

## Análise de Motores Gráficos 3D para Simulação de Fluidos

Aluno: Bruno Baère Pederassi Lomba de Araujo

Orientador: Bruno Feijó

### Introdução

O estudo anterior foi feito sobre o motor de jogos Torque Game Engine, que apresenta uma solução completa, em se tratando de motor gráfico, motor de física, *scripting* e tratamento de entradas, para desenvolvimento de jogos e simulações. Entretanto, a opção pelo sistema OGRE 3D [1] vem se apresentando como uma alternativa muito forte, especialmente em simulações mais complexas, como a de fluidos em tempo real.

### Objetivos

O presente estudo busca realizar uma análise entre o TGE e o OGRE 3D, levando em consideração que ambos os motores têm grande destaque na comunidade de jogos e que o segundo tem uma crescente aceitação na comunidade científica. O estudo também objetiva comparar estes dois motores na simulação complexa de massas de água.

### Metodologia

A metodologia deste trabalho se resume no estudo detalhado dos dois motores e a realização de comparações na questão de simulação de massas de água, onde as questões de renderização em tempo real são mais complexas.

A escolha do motor gráfico OGRE 3D para estudo comparativo de desempenho gráfico foi realizada com base em sua aceitação na comunidade científica [2][3] e pelo fato de ser uma solução gratuita e de licença aberta (LGPL). O OGRE 3D é desenvolvido em C++ e implementa um grafo de cena, que é uma estrutura de dados que organiza hierarquicamente os elementos de uma cena, como modelos, transformações, iluminação e câmera, para visualização, sendo seu desenvolvimento focado em elegância e extensibilidade [1]. A arquitetura do motor gráfico OGRE 3D é mostrada nas Figuras 1 e 2.

Sobre a arquitetura do OGRE 3D, destacam-se os seguintes elementos: Root, SceneManager, PlatformManager, RenderSystem, Entity, Mesh, Material, SceneNode e Camera. O objeto Root é o ponto de entrada do OGRE 3D, permitindo a configuração do motor e acesso ao sistema. O objeto SceneManager gerencia o conteúdo da cena como câmeras, entidades, luzes e geometria estática (organizados em instâncias de SceneNode), podendo ser reescrito para permitir otimizações na organização dos objetos a serem renderizados conforme as características do projeto. O RenderSystem realiza a interface entre o sistema gráfico utilizado e o motor OGRE. O PlatformManager trata da interface com o sistema operacional, facilitando assim a implantação do motor em diversos sistemas operacionais.

A extensão do OGRE 3D dá-se por meio de *plugins*, que são bibliotecas que substituem ou implementam novas funcionalidades no motor, cujo desenvolvimento é incentivado na comunidade. Como exemplos, têm-se os *wrappers* para motores de física, tais como ODE, Newton Game Dynamics e o Hydrax [4], sendo este último um *plugin* de renderização de massas de água. Com o passar do tempo, foram incluídas bibliotecas de interface gráfica com o usuário (CEGUI) e de gerenciamento de entrada (OIS) na distribuição padrão do OGRE 3D, com a finalidade de acelerar o desenvolvimento da aplicação final, sendo essas bibliotecas elogiadas e aprovadas pela comunidade de desenvolvedores independentes.

