MAC0420/5744 – Introdução à Computação Gráfica
1o Semestre de 2011
Segundo exercício programa - Simulador de vôo

Data de entrega: sábado, 28/05/2011, até 23:55h

Esse exercício pode ser feito por grupos de até 2 alunos de MAC0420 e é individual para os alunos de MAC5744

1. Objetivo

Utilizar as rotinas gráficas tridimensionais interativas do OpenGL e GLUT para a implementação de um simulador de vôo simples. Existe um programa executável em nossa página que você pode utilizar para melhor compreender os requisitos desse projeto, bem como todos os arquivos citados nesse enunciado.

2. O que entregar

Os fontes e o makefile para o seu programa no formato tgz. Você deve se assegurar que seu programa compile e rode na rede Linux do IME.

3. Descrição do projeto

Você deve implementar um simulador de vôo. O simulador mostra a cena do ponto de vista do piloto. A aeronave é controlada pelo teclado e mouse da seguinte forma:

- teclas a/s: aumenta/diminui a velocidade da aeronave por um montante fixo [5m/s]. Você não precisa implementar um simulador de vôo real. Por exemplo, apertando repetidamente a tecla 's', a aeronave para, podendo inclusive voar para trás.
- teclas q: (quit) encerra o programa.
- A coordenada (x,y) do mouse controla a direção da aeronave (yaw e pitch)[1]. Imagine um sistema de coordenadas centrado no piloto, com x, y, e z apontando para a direita, para cima, e para trás respectivamente. *Roll* é uma rotação ao redor do eixo z, *yaw* é uma rotação ao redor do eixo y, e *pitch* é uma rotação ao redor do eixo x do piloto. Defina a velocidade de yaw como sendo um número de graus por segundo. Mapeie a coordenada x do mouse no intervalo [-1, 1], e então multiplique a coordenada por 10 graus por segundo. Isso define a velocidade de yaw instantânea. Para obter a velocidade de pitch, mapeie a coordenada y do mouse para o intervalo [-1,1] e multiplique esse valor por 10 graus por segundo. Note que movendo o mouse para a direita causa a nave a se mover para a direita, ou seja, a cena parece se mover para a esquerda,e quando o mouse é movido para baixo, a inclinação da aeronave aumenta (ela sobe), causando um movimento da cena para baixo.
- teclas z/x/c: As teclas 'z', 'x', e 'c' controlam a velocidade de roll, 'z' aumenta a velocide em 2 graus por segundo no sentido anti-horário, 'c' aumenta a velocidade em 2 graus por segundo no sentido horário, e 'x' interrompe a rotação instantaneamente.
- Botão direito do mouse: gera uma pausa no programa, que volta a rodar quando esse botão é pressionado novamente.
- Botão esquerdo do mouse: quando em pausa, esse botão executa um passo (single step) da animação, imprimindo o intervalo transcorrido, e a posição, direção de translação, e velocidades da nave. Se o usuário demorar muito a pressionar essa tecla, os efeitos da animação são difíceis de notar, portanto você pode considerar um intervalo de tempo máximo de 1 segundo, ou seja, se o intervalo passar de um segundo, você gera uma nova cena como se apenas um segundo tivesse se passado.
- PARTE OPCIONAL: tecla v: simule múltiplas aeronaves, e popule a sua cena com diversas delas (lembre-se de seguir o nosso formato dos arquivos de entrada). Quando a tecla v é pressionada, o simulador passa a

[1] Use a rotina de callback para passive motion

mostrar a cena do ponto de vista de outro aeroplano (a versão executável não possue esse recurso, apenas o básico), ciclicamente, até voltar à primeira nave. Considere que os demais aeroplanos ficam "parados", e portanto as velocidades se aplicam apenas ao aeroplano sendo simulado.

- PARTE OPCIONAL: tecla `b`: ao pressionar essa tecla, ao invés de gerar uma cena vista pelo piloto, crie uma vista alternativa, colocando a câmera a 50 metros atrás da aeronave. Ao pressionar `b`, novamente, a vista passa para um ponto 100m atrás da aeronave, e ao terceiro toque, volta a ser centrado no piloto.

