



Curso: *Desenvolvimento de Jogos e Entretenimento Digital – GT JEDi*

Disciplina: *Projeto de Jogos – Desenvolvimento Rápido de Jogos 3D (DBPro)*

Horário: 34 / 54 - **Semestre:** 2007/2

Prof.: Fernando Osório

E-mail: fosorio@unisinos.br

Web: <http://inf.unisinos.br/~osorio/jogos-gt.html>

DATA:

18/10/2007 (Quinta / 54)

23/10/2007 (Terça / 34)

ATIVIDADE INDIVIDUAL – ESTUDO DIRIGIDO

Exercício: Cenários, Agentes e Colisões

Atividades a serem realizadas na Disciplina de Projeto Rápido de Jogos 3D junto ao Laboratório de Informática (LAPRO). O prof. João Bittencourt estará no Lapro, e pode ajudá-los caso seja necessário. Realize o seguinte trabalho durante o período de aula, buscando executar as atividades descritas a seguir, cujo objetivo principal é realizar um exercício prático de programação com o DBPro, aplicando os conceitos vistos nas aulas anteriores (Cenários, Agentes e Colisões).

Realize o seguinte trabalho durante o período de aula, buscando executar as atividades descritas a seguir:

1. Crie um cenário inicial composto de um matrix, e adicione uma textura no solo, dando uma aparência mais interessante para o “chão” do terreno onde irá se desenvolver a ação desta implementação.

1.1. *Opcional:* considere a possibilidade de dar um relevo ao terreno, alterando a elevação deste terreno para que possamos ter um plano irregular.

2. Delimite o espaço do matrix, criando limites em torno dos quatro cantos do mesmo. Você pode adicionar 4 objetos do tipo box, ao redor deste matrix. O nosso objetivo é o de usar estes 4 cantos (paredes no limite do matrix) para evitar que o usuário “saia fora” da área de jogo, definida pelo matrix, testando a colisão contra estes objetos.

2.1. *Opcional:* adicione um skybox, para dar uma melhor noção de orientação no espaço. Você pode optar por usar um skybox do tipo cubo ou esfera. Este skybox poderá ficar além dos limites do matrix, e neste caso deveremos usar objetos ocultos (*hide object*) para delimitar o espaço de movimentação dentro do matrix. Exemplo: usamos um skybox do tipo esfera que é bem mais amplo do que o matrix, envolvendo este com uma certa folga, e criamos as 4 paredes que delimitam a movimentação, colocadas exatamente nas extremidades do matrix, porém com objetos ocultos (*hide*).

3. Crie um objeto controlado pelo usuário, podendo ser um personagem ou mesmo um veículo (use os objetos da biblioteca do DarkMatter). Este objeto-ator deve ser controlado pelo usuário e será visualizado em 3ª. pessoa, com uma câmera acoplada ao objeto-ator. A câmera segue o objeto, podendo ser usados os comandos vistos em aula: *newxvalue*, *newzvalue*.

3.1. Lembre-se de que o objeto-ator deve ter uma detecção e tratamento de colisões contra os limites do ambiente: detectamos a colisão contra as paredes e evitamos que o objeto passe através destas paredes que delimitam o terreno (matrix).

3.2. *Opcional*: a câmera também poderá usar o modo de detecção e tratamento de colisões ativado.

3.3. *Opcional*: se for usado um personagem humanóide, coloque uma animação neste objeto-ator. Faça com que seja simulado o caminhar (movimento) quando este objeto estiver se deslocando no cenário. Para animar o personagem, utilize um dos objetos que possuem animação da biblioteca do DarkMatter.

3.4. *Opcional*: se for usado um objeto do tipo veículo, lembre-se de tentar fazer o movimento dele mais realista, acelerando e desacelerando gradualmente o veículo. Faça isto de forma que ele não tenha sempre uma velocidade constante, com aceleração e freio instantâneos.

4. A partir da implementação descrita nos itens acima, implemente uma “coleta” de objetos que serão espalhados aleatoriamente neste ambiente. Utilize as funções de colisão para detectar quando o objeto-ator colide com um objeto do cenário a ser coletado. Quando um objeto for “coletado” (ou “atropelado”):

- Emitir um som indicando que o usuário capturou o objeto;
- Ocultar e/ou remover o objeto que modo que ele não se encontre mais visível (e nem passível de sofrer novas colisões). Dica: um objeto pode se movido para uma nova posição completamente fora do ambiente onde transcorre a ação do jogo, e deixado invisível;
- Incrementar o contador de objetos recolhidos, atualizando na tela esta informação (status), de modo que o usuário possa sempre saber quantos objetos já recolheu, e opcionalmente, quantos objetos tem ao total e/ou quantos ainda falta recolher;

Opcional: Se você adicionou um certo relevo no terreno, lembre-se de realizar o deslocamento do objeto-ator, de acordo com a elevação do terreno! Lembre-se também que ao criar os objetos a serem coletados, de colocar estes espalhados no terreno sempre considerando a elevação local do ponto onde está sendo inserido o objeto.

Extensões que podem ser realizadas caso sobre tempo para serem implementadas:

5. Crie dois tipos de objetos: os objetos com pontuação positiva e objetos com pontuação negativa. Quando ocorrer uma colisão com o objeto de pontuação positiva, o usuário ganhará pontos extras, e quando ocorrer uma colisão com um objeto de pontuação negativa, o usuário perderá pontos. Exemplo: implemente funções que permitam aumentar o *score* final do usuário e também descontar pontos de vida deste usuário.

6. Implemente o disparo de um tiro, que mesmo a distância poderá atingir um objeto e destruir (capturar) este objeto. Disparar um tiro consiste de:

- Criar um evento que ativa o tiro: sai da posição atual onde se encontra o objeto-ator, e vai na direção que este está apontando;
- O tiro deve estar em um dos dois estados: inativo (oculto, posicionado sem perigo de colisão) ou ativo (visível, se movendo e detectando colisão);
- Um tiro deve ter um “tempo de vida”: o tiro avança e se não colidir com nada depois de um certo tempo (ou distância), ele passa novamente a invisível e inativo. Se colidir, tem que atualizar a contagem de pontos e destruir/coletar o objeto atingido;
- Note que neste tipo de implementação, só temos um tiro de cada vez. Para o usuário disparar um novo tiro é necessário que o tiro anterior passe de ativo para inativo para que possa ser “reaproveitado”. Pense em como implementar a possibilidade de múltiplos tiros e rajadas de metralhadora.

7. Adicione uma trilha sonora ao jogo.

8. Adicione uma tela de abertura com o nome do jogo e os créditos (autoria).

9. Pode jogar o seu jogo a vontade, mas só para testar e ajustar o “Game Play” ;-)

Divirta-se!

=====

Bom trabalho nesta quinta e na próxima terça-feira. Aproveite para aperfeiçoar seu jogo!
Até a semana que vem, em nossa próxima aula na quinta-feira que vem (25/10)!

Prof. Fernando Osório
Email: fosorio@gmail.com