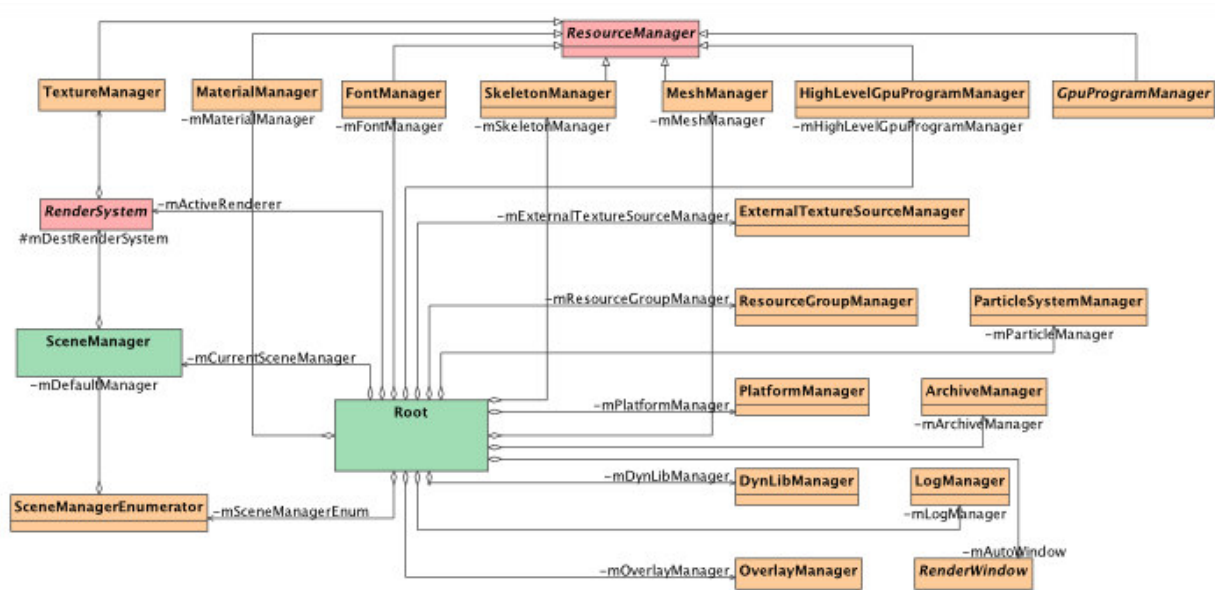


Figura 1 - Modelo UML da Arquitetura do OGRE 3D

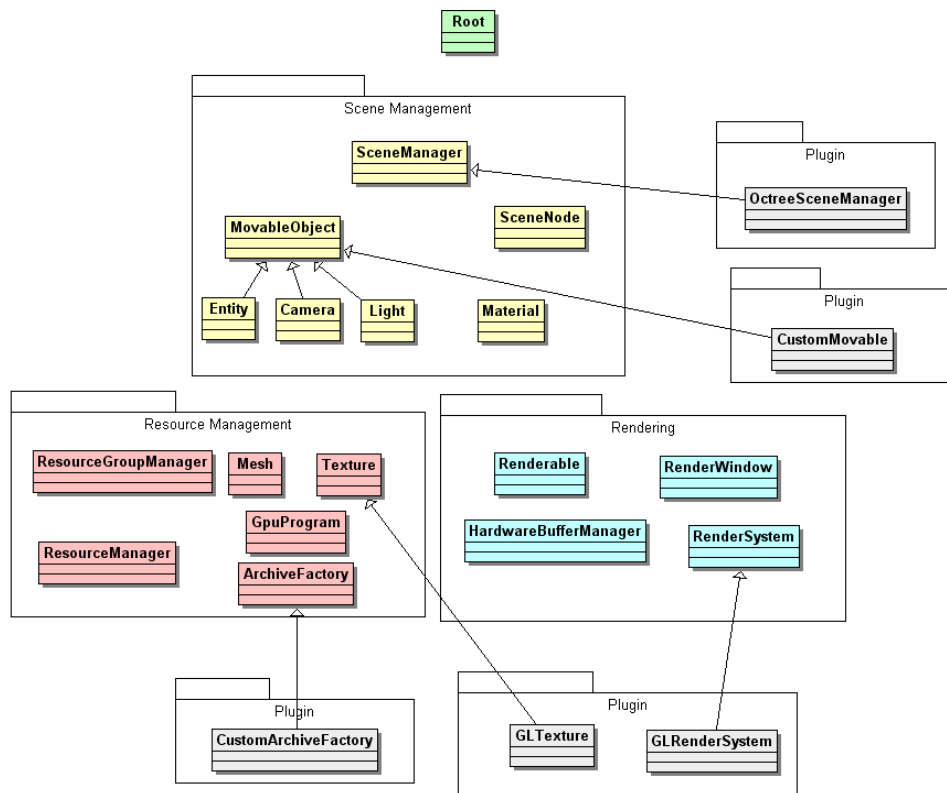


Figura 2 - Arquitetura do motor OGRE 3D, divisão por módulos

Um dos principais produtos desenvolvidos pela comunidade de interesse para este trabalho é o *plugin* Hydrax. Desenvolvido por Xavier Verguín González e atualmente na versão 0.3, o Hydrax utiliza o modelo de Perlin para geração de ruído a fim de se obter uma simulação gráfica mais realista de massas de água. A aparência é complementada com *shaders* escritos nas linguagens HLSL e GLSL e aplicação de *decals* sobre a *mesh* da água para representação de espuma de arrebentação, plantas aquáticas e trilha de espuma de corpos como embarcações. Na versão atual, o Hydrax não oferece suporte para efeitos em submersão, como difração subaquática de luz, que se mostra desejável para aplicações como

simulação de robôs-submarino. Para as futuras versões, o desenvolvedor pretende implementar geração de ondas com FFT, *Fast Fourier Transform*, e um editor auxiliar para os efeitos do Hydrax. A Figura 3 mostra uma cena renderizada com o módulo de Infinite Ocean do *plugin*.



