

UM SISTEMA INTEGRADO DE MODELAGEM 3D COM MOTOR DE JOGO - MIRAGEM3D

Aluno: Antonio Prates
Orientador: Bruno Feijó

Introdução

Nesta pesquisa, foi feito um estudo transversal entre diversos motores de jogos 3D (3D *game engines*) [1] para visualização de ambientes 3D de projetos de engenharia, propondo-se o 3dsmax [2] como ambiente de modelagem. As tecnologias de jogos sem fins meramente lúdicos, mas também usados em educação e simulação, caracterizam o novo paradigma de *serious games* (jogos sérios ou empresariais).

Os motores de jogos existentes no mercado (*e.g.*: RenderWare, Unreal, 3D GameStudio, Crystal Space, etc.) são muito caros, inacessíveis ou, então, limitados no que diz respeito à importação de modelos 3D e à visualização de cenários em céu aberto. No presente trabalho, optou-se pela utilização de um *framework* mais genérico, porém mais adequado para superar as limitações acima citadas: Ogre3D [3]. Este *framework* é reconhecidamente um motor gráfico 3D de alto nível, com a vantagem de ser em código aberto e possuir licença GPL, além de ter sido o projeto do mês no SourceForge.net em março de 2005. O Ogre3D não é um motor de jogos propriamente dito, pois não implementa física nem outras funcionalidades utilizadas na confecção de jogos, porém é muito utilizado pela comunidade desenvolvedora de jogos, porque possui uma arquitetura organizada, bem documentada e totalmente orientada a objetos - o que facilita a sua integração e adaptação.

O Ogre3D possui um gerenciamento flexível para diversos tipos de cenário, inclusive para cenários grandes em céu aberto, e dispõe de uma série de projetos desenvolvidos pela comunidade de usuários (de código aberto e licença GPL), tais como: exportação de malhas em diversos ambientes de modelagem (3dsmax, maya, blender, etc.) e integração com bibliotecas de colisão e física (ODE, OPCODE, etc.).

Baseado neste motor de jogos, foi implementado um simulador, denominado Miragem3D, que obtém imagens de alta qualidade de cenários modelados no 3dsmax e possibilita a interação do usuário em tempo real, com vistas a aplicações em engenharia.

Objetivos

Este trabalho objetiva estudar e programar um sistema para visualização de cenários do 3dsmax, com interação em primeira pessoa, baseado em um motor de jogo de código aberto e licença GPL.

Metodologia

Para exportar as malhas no 3dsmax e carregá-las no ambiente do Ogre3D, destacam-se dois exportadores (*plugins* para 3dsmax escritos em MaxScript):

- Ogre3DSExporter – que, embora seja a solução oficial para o Ogre3D, não atende os objetivos, pois exporta apenas uma malha individualmente e não um cenário completo.
- Octopus Exporter – capaz de exportar múltiplas malhas e gerar um arquivo XML de referência (DotScene) para o cenário. Este exportador apresentava uma série de pequenas falhas que foram parcialmente resolvidas ao longo do desenvolvimento deste projeto.

Uma das possibilidades para o carregamento do cenário é a geração de um arquivo binário de Octree a partir do arquivo DotScene exportado do ambiente de modelagem.

Entretanto, a implementação da biblioteca de Octree está com código desatualizado (desde 2002), o que a torna incompatível com as versões atualizadas do Ogre3D; não podendo, portanto, ser corrigida nem adequada para o uso neste projeto, por falta de documentação.

O Miragem3D implementa um *parser* para o arquivo XML (DotScene) gerado pelo Octopus, e um *loader* que utiliza o *framework* do Ogre3D para apresentar a cena ao usuário.

A aplicação-exemplo (Fig.1) utiliza um cenário do Cais do Porto do Rio de Janeiro (cujos arquivos 3dsmax foram cedidos pela empresa Nigraph) e um *Cubemap* (6 imagens JPEG que formam um cubo 360 graus com horizonte e nuvens) gerado no software *Terragen*.



Fig.1 Cais do Porto do Rio de Janeiro no Miragem3D

O posicionamento da câmera e a otimização para o carregamento de cenários grandes acontecem através de regras de nomeação de objetos no 3dsmax que servem de *triggers* para o *parser* de DotScene do Miragem3D. Um destes *triggers* é o nome de objeto “__player”, que é um objeto criado na cena do arquivo 3dsmax para indicar onde a câmera deve “nascer”. Outro *trigger* permite o uso de instâncias de uma mesma malha: “__instance|NOME_DA_MATRIZ|__|SEQUENCIAL”. Esse *trigger* permite que o *loader* reutilize uma malha (matriz) no lugar de uma outra que seja semelhante, em tempo de execução.

Para que a navegação simule a movimentação de um veículo, a aplicação-exemplo implementa gravidade e trata colisão usando a biblioteca OPCODE. A integração desta biblioteca de colisão com Ogre3D é denominada OgreOpcode. O *loader* instancia as malhas do cenário na biblioteca OgreOpcode e a câmera colide com o cenário usando uma esfera envolvente (*bounding sphere*). A componente normal da colisão é praticamente anulada, fazendo com que a câmera “escorregue” no sentido tangencial à superfície do objeto colidido.

Conclusões

A solução gerada neste trabalho viabiliza uma transposição bastante eficiente de cenários produzidos no 3dsmax para um ambiente de tempo real com visão de primeira pessoa.

O desempenho computacional do MIRAGEM3D é bastante razoável. Em um cenário com aproximadamente 2500 triângulos, executado em um processador Celeron 2.8Ghz, com placa gráfica 3D integrada de 32MB, o MIRAGEM3D apresenta uma taxa média de 50 fps (*frames per second*) e uma taxa mínima de 20 fps.

Para o futuro, pode-se melhorar o sistema, revendo o uso de Octree, aperfeiçoando o tratamento de colisão e implementando o uso das opções de sombra (*shadow casting*) do Ogre3D. O tratamento de sombras traria um maior realismo para a visualização da cena.

Referências

- 1 - ZERBST, S. and DUVEL, O. **3D Game Engine Programming**, Premier Press, 2004.
- 2 – AUTODESK, **3ds Max**, www.autodesk.com/3dsmax [acessado em 7/07/06].
- 3 – OGRE, **Ogre v.1.2**, <http://www.ogre3d.org> [acessado em 7/07/2006].
- 4 – Terragen, **Photorealistic scenery rendering software**, www.planetside.co.uk/terrigen [acessado em 7/07/2006].