

# Ambiente Virtual para Treino e Avaliação de Pilotagem

Manuel Próspero dos Santos

Hugo Filipe Abreu

Departamento de Informática  
Faculdade de Ciências e Tecnologia da  
Universidade Nova de Lisboa

Quinta da Torre, 2825-114 Caparica  
{ps@di,hfa@students.si}.fct.unl.pt

## Sumário

*Apresenta-se um ambiente de simulação de voo destinado ao treino e avaliação de pilotos de aviões de combate numa plataforma de baixo custo mas, em contrapartida, exibindo um grande rigor técnico e realismo. A arquitectura foi concebida de forma modular, o que permite uma adaptação a outro género de aviões, quer comerciais, quer particulares. Pela mesma razão se podem também adaptar diferentes dispositivos de interface. O ambiente virtual é de multi-utilizador, em rede de computadores pessoais. Uma vez que os aspectos de interacção e de síntese de imagem são muito importantes nos produtos finais para simulação de voo, sendo isso até um dos critérios mais comuns para se julgar a sua qualidade, o artigo refere predominantemente os assuntos relacionados com tal vertente, em prejuízo dos temas que têm a ver com a modelação aerodinâmica e as regras de voo.*

## Palavras-chave

*Ambientes virtuais, simulação de voo, sistemas distribuídos, OOP, OpenGL.*

## 1. INTRODUÇÃO

Os simuladores de voo são ferramentas muito úteis para testes de equipamento e aprendizagem de pilotagem em situações inviáveis com aviões reais. O desenvolvimento de um simulador proprietário permite a expansão, experimentação e acréscimo de características e funcionalidades, capacidades essas que normalmente não se encontram num sistema comercial. Foi este desafio que o Instituto Nacional de Engenharia e Tecnologia Industrial (INETI) lançou ao Departamento de Informática da UNL, no contexto de um projecto de simulação em defesa que mantinha com o Ministério da Defesa, e no âmbito do convénio INETI-UNL existente.

O denominado projecto ACTES (*Air Combat Training and Evaluation System*) pretendia desenvolver um sistema de baixo custo, a ser implementado em torno de aviões de combate F-16A e tendo como principal intuito a melhoria da prática dos pilotos que voam tais aeronaves, quer a nível nacional, quer internacional. Por outro lado, como objectivo secundário estava também a intenção de se constituir um laboratório de experimentação nesta área. O que se obteve ao fim de quase dois anos de trabalho foi um sistema a funcionar em rede com vários postos de interacção, estruturado por uma arquitectura genérica, capaz da maior parte das simulações de voo em combate aéreo e com um elevado grau de realismo para uma plataforma relativamente modesta. Ainda não é possível encontrar, nos circuitos comerciais, qualquer sistema equivalente ao ACTES. E apenas por não se enquadrarem nos objectivos do

projecto, ficaram por desenvolver as operações de descolagem e aterragem dos aviões.

Qualquer ambiente virtual que actualmente se possa designar como simulador de voo, seja de tipo militar, comercial ou outro, tem necessariamente uma forte componente gráfica, sendo mesmo uma das primeiras referências de comparação dos sistemas de alto nível. É neste contexto que o artigo descreve o modelo conceptual e aborda aspectos importantes da implementação do sistema de simulação de voo que, além de permitir o treino de pilotagem com mais do que um avião em presença, numa rede de computadores, teve também como objectivo a avaliação do comportamento dos pilotos pelo acompanhamento das missões numa estação de supervisão.

## 2. COMPONENTES DO AMBIENTE DE SIMULAÇÃO

Na sua essência, um simulador de voo deve incorporar uma representação do mundo onde a simulação vai decorrer, com os protagonistas das sessões de simulação e as regras da simulação, onde se insere a física aerodinâmica e o funcionamento das aeronaves. Para além disto, naturalmente, deve incluir uma interface com o utilizador final.

Como está ilustrado na Figura 1, o modelo do MUNDO será constituído pelos protagonistas, as leis que os regem e o cenário envolvente. Cada protagonista, identificado na mesma figura por OBJECTO, tem uma relação de dependência forte com as leis da física usadas na

simulação.

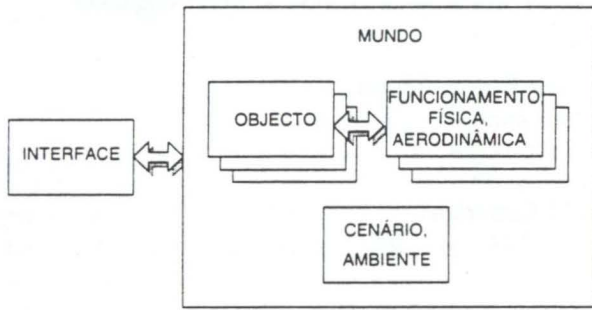


Figura 1: Principais componentes de um simulador de voo.

