

Sistemas Gráficos e Interação

Época de Recurso

2016-02-12

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] As vulgares impressoras de jacto de tinta constituem exemplos de dispositivos
- i. Vectoriais
 - ii. Tensoriais
 - iii. Matriciais
 - iv. Nenhuma das anteriores
- b. [2.5] Na representação de um ponto 3D em coordenadas homogéneas são usadas
- i. Apenas três componentes: x , y e z
 - ii. Quatro componentes: x , y , z e w , em que $w = 0$
 - iii. Quatro componentes: x , y , z e w , em que $w \neq 0$
 - iv. Nenhuma das anteriores
- c. [2.5] A qual das seguintes sequências de transformações corresponde uma matriz de transformação composta igual à matriz identidade?
- i. `glTranslated(1.0, 2.0, 3.0); glTranslated(-1.0, -2.0, -3.0);`
 - ii. `glRotated(30.0, 0.0, 1.0, 0.0); glRotated(-30.0, 0.0, 1.0, 0.0);`
 - iii. `glScaled(8.0, 4.0, 2.0); glScaled(0.125, 0.25, 0.5);`
 - iv. Todas as anteriores
- d. [2.5] Na codificação de sólidos com base em ponteiros para uma lista de vértices
- i. Cada face do sólido armazena explicitamente a lista ordenada das coordenadas dos seus vértices
 - ii. Há uma lista de vértices e as faces referenciam os seus vértices através de apontadores para essa lista
 - iii. A redundância de informação é maior do que na codificação explícita
 - iv. Todas as anteriores

- e. **[2.5]** Nas representações de objectos por células
- i. As grelhas dividem o espaço em cubos de igual dimensão
 - ii. As *octrees* dividem o espaço em cubos cujos lados são potências de 2
 - iii. As *BSP-trees* dividem o espaço em poliedros convexos
 - iv. Todas as anteriores
- f. **[2.5]** Em OpenGL, a alteração da propriedade `GL_SHININESS` de um material afecta a forma como é reflectida
- i. A componente ambiente de iluminação
 - ii. A componente difusa de iluminação
 - iii. A componente especular de iluminação
 - iv. Todas as anteriores
- g. **[2.5]** Um projector constitui um exemplo de uma fonte de luz
- i. Posicional
 - ii. Direccional
 - iii. Omnidireccional
 - iv. Nenhuma das anteriores
- h. **[2.5]** A correcção perspectiva permite
- i. Corrigir o efeito de discretização (*aliasing*) que decorre da utilização de *frame buffers* de baixa resolução
 - ii. Corrigir o efeito de diminuição da dimensão aparente de um objecto quando a distância do mesmo à câmara aumenta
 - iii. Corrigir o efeito de deformação que decorre da utilização de técnicas simples de interpolação linear no mapeamento de texturas em polígonos
 - iv. Nenhuma das anteriores

Sistemas Gráficos e Interação

Época de Recurso

2016-02-12

Parte Teórico-Prática

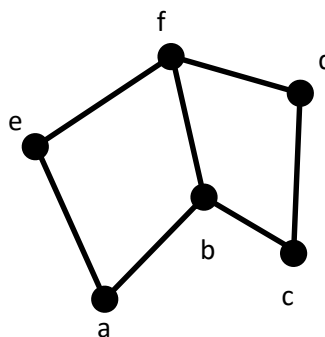
40%

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 objecto ilustrado na figura. Comente os dois extractos de código apresentados.



```
glBegin(GL_QUADS);
glVertex3fv(a);
glVertex3fv(b);
glVertex3fv(f);
glVertex3fv(e);
glVertex3fv(b);
glVertex3fv(c);
glVertex3fv(d);
glVertex3fv(f);
glEnd();
```

```
glBegin(GL_LINE_STRIP);
glVertex3fv(f);
glVertex3fv(d);
glVertex3fv(c);
glVertex3fv(b);
glVertex3fv(a);
glVertex3fv(e);
glVertex3fv(f);
glVertex3fv(b);
glEnd();
```

O extracto de código da esquerda desenha dois polígonos quadriláteros; o da direita desenha uma linha contínua.

- b. **[2.0]** Pretende-se aplicar uma textura a uma “parede” numa cena em que apenas existe uma fonte de luz. Esta está situada numa das extremidades da referida parede. Indique como poderia modelar o(s) polígono(s) e configurar a função de textura de modo a obter o efeito correcto de iluminação.

Garantir que a “parede” é modelada com vários polígonos, de modo a que cada polígono tenha o seu próprio cálculo de iluminação; e configurar a função de textura de forma a combinar os efeitos da iluminação e da aplicação de texturas (GL_MODULATE).

- c. **[2.0]** A função `glutSolidCube()` recebe apenas um argumento de dimensionamento do objecto a desenhar; e não recebe quaisquer argumentos de posicionamento desse objecto. No entanto, é possível utilizá-la para desenhar paralelepípedos de qualquer dimensão em qualquer posição. Explique como e porquê.

Em OpenGL existe a noção de sistema de coordenadas locais, já que todos os vértices a desenhar são sempre transformados pela matriz de transformação corrente. Neste sentido, uma vez efectuadas as transformações de translação, rotação e escala pretendidas antes do desenho dos polígonos, os vértices destes últimos serão na realidade desenhados no sistema de coordenadas transformado.