Figura 3 - Infinite Ocean renderizado no OGRE usando plugin Hydrax

O motor Torque Game Engine Advanced é reconhecido por ser um motor de qualidade de baixo custo para desenvolvedores independentes. Diferencia-se da versão simples, Torque Game Engine, principalmente pelo suporte a *shaders* e melhor desempenho gráfico, exigindo uma máquina mais potente. Ambas as versões apresentam uma solução completa em termos de motor de jogos, incluindo componentes de física e *networking*, editores internos de interface gráfica e de mundo e estão disponíveis para as plataformas PC, Mac, Linux, Xbox 360 e Wii. A aquisição das licenças fornece acesso ao código para edição e à documentação da Torque Developer Network. O TGEA também conta com o suporte de uma comunidade ativa nos fóruns e tem atraído a atenção de desenvolvedores pela sua qualidade, apesar da reconhecida curva de aprendizado necessária para dominar sua utilização.

O motor TGEA utilizado neste trabalho é a versão 1.0.2, que implementa para simulação gráfica de corpos de água apenas a reflexão Fresnel. A simulação de ondas é possível com alteração dos parâmetros do *waterblock* (velocidade, direção e escala da textura de água), como mostrado na Figura 4, porém não é tão realista quanto desejado para uma simulação de embarcações. O desenvolvedor do motor pretende disponibilizar a versão 1.7 com melhorias na renderização de água, incluindo simulação de ondas por *normal mapping* e efeitos de luz subaquáticos, usando o motor Atlas Terrain Engine.

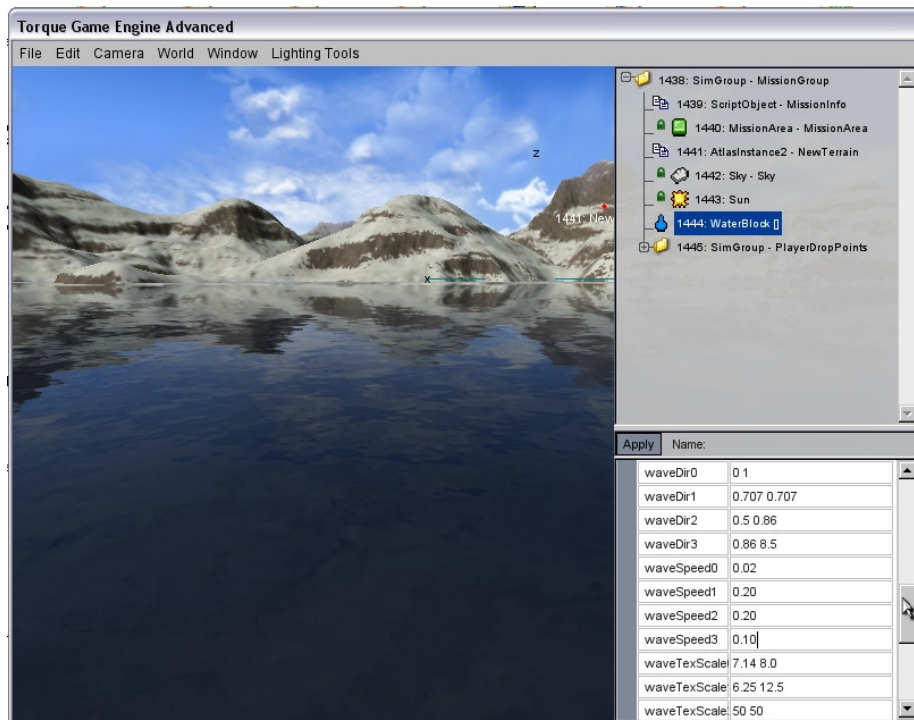


Figura 4 - Parâmetros do waterblock no editor interno do TGEA

## Resultados

Ambos os motores oferecem um bom suporte gráfico através de *shaders*, sendo reconhecida a qualidade dos produtos recentemente desenvolvidos para cada um [5][6]. Para o TGE foram desenvolvidos alguns simuladores, tais como o Take a Walk on Mars e ferramentas de virtualização, como Virtual Canada e 3D Language Spain. Nota-se, porém, que há poucos trabalhos científicos que citem o TGE, dada a preferência da comunidade por software livre.

