Bacharelado em Ciência da Computação
Departamento de Computação de Sorocaba – UFSCar

Computação Gráfica – Terceira Lista de Exercícios

1) Implemente uma função que retorne o produto vetorial de dois outros vetores dados.
2) Implemente uma função que retorne o *versor* associado a um vetor de entrada.
3) Implemente uma função que considere duas bases para o espaço tridimensional (fornecidas pelo usuário). A primeira pode ser a base canônica para o espaço Euclidiano. A segunda, uma base ortonormal arbitrária. Dado um ponto do espaço, representado em relação a primeira base, a função deve retornar as coordenadas do ponto em relação a segunda base.
4) O que você entende por observador virtual? Exemplifique.
5) Qual a relação entre o observador virtual e o que chamamos de plano de visualização?
6) Considere o modelo do *pipeline de visualização* conforme discutido em sala de aula. Construa um objeto 3D centralizado na origem de um SRU para o espaço tridimensional (a forma do objeto é deixada a critério do aluno). Assuma que o sistema de coordenadas de observação (ou de visualização), para este exercício, coincida com o sistema de coordenadas do mundo. Implemente uma função que, dados o ponto a ser projetado, o centro de projeção e a localização do plano de visualização, retorne as coordenadas do ponto projetado. Você deve considerar o modelo de projeção perspectiva e que o centro de projeção esteja sobre o eixo perpendicular ao plano de visualização. Neste caso, o plano de observação, ou visualização, será paralelo ao plano formado pelos dois primeiros vetores da base que define o sistema de coordenadas de visualização (não necessariamente coincidente). Faça diversos testes com diferentes posições para o centro de projeção e para diferentes localizações do plano de projeção.
7) No exercício anterior, uma vez projetado o objeto no plano de visualização, implemente uma função para exibir o resultado na tela de um dispositivo gráfico matricial de saída. Considere uma janela de exibição adequada para que todo o objeto seja mostrado na tela do computador.
8) Considerando os dois exercícios anteriores, antes da projeção do objeto no plano de visualização, implemente uma função que rotacione o mesmo em relação aos três eixos do *sistema de coordenadas do mundo*, onde os ângulos de rotação são dados pelo usuário.

Após as rotações, utilize as rotinas implementadas nos exercícios anteriores para projetar o objeto considerando o sistema de coordenadas de visualização e finalmente exibir o resultado na tela do computador. Note que o sistema de coordenadas de visualização ainda coincide com o sistema do mundo para os exercícios acima.

9) Agora, generalize as rotinas anteriores para um sistema de coordenadas de visualização arbitrário, onde para descobrir esse sistema, o usuário deve fornecer o vetor normal ao plano de visualização e o vetor "view up". Deve ser fornecido também, como entrada, a posição do observador virtual (ou do próprio plano de visualização). Esse exercício corresponde ao segundo estágio do *pipeline de visualização* visto em aula. Exiba o resultado da projeção perpectiva do objeto definido nos exercícios anteriores considerando um sistema de coordenadas de visualização criado por você.
10) Em sala de aula derivamos as expressões para as coordenadas de um ponto projetado através do modelo da projeção perspectiva. Descreva como é a matriz que define a transformação linear da projeção perspectiva considerando coordenadas homogêneas.
11) (Opcional) Implemente três funções para a visualização de um objeto tridimensional considerando as projeções planares paralelas ortográficas do tipo planta, vista frontal e vista lateral, respectivamente. Cada função deve receber como entrada a posição do plano de projeção.
12) (Opcional) Considere um sistema de *coordenadas cartesiano tridimensional* associado a um SRU e implemente uma função para determinar pontos arbitrários de uma curva representada por $P(t) = (x(t),y(t),z(t))$, onde $0 \leq t \leq 1$. Para isso, considere a abordagem das curvas de *Bézier*. O usuário deverá entrar com o número de pontos de controle e as coordenadas de cada um destes pontos, respectivamente. Considere na implementação os casos onde os pontos de controle são igualmente espaçados e depois o caso onde os pontos dados pelo usuário não possuem um espaçamento uniforme. Tente descrever curvas com regiões de curvatura elevada. A saída da função deve ser apenas o valor das coordenadas da curva no ponto desejado. Avalie sua função para diversas entradas.
13) (Opcional) Da mesma forma que no exercício anterior, implemente uma função para determinar pontos arbitrários de uma curva representada por $P(t)=(x(t),y(t),z(t))$. Contudo, considere a abordagem das curvas *B-Slines*. O usuário deve entrar com o número de pontos de controle e as coordenadas desses pontos, respectivamente, assim como o conjunto de nós (t_0, t_1, …) da curva. O parâmetro k, que controla a ordem de continuidade da curva, também deve ser dado pelo usuário. Como antes, avalie sua função considerando pontos equidistantes e pontos de controle que não sejam uniformemente espaçados. A saída da

função deve ser apenas o valor das coordenadas da curva no ponto desejado.

14) (Opcional) Implemente uma função que determine um ponto do espaço que pertence a uma superfície (imagine uma!). Considere a abordagem das superfícies de *Bézier*.
15) (Opcional) Repita o exercício anterior considerando a abordagem das superfícies *B-Spline*.

Observações:

a) Os exercícios de 1 a 10 não são novidades, sendo que todos foram pedidos em sala de aula, alguns com muita antecedência, como o primeiro, por exemplo;

b) As implementações para os exercícios opcionais desta lista não serão objeto de prova, contudo os conceitos associados aos exercícios podem ser cobrados;

c) Em todos os exercícios, os resultados devem ser armazenados na estrutura que chamamos em aula de aula de *buffer de visualização*. Esse buffer (matriz) é que deverá ser passado para uma rotina específica para exibição na tela do computador.