

## Sistemas Gráficos e Interação

Época Especial

2016-09-07

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 tarefa de supressão dos elementos de cena que se encontram fora da janela de visualização
- i. Designa-se por anti-discretização (*anti-aliasing*) e é normalmente efectuada pelo GPU, se existir
  - ii. Designa-se por rasterização (*scan conversion*) e é sempre efectuada pelo CPU
  - iii. Designa-se por recorte (*clipping*) e é normalmente efectuada pelo GPU, se existir
  - iv. Nenhuma das anteriores
- b. **[2.5]** As rotações e as escalas
- i. Não podem ser representadas na forma matricial
  - ii. São exemplos de transformações rígidas
  - iii. São exemplos de transformações projectivas
  - iv. Nenhuma das anteriores
- c. **[2.5]** Do produto escalar de dois vectores unitários resulta
- i. Um vector unitário
  - ii. Um vector perpendicular aos vectores originais
  - iii. O co-seno do ângulo descrito pelos vectores
  - iv. Nenhuma das anteriores
- d. **[2.5]** Em OpenGL, para se obter uma projecção em perspectiva associada a um volume de visualização simétrico, deverá recorrer-se às instruções
- i. `glOrtho()` ou `glFrustum()`
  - ii. `glOrtho()` ou `gluPerspective()`
  - iii. `glFrustum()` ou `gluPerspective()`
  - iv. Nenhuma das anteriores

- e. **[2.5]** A codificação *Winged-Edge* de sólidos
- i. É usada na representação de sólidos por fronteira (*B-Rep*)
  - ii. Armazena informação numa estrutura associada às arestas
  - iii. Permite determinar em tempo constante os 9 tipos de adjacência de vértices, arestas e faces
  - iv. Todas as anteriores
- f. **[2.5]** O cálculo das contribuições dadas pelas componentes difusa e especular do modelo de iluminação do OpenGL
- i. Não requer o conhecimento das normais
  - ii. Depende da posição da fonte de luz
  - iii. Depende da posição do observador
  - iv. Todas as anteriores
- g. **[2.5]** Para iluminar uma cena com uma fonte de luz direccional, deverá
- i. Especificar para a componente de luz ambiente emitida pela fonte um valor não nulo
  - ii. Especificar para a posição da fonte um conjunto de coordenadas homogéneas tal que  $w = 1$
  - iii. Especificar para o ângulo de *cutoff* um valor compreendido entre  $0^\circ$  e  $90^\circ$
  - iv. Nenhuma das anteriores
- h. **[2.5]** De que forma ou formas permite o mecanismo de mapeamento de texturas do OpenGL aplicar uma textura à superfície de um objecto?
- i. Misturando a cor da superfície com uma cor predefinida
  - ii. Modulando a cor da superfície com a dos téxeis
  - iii. Substituindo a cor da superfície pela dos téxeis
  - iv. Todas as anteriores

## Sistemas Gráficos e Interação

Época Especial

2016-09-07

### Parte Teórico-Prática

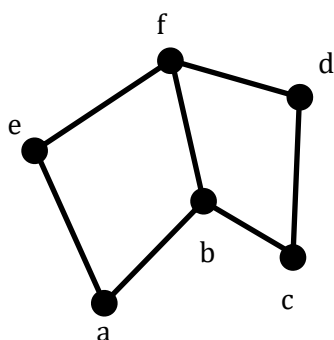
40%

**Resolução:** Em folhas próprias e separadas (A4). Não se esqueça de indicar, em cada folha, o número de estudante e o nome completo

**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

- a. **[2.0]** Pretende-se modelar o seguinte objecto com o recurso à primitiva `GL_QUADS`. Suponha que estamos a ver a face da frente dos *quads* e que as definições por omissão do OpenGL não foram alteradas. Complete o seguinte extracto de código, completando as instruções `glVertex3fv()` com as letras que considerar adequadas (na folha de resposta indique somente os vértices pela ordem pretendida).



```
glBegin(GL_QUADS);
  glVertex3fv(  a  );
  glVertex3fv(____);
  glVertex3fv(____);
  glVertex3fv(____);
  ...
  glVertex3fv(____);
glEnd();
```

- b. **[2.0]** Indique quais as funcionalidades normalmente associadas à função registada como *callback* de *reshape*.
- c. **[3.0]** Numa aplicação desenvolvida em OpenGL/GLUT, pretende-se actualizar ciclicamente a posição de um objecto. Indique todos os passos necessários para o fazer, considerando que somente estão registados os *callbacks* de *display* e de *reshape*.

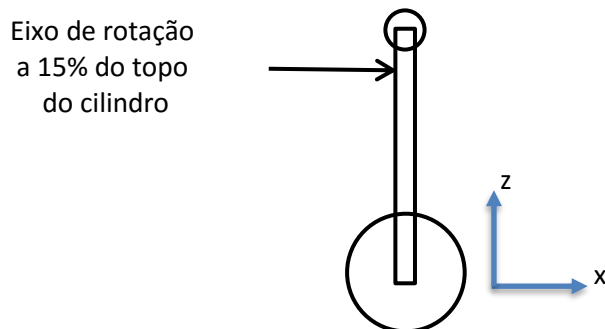
**Nota:** mude para uma nova folha de respostas

- d. **[2.0]** Indique quais os três tipos de posicionamento de luzes que podem ser definidos no OpenGL e que foram apresentados nas aulas teórico-práticas. Descreva sucintamente como podem ser obtidos.
- e. **[2.0]** Indique quais as principais diferenças na aplicação de texturas nos modos `GL_MODULATE` e `GL_REPLACE`.

**Nota:** mude para uma nova folha de respostas

- f. Considere o pêndulo ilustrado na figura. Este é desenhado com o recurso a um cilindro e a duas esferas centradas no topo e na base do cilindro. Utilize a função `cilindro()`, que desenha o cilindro da figura centrado na origem e com a orientação apresentada, e a função `esfera()`, que desenha uma esfera de raio unitário centrada na origem.

**Nota:** considere a existência das constantes `ALTURA_CILINDRO`, `RAIO_ESFERA_TOPO` e `RAIO_ESFERA_BASE` que definem as dimensões necessárias à modelação do pêndulo.



- i. **[3.0]** Construa a árvore de cena para a modelação do pêndulo. Tenha em consideração a rotação e a orientação dos eixos indicados na figura.
- ii. **[3.0]** Escreva o código para desenharmos o objecto. Considere a existência da variável `modelo.rotacao`, que representa a rotação do pêndulo.

**Nota:** mude para uma nova folha de respostas

- iii. **[3.0]** Complete o `timer` para realizar a animação do pêndulo do seguinte modo:

- O pêndulo roda continuamente até aos  $60^\circ$ ;
- Ao atingir a posição correspondente ao limite anterior, inverte o sentido da rotação e roda continuamente até aos  $-60^\circ$ ;
- Repete a sequência de animação.

Pode usar as variáveis no modelo que achar necessárias.

```
void Timer(int value)
{
    glutTimerFunc(100, Timer, 0);
    ...
    glutPostRedisplay();
}
```