



Curso: *Desenvolvimento de Jogos e Entretenimento Digital – GT JEDi*
Disciplina: *Projeto de Jogos – Desenvolvimento Rápido de Jogos 3D (DBPro)*
Horário: 34 / 54 - Semestre: 2006/2

Prof.: Fernando Osório
E-mail: fosorio@unisinos.br
Web: <http://inf.unisinos.br/~osorio/jogos-gt.html>

DATA:
24/10/2006

ATIVIDADE INDIVIDUAL – ESTUDO DIRIGIDO

Exercício: Uso de Arquivos

Atividades a serem realizadas na Disciplina de Projeto Rápido de Jogos 3D junto ao Laboratório de Informática (LAPRO). Realize o seguinte trabalho durante o período de aula, buscando executar as atividades descritas a seguir:

1. Crie um programa para ler um arquivo que descreve a posição e tamanho de objetos a serem criados em um ambiente 3D.

Sugestão: crie um programa que leia sempre conjuntos de 4 linhas de um arquivo texto, contendo: coordenada X na primeira linha, coordenada Y na segunda linha, coordenada Z na terceira linha e tamanho T do objeto na quarta linha. Exemplo de arquivo:

Arquivo: Objetos.txt

10
20
30
40
30
30
30
10

Lendo o arquivo acima, o programa deve criar 2 objetos:

- Objeto 1: Coordenada (10, 20, 30) com Tamanho 40
- Objeto 2: Coordenada (30, 30, 30) com Tamanho 10

Sugestão: Crie esferas ou cubos com a posição e tamanhos especificados... e use os seguintes comandos no logo início do programa:

```
autocam off : position camera 0,25,-30  
point camera 0,0,0 : make matrix 1,100,100,10,10
```

Se você quiser pode obter um exemplo do arquivo com diversos objetos definidos nele e no formato descrito acima em:

<http://www.inf.unisinos.br/~osorio/rgd/dados/objetos.zip>

[Questão para responder na próxima aula: qual o objeto desenhado no arquivo “objeto1.txt”]

2. Baseado no programa anterior faça um programa que crie um cenário (pode ser lendo um arquivo BSP) e coloque sobre este cenário diversos objetos (podem ser objetos .X), onde as coordenadas e orientação destes objetos são lidas de um arquivo texto. O arquivo texto deve conter os dados referentes a posição (X, Y, Z) e orientação (rotação em Y) dos objetos.

A idéia deste programa é que você crie o cenário de um jogo: (escolha uma das opções abaixo)

A) Onde existem objetos a serem capturados... escondidos em posições previamente determinadas, posições estas que estão armazenadas no arquivo texto.

B) Onde existem objetos que devem ser coletados... os objetos devem estar dispostos em seqüência e se seguir um caminho, mostrando o caminho que devemos seguir e que está indicado por estes objetos. Os caminho deve levar para algum lugar especial do cenário. Os objetos tem sua posição lida de um arquivo texto.

Sugestões:

- Carregue o BSP do blackbridge ou do 20kdm2, disponíveis em:
<http://www.inf.unisinos.br/~osorio/rgd/Aulas/BSP/pk3/> ou
no AVA na comunidade “PRES-GRAD-Projeto de Jogos: Ferramentas de Desenvolvimento Rápido de Jogos” em:
Arquivos coletivos / Semestre-2006-2 / Aula-BSP-Out03 / pk3

- Use o programa “Explore-BSPv1”, que permite explorar o ambiente BSP e salvar em um arquivo texto algumas posições específicas selecionadas neste ambiente. Este programa está disponível no site “inf” e no AVA, junto com o material da aula de BSP.

3. Se sobrar tempo: crie uma “visita guiada” em um arquivo BSP. Crie um novo programa onde o movimento da câmera seja controlado por uma lista de coordenadas (pontos chave) indicados em arquivo texto...

A) O arquivo texto deve conter uma seqüência de coordenadas X, Y e Z. Cada coordenada é uma posição destino da câmera, onde devemos poder avançar de um ponto a outro sem bater em nenhum obstáculo. Os pontos-chave (key positions) devem descrever retas e curvas adequadamente de modo a evitar colisões;

B) Inicie na primeira coordenada, posicionando a câmera nesta posição e orientando ela de acordo com a posição do próximo ponto indicada no arquivo. Sendo assim a câmera está localizada no POSITION CAMERA das coordenadas X, Y e Z atuais e com o POINT CAMERA para as coordenadas X, Y e Z do próximo ponto;

C) Com o comando MOVE CAMERA desloque a câmera lentamente de modo que ela vá em direção ao próximo ponto (para o qual ela está direcionada). Quando você estiver bem próximo do ponto destino, ou seja, a posição atual da câmera está a uma distância em X e em Z (e opcionalmente em Y) muito próxima da coordenada destino (menor que um certo limite, que deve ser levemente maior que o passo do deslocamento usado no move!), então você deve posicionar a câmera sobre o ponto destino e recomeçar o movimento em direção ao próximo ponto, conforme indicado no passo B.

Repita os passos acima até que não tenham mais coordenadas para ler no arquivo.

4. Se mesmo assim ainda sobrar tempo livre... faça um pequeno visualizador gráfico de modelos compostos com diferentes objetos posicionados (posição), orientado (rotação) e dimensionados (escala) de acordo com as informações descritas em um arquivo texto. O formato do arquivo texto poderá ser da seguinte forma:

A primeira linha do arquivo descreve o tipo do objeto (string)

<objeto>

As 3 linhas seguintes do arquivo descrevem a posição do objeto (3 números = posição)

<Coordenada_X> <Coordenada_Y> <Coordenada_Z>

As 3 linhas seguintes do arquivo descrevem a orientação do objeto (3 números = ângulo)

<Angulo_X> <Angulo_Y> <Angulo_Z>

As 3 linhas seguintes do arquivo descrevem a escala do objeto (3 números = tamanho)

<Escala_X> <Escala_Y> <Escala_Z>

Note que na primeira linha podemos ter objetos simples como por exemplo:

“esfera”, “cubo”, “cilindro”, etc.

Caso seja indicado um nome de um objeto que não é simples, ele será considerado como um objeto composto (.x) e o nome lido será adotado como o nome do arquivo que contém este objeto. Exemplo:

```
read string arquivo,objeto$
if objeto$="sphere"
  make object sphere numobj,EscalaX#
else if objeto$="box"
  make object box numobj, EscalaX#, EscalaY#, EscalaZ#
else
  load object objeto$,numobj
  scale object numobj, EscalaX#, EscalaY#, EscalaZ#
endif
endif
```

Note que a esfera usa a EscalaX# para definir seu raio, pois sua escala é a mesma nas três dimensões. O mesmo deve ocorrer com o cubo, mas no entanto na caixa ou em objetos compostos as escalas são aplicadas para cada uma das dimensões do objeto.

Este programa permite criar todo um cenário a partir de uma descrição contida em um arquivo texto!

Bom trabalho e até a próxima aula!