5
Resultados

5.1.
Integração do Sistema de Macros com o Motor de Jogos Fly3D

O motor de jogos Fly3D é um conjunto de aplicações, bibliotecas e ferramentas construídas para proporcionar um ambiente robusto e intuitivo para o desenvolvimento de jogos eletrônicos. Atualmente em sua versão 3.0, o Fly3D é desenvolvido pela Paralelo Computação Ltda. (Paralelo, 2003).

As bibliotecas que integram o motor Fly3D têm como objetivo fornecer o acesso a implementações otimizadas de classes e métodos comuns aos jogos eletrônicos. O desenvolvedor de jogos tem à sua disposição módulos que cuidam automaticamente de várias tarefas necessárias à execução de jogos, como renderização na tela, controle da simulação, processamento de entrada de dispositivos como teclado, *mouse* e *joystick*, geração de informações de depuração, entre outras.

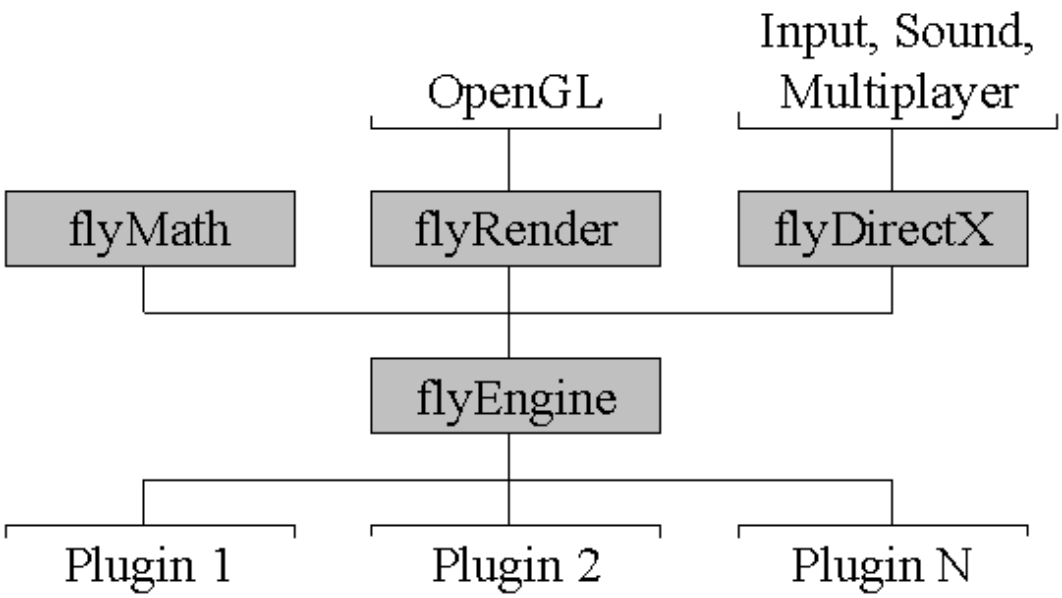

Figura 5.1 Arquitetura de bibliotecas do motor Fly3D.

A arquitetura das bibliotecas que compõem o Fly3D está mostrada na **Figura 5.1**. A seguir, são apresentadas descrições breves de cada um dos módulos componentes:

- *flyMath*: biblioteca matemática, cuida das tarefas de cálculo necessárias aos jogos, como algoritmos geométricos, bem como dos métodos de manipulação de números reais em ponto flutuante;
- *flyRender*: biblioteca de renderização, encapsula o uso da API *OpenGL* e cuida do desenho na tela a cada quadro da simulação, compreendendo também todas as variáveis que regulam a exibição no dispositivo de saída, como tamanho e aspecto da janela, quantidade de cores, etc.;
- *flyDirectX*: biblioteca de suporte à captura de dispositivos de entrada, reprodução de músicas e efeitos sonoros e conectividade para jogos multi-jogador, encapsula o uso da API *DirectX* da Microsoft para cuidar dessas tarefas, através dos módulos *DirectInput*, *DirectSound* e *DirectPlay*, respectivamente, fornecidos pela API;
- *flyEngine*: biblioteca principal, cuida da simulação do jogo, carregando a cena desejada do disco para a memória, montando as estruturas de dados que representam os objetos, acionando o desenho de cada quadro através da biblioteca *flyRender*, coordenando a execução do código de atualização do estado do jogo de cada *plug-in* existente na simulação, processando e validando a cena de jogo atual.

O motor Fly3D serve como componente básico ao desenvolvimento de jogos. O código específico do jogo é implementado através de *plug-ins*, que utilizam as classes e métodos fornecidos pelo motor através de ligação dinâmica em tempo de execução.

Analogamente, a implementação de um sistema de suporte ao desenvolvimento de agentes inteligentes para jogos pode ser acoplada ao motor Fly3D. Uma nova biblioteca, *FlyAI*, está sendo desenvolvida, e será adicionada ao conjunto de bibliotecas do Fly3D. A biblioteca *FlyAI* é responsável por acionar o código que atualiza o estado dos agentes a cada quadro. A implementação dos agentes em si está contida em cada *plug-in*, pois depende do jogo a ser desenvolvido.

O Sistema Baseado em Macros apresentado no **Capítulo 4** é integrado ao motor Fly3D da seguinte maneira: as macros que implementam a lógica do sistema são definidas em um arquivo pertencente à biblioteca *flyAI*; essa biblioteca também é responsável pelo controle do fluxo de execução de cada *plug-in* durante a atualização do estado antes do desenho de um quadro da simulação; as Máquinas de Estados que compõem os agentes são definidas nos *plug-ins*, pois dependem do jogo que será desenvolvido; é a biblioteca *flyAI* quem aciona o método *process* de cada agente definido nos *plug-ins*.

