

U.C. 21020

Computação Gráfica

25 de fevereiro de 2014

-- INSTRUÇÕES --

- O estudante deverá responder à prova na folha de ponto e preencher o cabeçalho e todos os espaços reservados à sua identificação, com letra legível.
- Sempre que não utilize o enunciado da prova para resposta, poderá ficar na posse do mesmo.
- No caso de provas com escolha múltipla, **sem grelha de resposta**, deverá indicar a resposta correcta na folha de ponto, indicando o número da pergunta e a resposta que considera correcta.
- No caso de provas com escolha múltipla, **com grelha de resposta, tabela e/ou espaços para preenchimento**, deverá efectuar as respostas no enunciado, pelo que o mesmo deverá ser entregue ao vigilante, juntamente com a folha de ponto, **não sendo permitido ao estudante levar o enunciado**.
- Verifique no momento da entrega da(s) folha(s) de ponto se todas as páginas estão rubricadas pelo vigilante. Caso necessite de mais do que uma folha de ponto, deverá numerá-las no canto superior direito.
- Em hipótese alguma serão aceites folhas de ponto dobradas ou danificadas.
- Exclui-se, para efeitos de classificação, toda e qualquer resposta apresentada em folhas de rascunho.
- Os telemóveis deverão ser desligados durante toda a prova e os objetos pessoais deixados em local próprio da sala de exame.
- A prova é constituída por 3 páginas e termina com a palavra **FIM**. Verifique o seu exemplar e, caso encontre alguma anomalia, dirija-se ao professor vigilante nos primeiros 15 minutos da mesma, pois qualquer reclamação sobre defeito(s) de formatação e/ou de impressão que dificultem a leitura não será aceite depois deste período.
- Utilize unicamente tinta azul ou preta.
- Responda às questões de forma clara, sucinta, e apresente todos os cálculos.
- Quando solicitado, apresente ainda uma representação gráfica do resultado final obtido na questão.
- A cotação de cada uma das questões é indicada junto do enunciado da mesma.
- A prova é **SEM CONSULTA**. Todos os elementos necessários à resolução são fornecidos no enunciado.

Duração: 90 minutos

QUESTÃO 1 (3 valores) (1.1 = 1.0; 1.2 = 2.0)

Recorra ao algoritmo *scan-line* para calcular as coordenadas dos pixels de preenchimento da área bidimensional definida pelo polígono constituído pelos vértices $A(4,1)$, $B(7,4)$, $C(4,7)$, $D(1,7)$ e $E(1,4)$

- 1.1. Apresente o estado da tabela de arestas (ET - *Edge Table*) e tabela de arestas activas (AET - *Active Edge Table*) no início do algoritmo.
- 1.2. Calcule as coordenadas dos pixels de preenchimento, apresentando cada iteração do algoritmo separadamente, indicando o estado da ET e AET, e apresente no final, de forma gráfica, o preenchimento.

(Resposta: 25 linhas)

QUESTÃO 2 (2 valores)

Dada a matriz de coeficientes geométricos B obtenha os vectores coeficientes algébricos a , b , c e d sendo que

$$a = [a_x \ a_y \ a_z], \quad b = [b_x \ b_y \ b_z], \quad c = [c_x \ c_y \ c_z], \quad d = [d_x \ d_y \ d_z].$$

Apresente todos os cálculos e matrizes(s) utilizada(s).

$$B = \begin{bmatrix} 0 & 0 & 1 \\ 0 & -1 & 0 \\ 1 & 0 & -2 \\ 0 & 1 & 1 \end{bmatrix}$$

(Resposta: 15 linhas)

QUESTÃO 3 (2 valores)

Admita que temos as seguintes funções de mistura Bézier.

$$B_0 = 4u - 1$$

$$B_1 = u^2 + u + 1$$

$$B_2 = -u^2 - 2u + 5 \quad \text{com } u \in [0, 1].$$

Sendo os três pontos de controlo $P_0 = (1, 0, 1)$, $P_1 = (-1, 0, 1)$, $P_2 = (-2, 1, 0)$ calcule o ponto $P = (p_x, p_y, p_z)$ da curva Bézier em $u = 1/2$

(Resposta: 15 linhas)

QUESTÃO 4 (1 valor)

Apresente e justifique uma técnica de representação de sólidos que recomendaria para modelação de uma família de peças mecânicas de automóvel.

(Resposta: 10 linhas)

QUESTÃO 5 (4 valores)

Considere o código abaixo. Explique sucintamente o que ele executa e descreva/comente o que cada linha de código faz:

```
public void init(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL().getGL2();
    glu = new GLU();
    gl.glClearColor(0.0f, 0.0f, 0.0f, 0.0f);
    gl.glClearDepth(1.0f);
    gl.glEnable(GL_DEPTH_TEST);
}

@Override
public void reshape(GLAutoDrawable drawable, int x, int y, int width, int height) {
    GL2 gl = drawable.getGL().getGL2();

    if (height == 0) height = 1;
    float aspect = (float)width / height;
    gl.glViewport(0, 0, width, height);

    gl.glMatrixMode(GL_PROJECTION);
    gl.glLoadIdentity();
    glu.gluPerspective(45.0, aspect, 0.1, 100.0);
    gl.glMatrixMode(GL_MODELVIEW);
    gl.glLoadIdentity();
}

@Override
public void display(GLAutoDrawable drawable) {
    GL2 gl = drawable.getGL().getGL2();
    gl.glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    gl.glLoadIdentity();

    gl.glTranslatef(0.0f, 0.0f, -6.0f);
    gl.glBegin(GL_TRIANGLES);
    gl.glVertex3f(0.0f, 1.0f, 0.0f);
    gl.glVertex3f(-1.0f, -1.0f, 0.0f);
    gl.glVertex3f(1.0f, -1.0f, 0.0f);
    gl.glEnd();
}
```

FIM