memorização para o utilizador
- b) Deve ser centrado no sistema e não no utilizador, pois o comportamento deste último é imprevisível
- c) Não deve basear-se em analogias com o mundo real, pois pode suscitar confusão na mente do utilizador
- d) Nenhuma das anteriores

9. O desencadeamento de acções potencialmente destrutivas

- a) Deve ser acompanhado de uma mensagem de alerta para o administrador do sistema
- b) Não deve ser permitido sejam quais forem as circunstâncias, pois a sua incorrecta utilização pode acarretar custos muito elevados para o utilizador
- c) Deve ser sujeito a confirmação, não obstante o esforço extra que tal implica para o utilizador
- d) Nenhuma das anteriores

## Parte Teorico-Prática

N.º \_\_\_\_\_ Nome \_\_\_\_\_

10. Quando aplicamos texturas e iluminação, que cuidado devemos ter na configuração da textura para que a iluminação se note na cena final?

---



---

11. Ao importar modelos é normalmente necessário aplicar uma transformação de escala para que se adequem ao resto da cena do nosso programa. Qual é o cuidado necessário para que a iluminação e texturas do objecto sejam visualizadas correctamente?

---



---

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

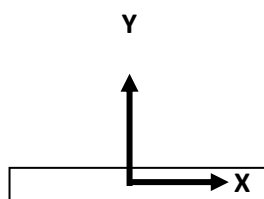


Figura 1

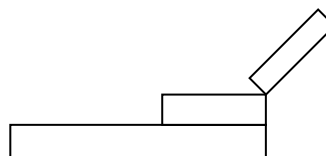


Figura 2

---



---



---



---



---



---

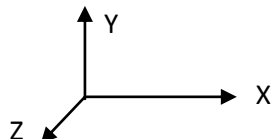


---



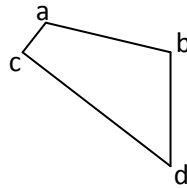
---

13. Se as cores de um objecto se alteram quando movemos a câmara, e tal não era suposto, isso normalmente indica problemas de
- Iluminação mal definida
  - Normais mal definidas
  - Texturas erradas
  - Callback* do temporizador muito lento
14. Qual dos seguintes vectores define uma luz direcciona paralela com a linha de visão da câmara na sua posição por omissão?
- (1, 1, 1, 0)
  - (1, 0, 0, 0)
  - (0, 1, 0, 0)
  - (0, 0, 1, 0)
15. Um objecto que se mova no espaço bidimensional XY pode ser caracterizado pela posição (x, y) e direcção (ângulo em radianos). Se pretendermos que o objecto se desloque para uma nova posição, devemos usar
- $nx = obj.x + r * \cos(obj.dir);$   
 $ny = obj.y + r * \sin(obj.dir);$
  - $nx = obj.x - r * \cos(obj.dir);$   
 $ny = obj.y - r * \sin(obj.dir);$
  - $nx = obj.x + r * \cos(obj.dir);$   
 $ny = obj.y - r * \sin(obj.dir);$
  - $nx = obj.x - r * \cos(obj.dir);$   
 $ny = obj.y + r * \sin(obj.dir);$
16. Qual dos seguintes comandos **não** pode ser utilizado dentro de glBegin/glEnd?
- glVertex3f
  - glColor3f
  - glRectf
  - glTexCoord2f
17. Caso queira que o seu sistema de eixos fique orientado como indicado na figura, deve usar



- gluLookAt(0, 0, -1, 0, 0, 0, 0, 1, 0)
- gluLookAt(0, 0, 1, 0, 0, 0, 0, 1, 0)
- gluLookAt(0, 1, 0, 0, 0, 0, 0, 0, 1)
- gluLookAt(1, 0, 0, 0, 0, 0, 0, 0, 1)

18. Qual o extracto de código que desenharia o polígono da figura seguinte?



- a) glBegin(GL\_LINES);  
    glVertex3fv(a);  
    glVertex3fv(b);  
    glVertex3fv(c);  
    glVertex3fv(d);  
glEnd();
- b) glBegin(GL\_POLYGON);  
    glVertex3fv(a);  
    glVertex3fv(c);  
    glVertex3fv(d);  
    glVertex3fv(b);  
glEnd();
- c) glBegin(GL\_RECTANGLE);  
    glVertex3fv(a);  
    glVertex3fv(b);  
    glVertex3fv(c);  
    glVertex3fv(d);  
glEnd();
- d) glBegin(GL\_TRIANGLES);  
    glVertex3fv(a);  
    glVertex3fv(b);  
    glVertex3fv(c);  
    glVertex3fv(b);  
    glVertex3fv(a);  
    glVertex3fv(d);  
glEnd();