Por exemplo, a AERODINÂMICA define o envelope de manobra associado ao avião, ou seja, os movimentos que podem ser pedidos a este, dadas todas as circunstâncias particulares de voo. Tanto o CENÁRIO como o AMBIENTE gerado são, em geral, componentes mais estáticas que completam a simulação, muito embora possam existir algumas dependências excepcionais, como é o caso da atmosfera ao interagir com a dinâmica dos objectos. Por outro lado, a INTERFACE é uma construção que só faz parte do próprio simulador, visto não existir no mundo real uma janela e um sistema gráfico com os dispositivos através dos quais o piloto vai interagir com o sistema.

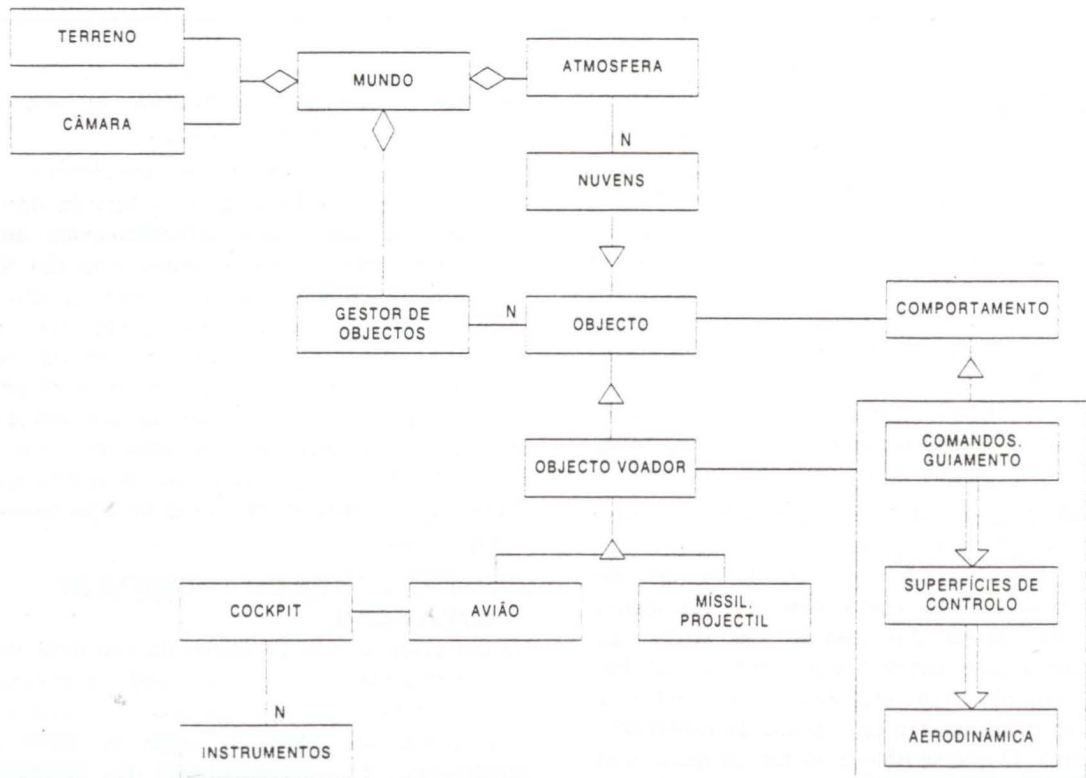


Figura 2: Arquitectura do ambiente virtual implementado.

A especificação de combate que foi tida em conta no projecto leva a que tenhamos no cockpit instrumentos dedicados ao armamento (mísseis e balas de canhão), que se enquadram perfeitamente dentro da arquitectura genérica definida e concretizam o sistema militar de

Para melhor entendimento deste modelo de ambiente virtual, comece-se por referir que os equipamentos de comando (como o manche, interruptores e teclas, entre outros) deverão ter também a sua representação no MUNDO e associar-se-ão ao OBJECTO respectivo (o F-16A, por exemplo, para o caso do manche). Porém, o equipamento de controlo que está ligado ao computador não faz parte do MUNDO da simulação: no mundo real, o piloto interage directamente com cada OBJECTO do sistema e não com uma interface para eles. Por esta razão, a INTERFACE não pertence ao MUNDO do nosso modelo, mantendo no entanto uma ligação muito forte com os componentes deste último.

### 3. ARQUITECTURA DO SISTEMA ACTES

A organização do sistema que engloba a colocação e interligação das estruturas, as hierarquias e as relações entre os módulos define a respectiva arquitectura. A que se definiu para o ACTES é bastante flexível, adequa-se a um funcionamento multi-utilizador em rede de computadores e abrange, em funcionalidade, a maioria dos diferentes tipos de simuladores de voo. Assim, esta metodologia não é alterada pelo facto do actual simulador ser dedicado ao treino de combate, mantendo-se a mesma arquitectura se fosse aplicada a simuladores de voo para aviões comerciais ou particulares.



o seu conjunto específico de instrumentos, os quais definem uma parte importante da interface com o utilizador quando esse avião é pilotado.

O ACTES foi projectado segundo o paradigma da programação orientada pelos objectos, de forma a que cada membro da estrutura pudesse ser substituído ou melhorado individualmente, derivando novas classes [Sommerville95]. Por outro lado, a estrutura de classes permite uma associação imediata das entidades do simulador com as do mundo real.