Nota: mude para uma nova folha de respostas

- d. **[2.0]** Uma das diferenças na utilização de `glOrtho()` versus `glFrustum()` reside no facto de a distância dos objectos em relação à câmara ser ou não perceptível. A que se deve esse efeito?

Uma projecção ortográfica (`glOrtho()`) define o volume de visualização como sendo um paralelepípedo, enquanto que uma projecção em perspectiva (`glFrustum()` ou `gluPerspective()`) define o referido volume como sendo uma pirâmide truncada. Desta forma, os objectos mais afastados da câmara ocupam, no caso da projecção ortográfica, a mesma proporção de espaço que os restantes; mas no caso da perspectiva, os objectos mais afastados ocupam um espaço proporcionalmente maior que os objectos mais próximos devido ao facto de a base da pirâmide ser mais larga do que o topo.

- e. **[3.0]** Pretende-se simular um foco de luz posicionado sempre por cima do personagem principal. Este desloca-se no plano OZX e a sua posição é dada por (`obj.x`, `obj.y`, `obj.z`). Transcreva para a folha da prova o seguinte extracto de código, completando-o de modo a configurar correctamente a referida fonte de luz. Pode definir constantes, se assim o entender.

```
float luzpos[] = {_____, _____, _____, _____};
float luzdir[] = {_____, _____, _____};
glLightfv(GL_LIGHT0, GL_DIFFUSE, luz_dif);
glLightfv(GL_LIGHT0, GL_SPECULAR, luz_spec);
glLightfv(GL_LIGHT0, _____, _____);
glLightfv(GL_LIGHT0, _____, _____);
glLightf(GL_LIGHT0, _____, _____);
```

```
float luzpos[] = {obj.x, obj.y + ALTURA_FOCO, obj.z, 1.0};
float luzdir[] = {0.0, -1.0, 0.0};
glLightfv(GL_LIGHT0, GL_DIFFUSE, luz_dif);
glLightfv(GL_LIGHT0, GL_SPECULAR, luz_spec);
glLightfv(GL_LIGHT0, GL_POSITION, luz_pos);
glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, luz_dir);
glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0); // qualquer valor
entre 0.0 e 90.0 graus
```

Nota: mude para uma nova folha de respostas

Sistemas Gráficos e Interação

Época de Recurso

2016-02-12

- f. **[2.0]** Suponha que necessita de criar a função `disco()`, que desenha um objecto correspondente a uma coroa circular (um círculo “subtraído” de outro círculo menor com o mesmo centro). Que tipo de primitiva utilizaria para conseguir desenhá-lo? **Justifique a sua opção.**



```
void disco(GLfloat xCentro, GLfloat yCentro,
           GLfloat raioMaior, GLfloat raioMenor,
           GLint nSegmentos)
{
    ...
    glBegin(_____);
    ...
}
```

Dado não se tratar de um polígono convexo, o OpenGL não o consegue desenhá-lo directamente, pelo que é necessário dividir o objecto em vários polígonos.

`GL_QUADS`, `GL_TRIANGLES`, `GL_QUAD_STRIP` ou `GL_TRIANGLE_STRIP`.

(`GL_POLYGON` também pode ser aceite como resposta, desde que se refira que os polígonos terão de ser triângulos ou quadriláteros).

- g. **[3.0]** Crie a função `mil_folhas()` que reutiliza a função anterior para desenhá-la. Assuma que a função anterior desenha o disco no plano OXY.

```
void mil_folhas(GLfloat x0, GLfloat z0,
               GLfloat rMaior, GLfloat rMenor, GLint nSegmentos,
               GLint nfolhas, GLfloat espacamento)
```

```
void mil_folhas(GLfloat x0, GLfloat z0, GLfloat rMaior, GLfloat rMenor,
               GLint nSegmentos, GLint nfolhas, GLfloat espacamento)
{
    glPushMatrix();
    glRotatef(90.0, 1.0, 0.0, 0.0); // também pode ser -90.0 graus
    for (GLint i = 0 ; i < nfolhas; i++)
    {
        disco(x0, z0, rMaior, rMenor, nSegmentos);
        glTranslatef(0.0, espacamento, 0.0);
    }
    glPopMatrix();
}
```

Nota: mude para uma nova folha de respostas

- h. **[2.0]** Escreva o código que permite realizar uma animação contínua do “mil folhas”, de modo a que o espaçamento entre discos comece em 0.1 e termine em 10.0 em incrementos de 0.1, após o que regressa a 0.1 em decrementos de 0.1, e assim continuamente.

```
void init()
{
    ...
    modelo.espacamento = 0.1;
    modelo.inc = 0.1;
    ...
}

void Timer(int value)
{
    glutTimerFunc(estado.delay, Timer, 0);
    modelo.espacamento += modelo.inc;
    if (modelo.espacamento <= 0.1 || modelo.espacamento >= 10.0)
    {
        modelo.inc *= -1.0;
    }
    glutPostRedisplay();
}
```

- i. **[2.0]** Suponha que pretende desenhar três “mil folhas” na mesma cena em que cada um tem uma animação diferente. Indique sucintamente como deveria estruturar o seu código.

Seria necessário definir uma estrutura `modelo` para cada um dos objectos (por exemplo, um `array` de três “mil folhas”. Na função `Timer()` seria necessário efectuar a lógica de animação de cada objecto, alterando o modelo correspondente.