

Sistemas Gráficos e Interação

Época Normal

2016-01-27

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]** Num sistema gráfico dotado de um *frame buffer* RGB de 1024 x 1024 píxeis, 8 bits/píxel
- i. Cada píxel é descrito por 3 bits para a componente vermelha, 3 bits para a verde e 2 bits para a azul
 - ii. Cada píxel é descrito por 8 bits para a componente vermelha, 8 bits para a verde e 8 bits para a azul
 - iii. Cada píxel é descrito por 1024 bits para a componente vermelha, 1024 bits para a verde e 1024 bits para a azul
 - iv. Nenhuma das anteriores
- b. **[2.5]** A realização de uma translação de um ponto P para um ponto P' distinto do anterior sem usar coordenadas homogéneas
- i. Não é de todo possível
 - ii. É possível mediante uma simples multiplicação de matrizes
 - iii. É possível, mas requer uma adição de matrizes
 - iv. Nenhuma das anteriores
- c. **[2.5]** No *pipeline* OpenGL de transformações
- i. A transformação de *viewport* converte as coordenadas normalizadas de dispositivo em coordenadas de janela
 - ii. A matriz de projecção converte as coordenadas de recorte em coordenadas normalizadas de dispositivo
 - iii. A matriz de modelação e visualização converte as coordenadas do objecto em coordenadas de recorte
 - iv. Todas as anteriores
- d. **[2.5]** Numa projecção perspectiva
- i. O volume de visualização tem a forma de um paralelepípedo e as dimensões aparentes dos objectos não dependem da distância à câmara
 - ii. O volume de visualização tem a forma de um paralelepípedo e as dimensões aparentes dos objectos diminuem com o aumento da distância à câmara
 - iii. O volume de visualização tem a forma de um tronco de pirâmide e as dimensões aparentes dos objectos não dependem da distância à câmara
 - iv. O volume de visualização tem a forma de um tronco de pirâmide e as dimensões aparentes dos objectos diminuem com o aumento da distância à câmara

- e. **[2.5]** Na representação por fronteira de um objecto (**B-rep**)
- É fácil exibir um ponto sobre a superfície do objecto
 - É fácil determinar, dado um ponto, se o mesmo se encontra no interior, na fronteira ou no exterior do objecto
 - As operações booleanas são avaliadas com facilidade
 - Nenhuma das anteriores
- f. **[2.5]** No modelo de iluminação do OpenGL, o vector *halfway* é usado para calcular de uma forma simplificada
- A componente especular de iluminação
 - A componente difusa de iluminação
 - A componente ambiente de iluminação
 - Todas as anteriores
- g. **[2.5]** Para iluminar uma cena com uma fonte de luz posicional do tipo projector, deverá
- Desactivar o mecanismo de mapeamento de texturas
 - Especificar para a posição um conjunto de coordenadas tal que $w = 0$
 - Especificar para o ângulo de *cutoff* um valor compreendido entre 0° e 90°
 - Todas as anteriores
- h. **[2.5]** A técnica de filtragem de texturas designada por *mipmapping*
- É aplicável aos contextos de magnificação
 - Permite que texturas de diferentes níveis de resolução sejam aplicadas de forma adaptativa
 - Não pode ser usada em conjugação com as projecções perspectivas
 - Nenhuma das anteriores

Sistemas Gráficos e Interação

Época Normal

2016-01-27

Parte Teórico-Prática

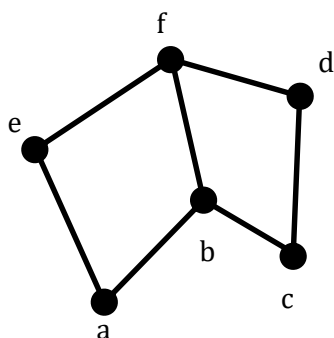
40%

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]** Suponha que pretende modelar o seguinte objecto com o recurso à primitiva `GL_QUAD_STRIP`. Transcreva para a folha da prova o seguinte extracto de código, completando as instruções `glVertex3fv()` com as letras que considerar adequadas. A primeira já se encontra preenchida.



```
glBegin(GL_QUAD_STRIP);
  glVertex3fv(  a  );
  glVertex3fv(    );
  glVertex3fv(    );
  glVertex3fv(    );
  glVertex3fv(    );
  glVertex3fv(    );
glEnd();
```

- b. **[2.0]** Qual é a utilidade da função `glEdgeFlag()` do OpenGL?
- c. **[2.0]** Qual é a importância do sentido com que se desenhavam os vértices de um polígono em OpenGL?

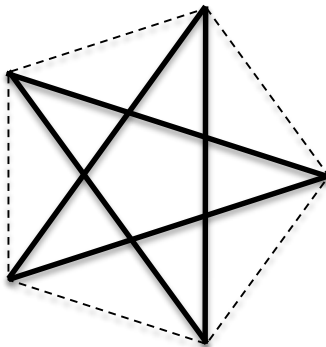
Nota: mude para uma nova folha de respostas

- d. **[2.0]** Considere um programa gráfico com uma janela principal e duas subjanelas. Indique o que terá de fazer para que o conteúdo das três janelas seja actualizado ao mesmo tempo.
- e. **[2.0]** Pretende-se simular a vista de uma câmara a seguir um automóvel de competição. Este está localizado em `(obj.x, obj.y, obj.z)` e desloca-se na direcção `obj.dir`. A câmara está situada na posição `(cam.x, cam.y, cam.z)`. Escreva o extracto de código que implementa a referida vista.

Nota: mude para uma nova folha de respostas

- f. **[4.0]** Crie a função `pentagrama()` que desenha o objecto representado com linhas contínuas na figura que se segue. O objecto deve ser desenhado com linhas.

Notas: O pentagrama está inscrito num círculo com o mesmo raio.
Não é necessário desenhá-lo a tracejado.



```
void pentagrama(GLfloat xCentro, GLfloat yCentro, GLfloat raio)
```

Nota: mude para uma nova folha de respostas

- g. Pretende-se realizar uma animação contínua do pentagrama na seguinte sequência (use as constantes e variáveis do modelo que entender necessárias):
- Aumenta o raio de 5 unidades em pequenos incrementos;
 - Roda continuamente 90° no sentido dos ponteiros do relógio;
 - Diminui o raio de 5 unidades em pequenos decrementos;
 - Roda continuamente 90° no sentido dos ponteiros do relógio;
 - Continua com o primeiro passo...
- i. **[3.0]** Implemente a função `desenhaCena()` que desenha o pentagrama e permita realizar esta animação.
- ii. **[3.0]** Implemente a função `timer()` para automatizar a animação.