A arquitectura definida dispõe de espaço para cada OBJECTO que povoará o MUNDO e obriga a que haja heranças para obter novas classes de OBJECTO a partir dos tipos básicos. Assinale-se que o tipo mais básico de OBJECTO não voa e, inclusivamente, é inerte, mas tem associado um modelo gráfico tridimensional. Deste tipo de OBJECTO são derivadas novas classes (como as classes dos objectos em movimento, dos objectos motorizados, dos objectos voadores motorizados ou dos aviões). Os futuros criadores de um sistema particular com esta arquitectura definirão uma estrutura de classes com o detalhe que acharem necessário. No estado actual do ACTES, as classes de objectos mais completas e bem definidas são aquelas que especificam o F-16A, o míssil Sidewinder, as nuvens e as explosões.

Os aviões são pilotados manualmente através de algum tipo de equipamento de comando – como o *manche* ou os pedais – que directa ou indirectamente actuam sobre as superfícies de controlo do avião (como os *aileron*s ou o leme) para levá-lo a fazer as manobras que o piloto idealiza. Aos objectos da classe AVIÃO, com maior ou menor especificidade, deve estar associada uma entidade de COMANDOS que define a actuação que se pretende que o objecto voador tenha. Os COMANDOS de um avião são essencialmente compostos pelo *manche*, pelo controlo de potência, pelos pedais e por uma quantidade de interruptores e potenciómetros auxiliares. Os COMANDOS de um míssil, por outro lado, são o seu sistema electrónico de guiamento e perseguição do alvo.

No caso dos aviões F-16A existe um sub-sistema de modulação dos ângulos de deflexão das superfícies de controlo que são, por esta via, indirectamente controladas pelos COMANDOS de forma a permitir uma pilotagem mais confortável. Este sub-sistema foi inserido, muito adequadamente, no sistema de COMANDOS do F-16A, transformando o estado do *manche* em ângulos das superfícies de controlo.

É de notar que os COMANDOS não são necessariamente um conjunto de peças de hardware como o *manche* e os pedais. Com efeito, existe a possibilidade de, numa qualquer sessão do ACTES, introduzir aviões que possuem um conjunto de COMANDOS em tudo semelhantes aos anteriormente descritos mas virtuais, accionados por um piloto igualmente virtual, a que se deu alguma inteligência artificial e objectivos bem definidos. Este piloto manipula o *manche* virtual que, por sua vez, irá actuar de forma normal sobre as superfícies de controlo e, a partir daí, a AERODINÂMICA tomará conta do comportamento desse avião como se se tratasse

de outro avião pilotado por um ser humano. Esta característica demonstra a modularidade da estrutura do simulador onde, por substituição do módulo de COMANDOS, se pode obter uma funcionalidade do avião completamente diferente.

Também é necessária uma entidade de tratamento da AERODINÂMICA. No caso do modelo típico de um avião, a AERODINÂMICA determina o movimento e reorientação do aparelho no ar em função do estado das suas superfícies de controlo, da potência do motor e dos momentos de inércia correntes [Kermode60, Kermode62]. Aqui deve estar definido o modelo matemático da aeronave e esta é uma das bases para um sistema de simulação de voo. Na Natureza, um só conjunto de regras da Física se aplica a todos os aviões. Num simulador, pelo contrário, é necessário simplificar as regras. Desta forma, pode dizer-se que não existe um simulador de aerodinâmica universal para todos os tipos de avião. Mas, dada a especificidade do assunto, que extravasa o âmbito duma conferência sobre Computação Gráfica, neste artigo não se abordará a problemática da aerodinâmica [Murray, Stevens92].

### 3.1 Comportamento dos Objectos

A estrutura geral do sistema exige que cada objecto tenha três funções básicas que o distingue, atribuindo-lhe características e um estado. Estas características, esquematizadas na Figura 3, podem ser herdadas ou redefinidas:

- O objecto encarrega-se do seu aspecto gráfico, determinando se deve usar os métodos de desenho genéricos das classes hierarquicamente superiores ou definir o seu próprio método de desenho.
- A cada classe de objectos é também atribuída a responsabilidade do controlo do comportamento, seja o objecto um avião (com a aerodinâmica complexa própria de um avião), um míssil (com a aerodinâmica algo simplificada) ou uma árvore (sem qualquer movimento). Um objecto protagonista pode ser controlado pelo computador local ou remotamente, neste último caso por outro computador na rede da simulação.
- Cada classe de objectos inclui código para empacotar e desempacotar o estado das suas instâncias para serialização e transmissão em rede.

OBJECTO
ATRIBUTOS DE ESTADO
DESENHAR_3D AVANÇAR( ΔTempo ) EMPACOTAR DESEMPACOTAR

Figura 3: Definição essencial de um objecto genérico.

Como os objectos são encarregados de fazer o seu próprio *rendering* a nível individual, o sistema utiliza o



algoritmo de Z-buffer para que funcione de maneira correcta, independentemente da ordem de chamada dos métodos de desenho dos diversos objectos.

Recorde-se que o MUNDO engloba todos os objectos. Assim, quando se faz AVANÇAR o MUNDO ligeiramente, este encarrega-se de fazer AVANÇAR todos os seus objectos, informando-os do acréscimo que o tempo sofreu. Trata-se de uma alteração incremental, com o valor do tempo que passou desde a projecção do último fotograma [Muchaxo95].

