



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Computação Gráfica

AP1 - 2º semestre de 2019.

Nome –

Assinatura –

Observações:

- i) Prova sem consulta e sem uso de máquina de calcular.
 - ii) Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 - iii) Você pode usar lápis para responder as questões.
 - iv) Ao final da prova devolva as folhas de questões e as de respostas.
 - v) Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

Na última página encontra-se a folha de respostas. Preencha corretamente e sem rasuras. Todas as questões tem o mesmo peso.

1) Defina o que é uma superfície (1.0 ponto). Dê um exemplo de (**através de sua fórmula matemática**):

- uma superfície implícita (0.5 ponto)
- uma superfície paramétrica (0.5 ponto).

Uma superfície S é um objeto gráfico tal que a vizinhança de um ponto $p \in S$ tenha uma estrutura semelhante a um disco aberto (ou semi-disco aberto se a superfície tiver bordo).

Um exemplo de superfície implícita é dada pela equação da esfera:

$$f(x,y,z) = x^2 + y^2 + z^2 - r^2 = 0.$$

Um exemplo de superfície paramétrica é dada por um parabolóide de revolução definido por:

$$\begin{aligned}x &= u \cos(\theta) \\ y &= u \sin(\theta) \\ z &= u^2\end{aligned}$$

$$0 \leq \theta < 2\pi, u \in \mathbb{R}^2.$$

- 2) Considere um sólido B definido implicitamente por uma função $f(x,y,z) = 0$. Explique como, dado um ponto p com coordenadas (x_p, y_p, z_p) , é possível verificar se p está dentro, fora ou sobre a superfície de B (2.0 pontos).

Se o sólido B é definido por uma função $f(x,y,z) = 0$ então, isso significa que $f(x,y,z) = 0$ é a fronteira que separa seu interior do restante do espaço. Sabe-se que para uma função implícita f , dado um ponto p com coordenadas (x_p, y_p, z_p) , se $f(x_p, y_p, z_p) < 0$, então p é interior ao sólido delimitado por f . Caso $f(x_p, y_p, z_p) > 0$ afirma-se que p é externo a B . Quando $f(x_p, y_p, z_p) = 0$ o ponto está exatamente na fronteira.

- 3) Suponha que você possui um objeto O no plano definido por um polígono e deseja fazer uma animação, onde o objeto percorre uma trajetória dada por uma curva suave. Considerando que a curva é especificada por um polígono de controle com n pontos $p_0, p_1, p_2, \dots, p_{n-1}$, mostre como construir tal curva usando uma B-Spline cúbica (2.0 pontos).

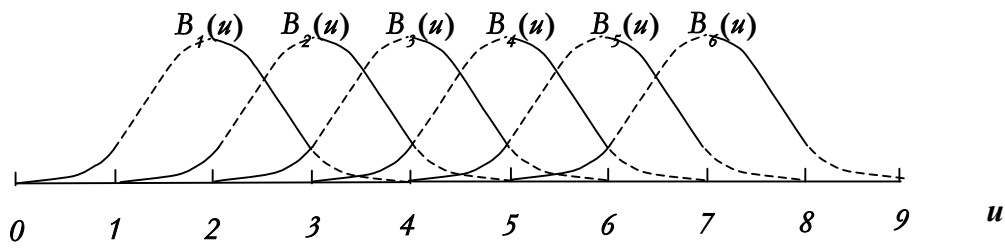
Uma B-Spline é uma curva paramétrica formada por segmentos de curva. Logo, dados n pontos de controle, é possível construir a trajetória do objeto através de uma equação no

parâmetro u do tipo
$$Q(u) = \sum_{i=0}^{n-1} p_i B_i(u).$$

Para cada ponto de controle existe uma função $B_i(u)$ de grau $k = 3$ que satisfaz:

$$B_i(u) = \begin{cases} x = 0, u < t_i \vee u > t_{i+k+1}, \\ x \neq 0, t_i \leq u \leq t_{i+k+1} \end{cases},$$

onde os valores t_i são os valores dos nós, isto é os valores no parâmetro u associados a cada um dos nós, que são os pontos onde os segmentos se juntam. No exemplo abaixo, para um polígono com 6 pontos de controle, temos 6 bases e 10 valores de nós, de forma que todo ponto de controle está associado a uma função de base definida sobre 4 intervalos consecutivos de u . Por exemplo, a base $B_0(u)$ tem como domínio os intervalos entre os valores de nó (0,1,2,3 e 4).



A multiplicidade dos valores de nós, isto é, o quanto eles se repetem, regulam a forma das funções $B_i(u)$ que são definidas pela seguinte expressão de recorrência:

$$B_{i,1}(u) = \begin{cases} 1, & u_i \leq u < u_{i+1} \\ 0, & \text{caso contrário} \end{cases}$$

$$B_{i,k}(u) = \frac{u - u_i}{u_{i+k-1} - u_i} B_{i,k-1}(u) + \frac{u_{i+k} - u}{u_{i+k} - u_{i+1}} B_{i+1,k-1}(u)$$

Uma vez definida a função $Q(u)$ pode-se ajustar os pontos de controle de forma a delinear a curva correspondente a trajetória que o objeto deve seguir.

- 4) Explique o que significa afirmar que a API da OpenGL funciona como uma máquina de estados. Dê um exemplo (2.0 pontos).

Afirmar que a API da OpenGL funciona com uma máquina de estados significa que ela possui um conjunto de propriedades globais, os estados, que permanecem em vigência até que sejam modificados por alguma chamada da API, que altere esses estados. Por exemplo, a cor utilizada para limpar o fundo é um estado da OpenGL que pode ser alterado usando a função `glClearColor(...)`.

- 5) O que é uma *callback* em GLUT/OpenGL? Mostre um exemplo de uma *callback* de desenho (*display callback*) (2.0 pontos).

Uma *callback* é uma função escrita pelo programador que é registrada, no sistemas baseados em programação orientada a eventos, como APIs ou frameworks gráficos. Estas funções são invocadas quando ocorrem eventos que elas devem tratar.

Na API de interface da GLUT, por exemplo, existem diferentes *callbacks* que são chamadas para tratar eventos de pressionar e liberar o botão do mouse, mover o mouse, redesenhar a tela, redimensionar a tela, pressionar uma tecla do teclado e de evento nulo (*idle*).

Existem funções próprias na API para registrar (ou cadastrar) as *callbacks*, que recebem ponteiros ou referências para estas.

A seguir é apresentada uma implementação de uma *callback* para *display*, cadastrada através da função `glutDisplayFunc((void)(*func)(void);`

```
static void display(void)
{

    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glColor3d(1,0,0);

    glPushMatrix();
        glTranslated(2.4,-1.2,-6);
        glRotated(60,1,0,0);
        glRotated(a,0,0,1);
        glutWireTorus(0.2,0.8,slices,stacks);
    glPopMatrix();

    glutSwapBuffers();
}
```