4. Terreno

A cena consiste de uma ilha flutuando no meio de um oceano azul. O fundo recebe a cor do céu. O oceano é modelado como um retângulo azul, horizontal e muito grande. No exemplo executável, as coordenadas globais foram definidas com relação ao plano do oceano (+x leste e +y norte) e o eixo z aponta para cima. As primeiras duas linhas do arquivo de entrada contem os valores mínimos e máximos para as coordenadas x e y do retângulo do oceano. A coordenada z do retângulo é zero (veja o arquivo input.txt).

A ilha é modelada através de um mapa de elevação, ou seja, uma matriz bidimensional (coordenadas x,y) de elevações (valores de z). A ilha é definida dentro do plano x,y pelo retângulo (xLandMin, xLandMax, yLandMin, yLandMax), que é composto portanto por (xint=xLandMax-xLandMin, yint=yLandMax-yLandMin) elementos em uma grade retangular, e portanto a ilha possue m = (xint + 1)×(yint+1) vértices. A elevação desses m elementos são então definidos, uma linha de cada vez, a partir do canto inferior esquerdo ao canto superior direito da ilha.

O terreno é mostrado subdividindo-se cada elemento da grade em dois triângulos, traçando uma diagonal do canto inferior esquerdo ao canto superior direito do elemento de grade. No exemplo executável, a cor de um vértice é baseada em sua elevação. Veja o arquivo proj2.txt para as cores utilizadas no arquivo executável.

Eu sugiro que você depure o seu programa inicialmente sem nenhuma iluminação (use apenas a cor específica de cada objeto). Quando você incluir a iluminação, você pode criar uma única fonte de luz (o sol) na posição (0, 0, ∞). Você pode utilizar tanto sobreamento flat ou smooth para os polígonos de sua ilha (sombreamento simples é cerca de 20% mais rápido).

5. Posição Inicial da Aeronave

Após a definição do terreno, o arquivo de entrada também fornece o número de aeronaves e suas posições iniciais e orientações. Para o projeto básico, você só precisa simular a primeira aeronave . O primeiro número é o número de aeronaves. Esse número é seguido por um conjunto de linhas, uma linha para cada aeronave. Os primeiros 3 números de uma linha definem as coordenadas (x, y, z) da aeronave (em coordenadas globais). Os próximos 3 ângulos definem os ângulos de Euler (θ_x,θ_y,θ_z) que definem a orientação dessa aeronave, em graus. Esses ângulos definem três rotações: em x, y, e z respectivamente (positivos no sentido anti-horário).

Após essas rotações, o plano está apontando na direção -z no frame resultante, com o eixo y apontando para cima (no sistema de coordenadas do piloto). Por exemplo, os ângulos '90 0 0' resultariam em um plano apontando no sentido do eixo +y, com a coordenada global +z para cima. O último número na linha é a velocidade inicial (para frente em metros por segundo). As velocidades iniciais das rotações de de pitch, yaw e roll são zeradas. O arquivo input.txt contem um exemplo de um arquivo de entrada.

6. Dicas para a implementação:

Existem várias formas possíveis para se modelar uma nave, eu modelei da seguinte forma. Para armazenar a posição, direção de translação e orientação da nave eu defini um frame de coordenadas centrado no piloto. Nessa estrutura também são armazenadas todas as velocidades, que são modificadas pelo teclado e mouse. Para cada "idle event", primeiro a translação da nave é calculada, e a seguir sua nova orientação, baseada nas velocidades de rotação. Como essas transformações afetam o estado de navegação da nave, você provavelmente terá de escrever essas funções, ao invés de utilizar as funções do OpenGL, que só afetam o desenho. Uma vez realizada todas as transformações, teremos um novo frame de coordenadas, que então é utilizado para dizer ao OpenGL (provavelmente através do gluLookAt) como gerar a nova cena.

Será necessário utilizar o depth-buffer para remover as superfícies escondidas. Para isso, você terá de incluir GLUT_DEPTH nos argumento de entrada da função glutInitDisplayMode(), e ao chamar glClear(), não se esqueça de incluir GL_DEPTH_BUFFER_BIT (além do GL_COLOR_BUFFER_BIT).

Para a implementação da parte adicional, você pode desenhar um (ou alguns) triângulos coloridos para cada nave, com um vértice na posição da nave, e um eixo de simetria ao longo de z. Por exemplo, dois triângulos ortogonais fornece a impressão de um avião com cauda.