Na verdade, com as funções de dinâmica, distribuição e síntese de imagem, cabe aos objectos protagonistas a maior parte do processamento envolvido no sistema. Devido ao uso intensivo dos objectos, é necessário que faça parte da arquitectura um sub-sistema de gestão capaz de controlar, no computador local ou remotamente, a criação, existência e remoção destes objectos protagonistas, fornecer informações de e para os mesmos, detectar as colisões entre objectos (locais e remotos) e actuar segundo os resultados destas. O designado GESTOR DE OBJECTOS permite criar e destruir objectos, tanto na fase de inicialização, como durante a sessão de simulação.

### 3.2 Funcionamento em Rede

Como já foi referido e o próprio nome indica, o ACTES é um sistema de treino de combate. Por essa razão, é quase fundamental que seja planeado e implementado em rede, para que vários utilizadores interajam simultaneamente em combate num cenário comum. É importante que a arquitectura esteja capacitada para tal, concretamente pela mão do GESTOR DE OBJECTOS e pelos próprios objectos. O GESTOR DE OBJECTOS pode conhecer o estado de todos os objectos da sessão, sejam eles locais ou remotos, mantendo um registo da existência de cada um. Encontra-se um GESTOR DE OBJECTOS em cada estação de simulação na rede.

Como se viu, é o GESTOR DE OBJECTOS que controla a distribuição dos objectos pela rede onde decorre a simulação. No entanto, os objectos individuais são responsáveis pelo empacotamento e desempacotamento (através de *stubs* de rede embebidos nos próprios objectos) das suas variáveis de estado que são transmitidas pelo GESTOR DE OBJECTOS via rede. A cada fotograma, o GESTOR DE OBJECTOS sincroniza os conteúdos da rede, podendo descartar as mensagens desactualizadas ou manter um pequeno historial de posições e orientações. Desta forma, é possível fazer a extrapolação desses valores quando se torne necessário projectar um fotograma e as mensagens correspondentes aos objectos das estações remotas não chegaram a tempo. No ACTES usa-se um historial de 10 posições, que são extrapoladas segundo uma função exponencialmente decrescente para achar a próxima posição provável do objecto remoto, dando maior peso às posições mais recentes.

O GESTOR DE OBJECTOS deve ainda divulgar na rede as colisões de objectos que foram detectadas localmente e ignorar divulgações de outros GESTORES DE

OBJECTOS da mesma rede acerca de colisões que já tenham sido detectadas. Recorre-se, para isso, a um historial das últimas colisões.

### 3.3 Supervisão

O método de divulgação na rede (*multicasting*) permite a criação de uma estação dedicada à supervisão do decurso das sessões de voo. Nesse tipo de estação também existe a possibilidade de criação ou destruição de objectos, de alteração do ambiente da missão ou de efeitos no cenário. Apesar da estação de supervisão estar inserida na rede de simulação, ela não vai controlar a AERODINÂMICA de nenhum OBJECTO. Apenas monitoriza os objectos existentes nos postos de simulação e cujo estado é divulgado através da rede.

O extremo oposto da incumbência da estação de supervisão – que não controla nenhum OBJECTO – é o de um servidor de simulação, em que a máquina inserida na rede não precisa de ter, nem sistema gráfico, nem *hardware* de controlo de aviões, mas pode controlar objectos automáticos como sejam os aviões com inteligência artificial. Isto permite reduzir o tempo de processamento dos postos de simulação nas sessões em que são usados muitos objectos, podendo mesmo vir a passar-se o controlo dos mísseis e projecteis – criados em tempo de simulação – para servidores desse tipo, conforme assim o indicasse o sistema de medição de carga dos processadores.

### 4. REALISMO VISUAL

Como se referiu anteriormente, o peso das imagens num ambiente para simulação de voo é extraordinariamente significativo. Com efeito, e na maioria das vezes, é em resposta à variação das imagens que o utilizador final irá reagir. Torna-se, pois, deveras importante tentar-se obter o maior grau de realismo possível sem afectar significativamente o desempenho da configuração usada. Na Figura 4 pode ver-se um fotograma correspondente a uma missão de voo em F-16A no ACTES, notando-se o cuidado com que os instrumentos de navegação e controlo foram desenhados. Com efeito, e tal como acontece na realidade, a navegação pode ser feita através desses mesmos instrumentos. De entre eles destaca-se, para um avião de combate do género, o HUD (*Head Up Display*) ao centro do ecrã, por cima do radar.

A existência da ATMOSFERA associada ao MUNDO é um ponto importante e praticamente exclusivo dos sistemas de simulação de voo. A atmosfera define características determinantes no comportamento em voo. A pressão e a densidade do ar condicionam a resistência ao movimento, o impulso dos motores, a velocidade do som e as funções de conversão de medidas de velocidade.