A Figura 5 apresenta uma comparação entre OGRE e TGE, sendo que no OGRE estão simuladas duas situações (a renderização padrão e aquela obtida pelo *plugin* Hydrax). A máquina utilizada para as simulações apresenta a seguinte configuração: Pentium D 930 3GHz, 2048MB RAM, GeForce 7900GT 256MB e foi utilizado o *software* FRAPS para captura de imagens e FPS. Com os seguintes parâmetros de simulação no OGRE e OGRE com *plugin* Hydrax: frequência da onda e velocidade da onda nos eixos x e y. Respectivamente foram obtidos 59 FPS e 56 FPS. No TGE, apesar de atingir a marca de 157 FPS, não foi possível obter uma simulação realística da onda com o que o motor oferece da mesma qualidade que o OGRE.

A qualidade do OGRE 3D para simulação de água em tempo real apresenta-se muito superior à permitida pelo TGE, levando em consideração a importância do realismo para fins de simulação. A utilização de um motor de física, como o Newton Game Dynamics, aliado ao *plugin* de renderização Hydrax, possibilita a criação de um simulador de navios com grande qualidade de realismo e baixo custo de desenvolvimento, visto utilizar-se de ferramentas de software livre.

Uma outra característica do OGRE 3D que se sobressai em relação ao TGE é sua portabilidade, que permite que seja facilmente transportada para vários sistemas operacionais [2], tais como Windows, MacOS e Linux, e para várias linguagens, tais como Python (através do projeto pyOgre). Além disto, com o OGRE 3D, tem-se uma maior facilidade em se desenvolverem *plugins*, o que se faz através da classe RefAppLayer.



Figura 5 - Comparação do rendering do OGRE, OGRE com plugin Hydrax e TGEA

## Conclusões

Por apresentar uma solução mais geral, envolvendo editor de cenário integrado, suporte a modelagem de personagens e elementos como ferramenta própria e um poderoso sistema de *scripting*, o Torque Game Engine é uma ótima opção para a rápida prototipação de um jogo ou simulador, como já é utilizado por grandes instituições [8], além de oferecer um produto final de qualidade.

O OGRE 3D apresenta uma melhor solução gráfica, principalmente porque se trata de um projeto nascido para ser um componente essencialmente gráfico para simulações. Ademais, no OGRE 3D, tem-se que a integração com outras bibliotecas é mais fácil (dado o suporte nativo pela RefAppLayer), o interesse da comunidade é maior e o código é livre e aberto. Tanto o Torque Game Engine Advanced quanto o OGRE 3D possuem suporte a *shaders*.

Ambos os motores se sobressaem por serem soluções de baixo custo. Entretanto, cada um destes motores tem seu próprio escopo de aplicação, sendo essa a principal consideração a ser tomada para a decisão de qual motor usar em um determinado projeto.

## Referências

- [1] JUNKER, G. 2006. **Pro OGRE 3D Programming**, 1. ed. Estados Unidos: Apress.
- [2], LIMA, J. P. S. do M., DE FARIAS, T. S. M. C., TEICHRIEB, V., KELNER, J. **Port of the OGRE 3D Engine to the Pocket PC Platform**. Proceedings of the Symposium on Virtual Reality, Belém, pp. 65-76, 02-05 de Maio, 2006.
- [3] PESSOA, S. A., DE FARIAS, T. S. M. C., TEICHRIEB, V., KELNER, J. **O Engine Gráfico OGRE**. Tutorial in the V Brazilian Symposium on Computer Games and Digital Entertainment (SBGames 06), Recife, 08-10 de Novembro, 2006.
- [4] GONZÁLEZ, X. V. **Hydrax v0.3**.  
<http://www.ogre3d.org/phpBB2/viewtopic.php?t=38148>. [acessado em: 5 de Maio de 2008]
- [6] **Dreamlords**. <http://www.dreamlords.com/home.action> [acessado em: 3 de Junho de 2008]
- [7] **Pacific Storm**. <http://www.pacificstorm.net/> [acessado em: 3 de Junho de 2008]
- [8] **Torque Game Engine**. <http://www.garagegames.com/products/torque/tge/> [acessado em: 5 de Maio de 2008]