5.2. Experimentos para Medição de Desempenho

Jogos eletrônicos são aplicações de computação gráfica interativa em tempo real, e como tal necessitam de um número mínimo de quadros renderizados por segundo para que a animação e a interatividade sejam considerados satisfatórios. Akenine-Möller e Haines (2002) argumentam que uma taxa de 15 fps (do inglês *frames per second*, ou *quadros por segundo*) é o valor mínimo para que a animação em uma aplicação desse tipo seja considerada aceitável.

Dessa forma, uma das maneiras de se mensurar o impacto da utilização de técnicas sofisticadas em aplicações como jogos eletrônicos é medir a taxa média de quadros por segundo obtida durante uma partida de jogo com um tempo de duração pré-determinado. Uma abordagem semelhante é utilizada na aquisição de resultados práticos em vários trabalhos que envolvem computação gráfica em tempo real, merecendo destaque a obra mais recente de Fonseca (2004).

Os experimentos descritos a seguir foram realizados em um PC Intel Pentium IV 2.4GHz, com 512MB de memória RAM e placa de vídeo NVidia QuadroFX 2000.

Primeiramente, foi medida a média de quadros por segundo de uma partida de 2 minutos na cena *ship_sp1*, que integra o pacote de demonstrações do Fly3D (Fly3D, 2003). Nessa primeira medição, o código referente à implementação de comportamento dos agentes inteligentes através do Sistema de Macros não foi incluído, para que fosse possível observar o desempenho original da aplicação.

Em seguida, foram realizados experimentos que incluíram o código desenvolvido nesta obra. Em cada um deles, um tipo diferente de Máquina de

Estados foi utilizado no controle de comportamento das naves-robôs inimigas, que batalham para destruir a nave do jogador. A **Tabela 6.1** mostra os resultados das medições de média de quadros por segundo de cada experimento, bem como o desvio padrão com relação à média em cada caso. As Máquinas de Estados utilizadas foram: a Máquina de Estados Finitos da **Figura 3.1** (FSM); uma Máquina de Estados *Fuzzy* correspondendo à versão *fuzzy* da **Figura 3.1** (FuSM); a Máquina de Estados Finitos Hierárquica da **Figura 3.2** (HFSM); e uma Máquina de Estados *Fuzzy* Hierárquica correspondendo à versão *fuzzy* da **Figura 3.2** (HFuSM).

Experimento	**Valor médio de Quadros por Segundo**	**Desvio Padrão**
Original	99 fps	2 fps
FSM	81 fps	3 fps
FuSM	44 fps	9 fps
HFSM	84 fps	5 fps
HFuSM	45 fps	9 fps

Tabela 6.1 Resultados dos experimentos na cena *ship_sp1* do Fly3D.

Os resultados da medição do desempenho original da aplicação mostram uma taxa relativamente alta de quadros por segundo; a utilização de Máquinas de Estados Finitos no controle de comportamento dos agentes diminuiu essa taxa em cerca de 20%, mas o valor ainda se manteve bem acima do mínimo aceitável. No caso das Máquinas de Estados *Fuzzy*, a queda de desempenho é bem maior, como se esperava, devido à multiplicidade de estados a serem considerados a cada momento. A utilização de Máquinas de Estados Hierárquicas promove uma pequena melhora no desempenho quando comparada com os casos não-hierárquicos correspondentes. O alto desvio padrão dos valores referentes às Máquinas de Estados *Fuzzy* demonstra que seu desempenho pode variar bastante durante a simulação, em virtude da oscilação no número de estados pertencentes ao conjunto estado corrente.

Além disso, o uso de Máquinas de Estados Hierárquicas proporciona uma diminuição na quantidade de memória necessária para representar as Máquinas de Estados. Isso acontece porque o número de transições representadas é bem menor

do que em uma Máquina de Estados não-hierárquica que implemente o mesmo comportamento. Nos experimentos apresentados, a diferença de quantidade de memória é pequena, porém tende a crescer conforme a complexidade das Máquinas de Estados utilizadas aumenta.

As figuras a seguir mostram imagens das partidas realizadas na cena *ship_sp1*. As naves-robôs enfrentam a nave do jogador tentando destruí-la, mediante o comportamento representado e controlado pelas Máquinas de Estados citadas acima.

Figura 6.1 A nave-robô azul busca uma melhor posição para atacar o jogador pelo lado.

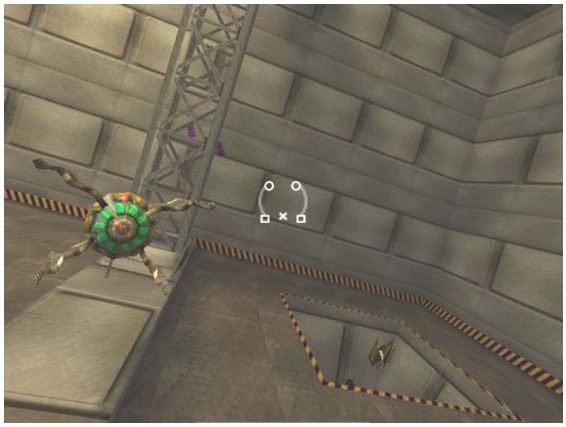

Figura 6.2 Nave-robô verde pronta para atacar.

Figura 6.3 Nave-robô lança um míssil tele-guiado contra o jogador.