As nuvens fazem parte da atmosfera e são objectos geridos pelo MUNDO, tal como os aviões. Existe actualmente muita dificuldade na modelação e representação gráfica das nuvens por estas não terem limites bem definidos. Os processos de representação de nuvens visualmente muito realistas tomam demasiado tempo de processamento, que é um recurso deveras limitado em sistemas de baixo custo. Por essa razão, no



ACTES usam-se camadas de quadriláteros que, com base em texturas e transparências, são um compromisso

bastante razoável para representar as nuvens (Figura 7. na imagem superior).

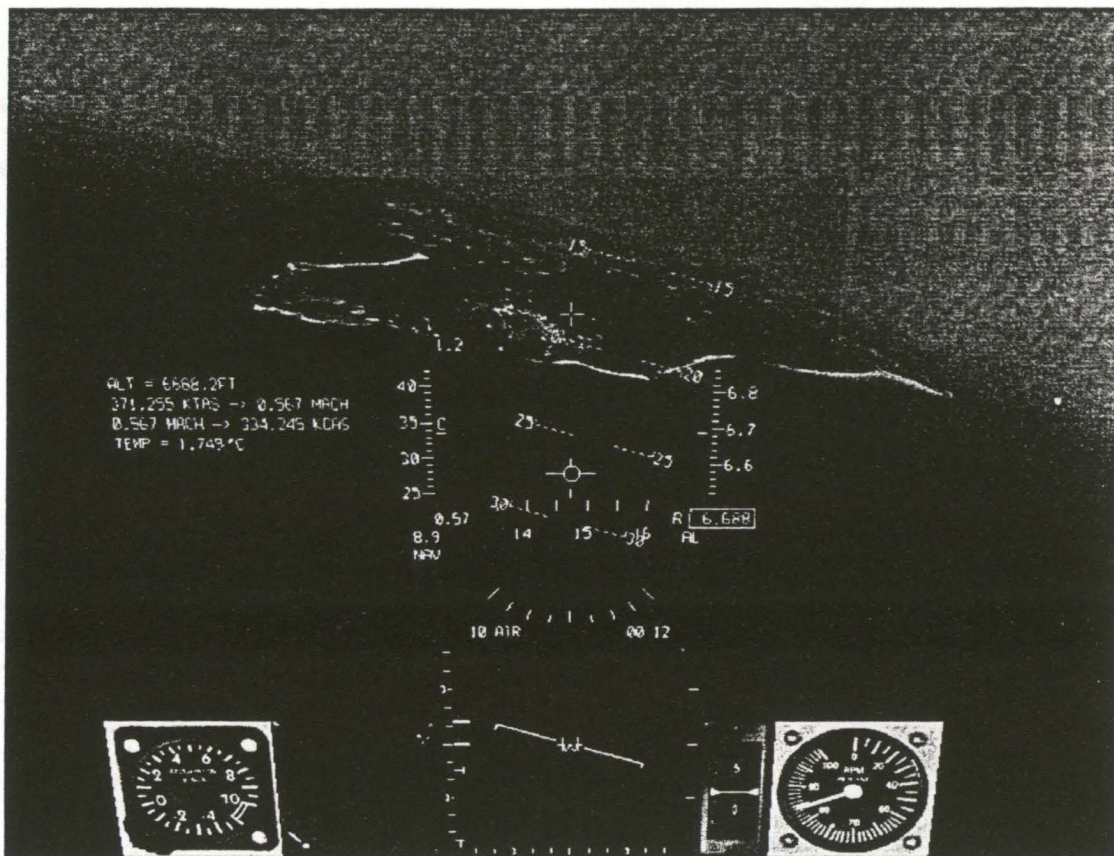


Figura 4: Aproximação a uma ilha durante a simulação de voo em F-16A.

Também é importante referir que as nuvens são especialmente relevantes neste tipo de ambiente virtual uma vez que, durante a sessão de simulação, elas têm a função estratégica de ocultar outros aviões quando são muito densas. Possuem, ainda, a capacidade de alterar o comportamento do avião que as trespassa, acrescentando alguma turbulência aerodinâmica. Este não é, porém, o caso de tratamento mais prioritário de colisão entre objectos. Na verdade, é óbvio que o teste de colisão do avião com outras aeronaves ou com o solo é muito mais importante.

A detecção de colisões envolve alguma estratégia para minimizar o custo de processamento e manter a coerência num sistema distribuído. Ela está integrada no GESTOR DE OBJECTOS e, para os objectos voadores, foi implementada de uma forma simples usando-se a intersecção de esferas de influência, como a seguir se descreve. Considera-se que a esfera de influência de um OBJECTO é o volume centrado no objecto com um raio pré-determinado para cada classe, usado como simplificação das formas do OBJECTO para detectar colisões. As posições ocupadas por esses volumes entre dois fotogramas consecutivos são obtidas por interpolação. Ao fotograma corrente atribui-se o tempo  $t=1$  e ao fotograma imediatamente anterior  $t=0$ . Se, como resultado da interpolação entre as duas posições, os dois volumes obtidos por extrusão se intersectarem para um mesmo valor de  $t$ , consideramos ter ocorrido uma colisão

(conforme se pode visualizar na Figura 5).

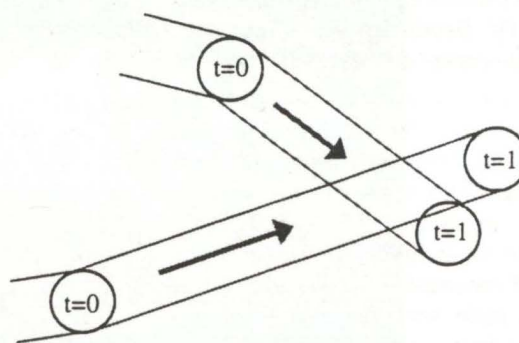


Figura 5: Esquema ilustrativo da detecção duma colisão (no instante  $t=0.75$ ).

Cada classe de objectos de simulação tem um determinado nível de integridade, um limite máximo de danos suportados e um valor de capacidade de danificar outros objectos aquando de uma colisão. Quando esta é detectada, são aplicados os danos consequentes a cada um dos objectos. A partir desse momento, esses objectos irão comportar-se de modo diferente. Um exemplo possível de tais comportamentos é a perda de controlo parcial do avião pilotado após este ter sido atingido por balas de canhão, ao mesmo tempo que vai deixando para trás um rasto de fumo. Se o nível de integridade física



ultrapassar o limite próprio do objecto, este é destruído e eliminado do GESTOR DE OBJECTOS.

Obviamente que se poderá criticar este sistema de detecção de colisões como sendo pouco preciso. Mas se tivermos em conta que os mísseis actuais são detonados alguns metros antes da colisão, devido à existência de um detector de influência, e que, das poucas vezes em que ocorrem colisões entre aviões, as consequências são desastrosas, constatamos que, afinal, o sistema implementado é suficiente para atingir os objectivos pretendidos sem degradação assinalável do desempenho.

Mesmo assim, é ainda importante reduzir o número de testes de colisão que são feitos a cada fotograma. Podem usar-se várias técnicas para eliminar a maioria desses testes, como sejam as que se baseiam na divisão do espaço por zonas e no tipo de objecto. Como exemplo concreto na actual implementação, as colisões com as nuvens não são detectadas, pois caso contrário, e devido ao seu grande número, isso iria reflectir-se numa notável degradação da actuação do sistema a troco do realismo de uma perturbação mínima na aerodinâmica.

Já a detecção da colisão com o terreno é um caso à parte, devendo o TERRENO ter um método de cálculo da altitude como função de todos os pontos definidos por longitude e latitude. Permite-se, assim, que o GESTOR DE OBJECTOS faça a comparação das altitudes com os pontos de referência do avião para saber se houve ou não colisão com o solo.

Ainda que as capacidades do equipamento gráfico usado no ACTES fossem de topo de gama, com elevado desempenho, seria sempre de extrema importância poupar e gerir bem os recursos. Assim sendo, tanto o módulo TERRENO como os próprios objectos deveriam estar preparados com uma modalidade de nível de detalhe (*Level Of Detail* ou LOD) que permitisse otimizar automaticamente a utilização do sistema gráfico.

Dado que os objectos podem desenhar a própria representação e têm conhecimento da sua posição em relação à câmara, também podem implementar individualmente o seu sistema de níveis de detalhe. No entanto, na classe base que define os objectos mais genéricos foi incluído um sistema de transição automática de LOD consoante a distância à posição da câmara. Tal sistema pode ser usado nas classes derivadas. Como exemplo disto, os aviões do ACTES usam o sistema de transição automática de níveis de detalhe correspondentes a modelos 3-D com número de polígonos de diversas grandezas, de acordo com as seguintes condições [CGA, F16]:

- A distâncias inferiores a 1000 pés é visualizada uma representação com 2890 polígonos.
- Entre 1000 e 15000 pés a representação possui apenas 296 triângulos.
- A mais de 15000 pés o avião não é mostrado.

Quanto aos mísseis, dado que só são avistados durante alguns décimos de segundo após o disparo, o único nível de detalhe do respectivo modelo é usado até à distância

de 2000 pés e possui apenas 93 polígonos (Figura 6).

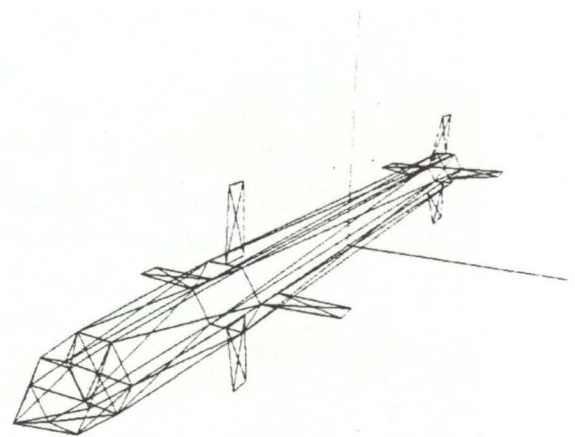


Figura 6: Modelo do míssil Sidewinder AIM-9P.

As nuvens, porém, usam um outro tipo de nível de detalhe: a partir de uma distância fixada previamente, transitam progressivamente de invisíveis para visíveis. Ou seja, implementam a sua modalidade de LOD de maneira diferente da dos objectos genéricos. Da mesma forma, as explosões rescrevem o funcionamento das classes básicas de OBJECTO e não implementam qualquer modalidade de LOD.

Os sistemas de simulação de voo são ideais para aplicação das inúmeras técnicas de níveis de detalhe de terreno. No entanto, dado que o projecto focava o problema do combate aéreo, as técnicas de terreno não eram o seu objectivo central. Assim, foi implementada uma projecção do terreno, segundo um mapa de altitudes regular, com apenas um nível de detalhe e usando-se um efeito de névoa para suavizar a linha do horizonte durante as animações.

## 5. CONCLUSÕES E DESENVOLVIMENTOS FUTUROS

Um ambiente virtual para treino de pilotos e avaliação do seu desempenho durante uma missão foi concebido e implementado com o projecto ACTES no INETI. Embora ainda não se encontre totalmente construído, na sua versão actual o simulador apresenta já uma capacidade de funcionamento em rede com três computadores pessoais, sendo um deles dedicado à supervisão e os outros dois à pilotagem dos respectivos aviões.

Usaram-se computadores Pentium II a 333 MHz, com 64 MB de RAM e 12 MB na placa gráfica 3dfx Voodoo-2. A programação foi realizada em MS-Visual C++ 5.0 com recurso a MFC (*Microsoft Foundation Libraries*) e OpenGL [Fosner97, Wright96]. Para materializar o manche do avião utilizou-se o joystick MS-Sidewinder Force-Feedback Pro. Tendo como suporte uma placa compatível com MS-DirectSound, o realismo da simulação também foi aumentado pela introdução de som (variável) dos motores, dos sinais de radar captados de outros aviões (pelo *Radar Warning Receiver*) e do



próprio deslocamento do ar em torno do avião.

Verificou-se ser imprescindível a utilização de diversos níveis de detalhe (LOD) nos objectos do mundo virtual. Na verdade, experiências feitas simulando dois aviões com 2890 polígonos cada um levaram o sistema a uma cadência média de 28 fotogramas por segundo. Quando o modelo foi simplificado para 296 polígonos, verificou-se que, voando simultaneamente seis aparelhos, a média era

de 34 fotogramas por segundo. Estes valores demonstram, portanto, uma diferença significativa no desempenho do sistema que justifica o uso daquela funcionalidade. No entanto, note-se que, independentemente do nível de detalhe e no sentido de se manter um elevado desempenho, os aviões ainda não são texturados nem sombreados.

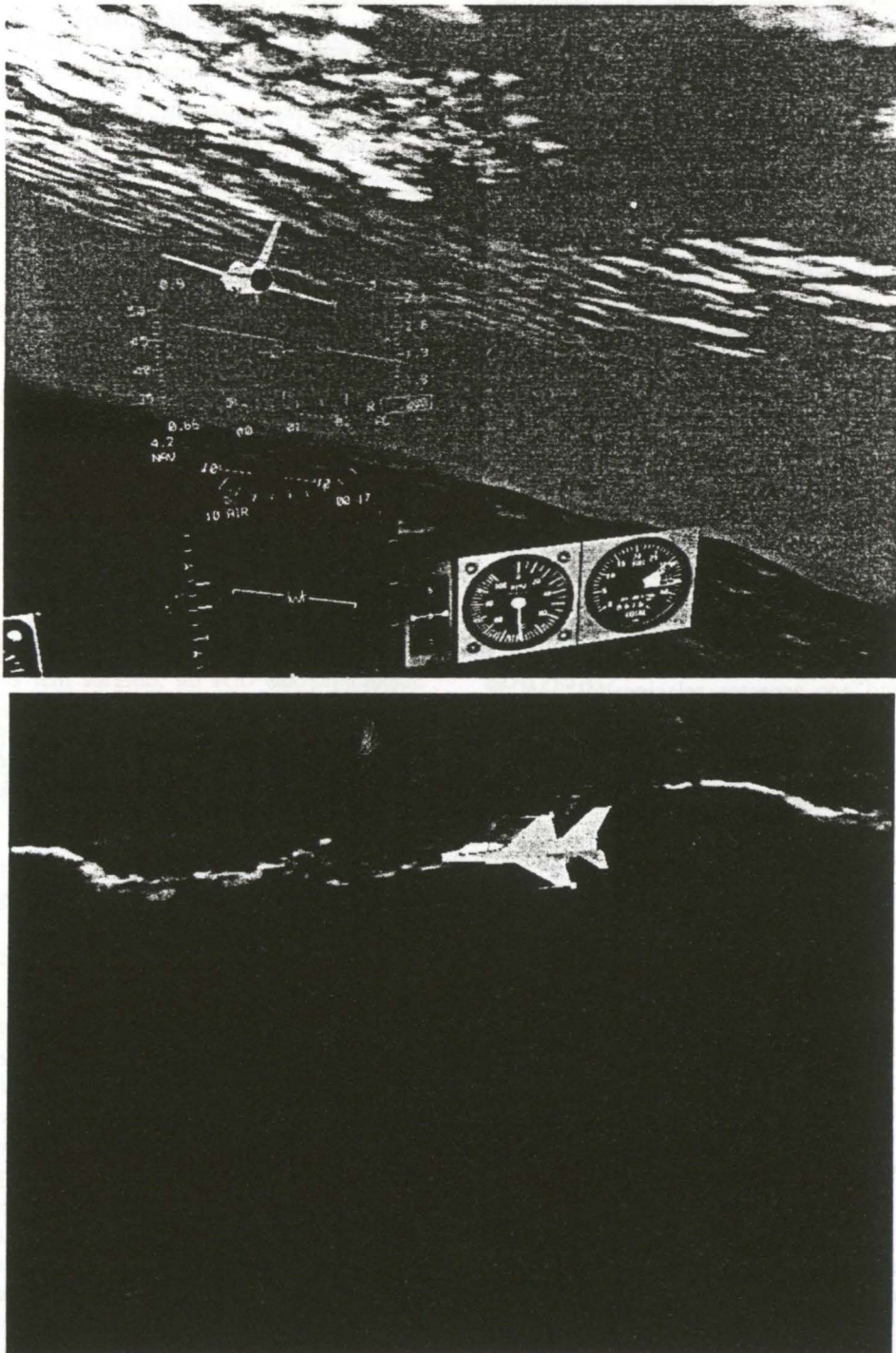


Figura 7: Efeito do desvio do olhar do piloto, num ambiente com nuvens e outros aviões.

Apesar do ACTES ter sido planeado e implementado em equipamento de baixo custo, esta característica não influencia a arquitectura, que é igualmente capaz de se

adaptar a hardware de mais alto nível. Por exemplo, não está excluída a hipótese de se poder vir a utilizar um HMD (*Head-Mounted Display*) com o respectivo sensor

de posição, permitindo que o simples movimento da cabeça do piloto actualize automaticamente a imagem e sirva como processo natural de ajuda à orientação. Mesmo sem HMD, o sistema actual já permite a simulação (através do *joystick*) da rotação da cabeça do piloto, como se comprova na observação das duas imagens da Figura 7. Esta mostra-nos duas fases do desvio do olhar para o lado direito do avião. Repare-se que, na imagem em que ainda se pode ver o painel de instrumentos de voo, a sua posição relativa no ecrã é bem diferente da da Figura 4, em que o piloto está a olhar para a frente da aeronave.

Outros desenvolvimentos futuros poderão passar por: incluir sub-sistemas de visualização panorâmica ou circundante com projecção de múltiplos canais; incluir maior número de comandos constituindo o *cockpit* completo do avião simulado; estar o sistema montado sobre uma plataforma móvel, controlada pelas classes de aerodinâmica do avião; acrescentar uma esquadrilha de aviões virtuais comandados por um sistema com inteligência artificial.

## 6. AGRADECIMENTOS

Este projecto não teria sido possível sem o esforço do Gen. António J. Bispo, cuja coordenação incansável proporcionou sempre os meios para que o trabalho pudesse progredir. Os autores agradecem ainda à Dra. Maria Manuel Freitas a oportunidade de realização de três estágios no INETI para o desenvolvimento do projecto. Também reconhecem como imprescindíveis, tanto a colaboração inicial de Paulo Tavares [Tavares98], no arranque dos trabalhos, como a de Pedro Ávila Lopes, no que diz respeito aos aspectos relacionados com a ligação em rede.

## 7. REFERÊNCIAS

[CGA] The Ultimate Resource for Computer Graphic Artists.  
<http://www.3dcafe.com/asp/freestuff.asp>

- [Fosner97] Ron Fosner. *OpenGL – Programming for Windows 95 and Windows NT*. Third Edition. Addison Wesley Developers Press, 1997.
- [F16] Three-Four-Nine – The Ultimate F-16 Reference. <http://www.f-16.net/>
- [Kermode60] A.C.Kermode. *Flight Without Formulae*. Without Formulae Series - Third Edition, Pitman Paperbacks, 1960.
- [Kermode62] A.C.Kermode. *Mechanics of Flight*. Seventh Edition, Pitman Publishing, 1962.
- [Muchaxo95] J. Muchaxo, E.M. Santos, M.J. Próspero. Navegação em Grandes Terrenos Virtuais com Coerência Espaço-Tempo. *Actas do 7.º Encontro Português de Computação Gráfica*, GPCG – EG, pg. 71-84, Monte de Caparica, 1995.
- [Murray] Richard Murray et al. *Robust Nonlinear Control Theory with Applications to Aerospace Vehicles*. <http://cds.caltech.edu/~vehicles/>
- [Rumbaugh91] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen. *Object-Oriented Modelling and Design*. Prentice Hall, 1991.
- [Sommerville95] Ian Sommerville. *Software Engineering*. 5th Edition, Addison Wesley, 1995.
- [Stevens92] Brian L. Stevens et al. *Aircraft Control and Simulation*. Wiley Interscience, 1992.
- [Tavares98] Paulo L. Tavares. Projecto ACTES. Relatório de Projecto de Licenciatura, DI/FCT/UNL – INETI, 1998.
- [Wright96] Richard S. Wright Jr., Michael Sweet. *OpenGL Superbible – The Complete Guide to OpenGL Programming for Windows NT and Windows 95*. Waite Group Press, 1996.