

# Humanos Virtuais na Plataforma *IViHumans* - a Odisseia da Integração de Ferramentas

Ricardo Abreu

Ana Paula Cláudio

Maria Beatriz Carmo

Faculdade de Ciências da Universidade de Lisboa  
Departamento de Informática  
Campo Grande, 1749-016 LISBOA  
ricardolafabreu@gmail.com, {apc,bc}@di.fc.ul.pt

---

## Resumo

A modelação e a animação de humanos virtuais são tarefas morosas. Apesar de complexa, a integração de diferentes componentes de software é um processo expedito para a construção de humanos virtuais porque permite congrega diversas funcionalidades, nomeadamente reutilizar modelos e animações, de modo a abreviar este processo.

Este artigo descreve a experiência de integração de três componentes – o Blender, o Poser e o OGRE – no contexto de desenvolvimento da plataforma *IViHumans*.

## Palavras-chave

Humanos virtuais, modelação, integração, agentes, expressões

---

## 1. INTRODUÇÃO

A plataforma *IViHumans* (*Intelligent Virtual Humans*) tem vindo a ser desenvolvida com o intuito de suportar a geração de cenas animadas com agentes inteligentes, com especial enfoque em agentes humanóides com capacidade de expressar emoções.

A plataforma é composta por duas camadas: uma camada gráfica e uma camada de Inteligência Artificial. A escolha dos softwares a adoptar para a concretização desta plataforma obedeceu a um conjunto de critérios tais como as capacidades oferecidas, o custo e a existência de uma comunidade activa de utilizadores [Silvestre04]. Os softwares escolhidos, todos de domínio público, foram: o *Blender* [httpBlender] para a modelação 3D, o *OGRE* [httpOGRE] como motor de *rendering* e a bancada de agentes *Jade* [httpJade].

Na camada gráfica desenvolveu-se uma biblioteca com as expressões faciais básicas definidas por Paul Ekman (alegria, tristeza, raiva, medo, nojo e surpresa) e as animações entre estas [Faustino06] e concretizou-se um algoritmo de visão sintética que identifica os objectos visíveis do ponto de vista do agente [Semião06].

Foi definido como objectivo seguinte do trabalho a visualização num cenário do *OGRE* de um modelo completo de humanóide que usa as expressões definidas na biblioteca de expressões faciais básicas.

Para alcançar este objectivo tornou-se necessário juntar um modelo de corpo de um humanóide ao rosto já produzido. Uma pesquisa na Internet levou-nos a concluir

que não existiam modelos gratuitos de humanóides que considerássemos adequados. Tínhamos então duas alternativas: modelar o corpo no *Blender* ou obter modelos já construídos, recorrendo aos serviços de um *designer* profissional ou a um produto, eventualmente comercial, que contivesse modelos já completos. Abandonámos a primeira alternativa por a considerarmos morosa e exigente em termos de habilidade artística. O recurso a um *designer* era uma escolha demasiado onerosa e assim, decidimos procurar um software adequado. Dois produtos foram postos à consideração: o *MakeHuman* [httpMakeHuman] e o *Poser* [httpPoser]. O primeiro foi eliminado porque os modelos de humanóides não tinham roupa, tendo-se então adquirido o *Poser* [httpPoser].

Este artigo descreve os detalhes da experiência efectuada de integração das três ferramentas: o *Blender 2.4.2*, o *Poser 7* e o *OGRE 1.2.4 (Dagon)*.

Na secção 2 descreve-se o funcionamento do programa *Faces* com o qual é possível obter em tempo-real novas expressões faciais que resultam de interpolações percentuais de expressões faciais base. A secção 3 é dedicada ao processo de modelação do corpo que foi efectuada no *Blender* a partir de um corpo exportado do *Poser*. A secção seguinte debruça-se sobre as dificuldades encontradas na integração do modelo do corpo com o modelo da face e descreve os três passos de complexidade crescente que foram adoptados para as superar. A exposição mais detalhada da concretização destes três passos é o conteúdo da secção 5. A secção final é dedicada ao trabalho futuro a curto e médio prazo.

## 2. EXPRESSÕES FACIAIS

O relacionamento entre seres humanos assenta, em grande medida, na capacidade de interpretação dos sentimentos e emoções alheias. A interacção entre humanos é moldada pela integração de diversos estímulos no reconhecimento de padrões de comportamento, transmitidos, nomeadamente, por expressões faciais. Ao comunicar, captamos o conteúdo emocional expresso pelo interlocutor através da distinção de traços faciais gerais que nos permitem recolher um padrão semelhante a partir de diferentes caras. Na realidade, somos capazes de reconhecer as mesmas expressões faciais em representações mais ou menos fiéis à realidade, como banda desenhada, ou mundos tridimensionais virtuais.

Na plataforma *IViHumans* pretendemos incluir a expressão facial como forma de ilustração das emoções dos agentes que povoam o ambiente virtual. Com o objectivo de diversificar as emoções que podem ser expressas, criámos uma interface gráfica de auxílio à construção de expressões através da mistura de expressões base. Com este programa, designado *Faces*, podem definir-se novas expressões faciais à custa da variação de parâmetros relativos às expressões básicas, observando-se o efeito em tempo real e enriquecendo-se assim, em qualquer altura, a biblioteca de expressões subjacente (figura 1).

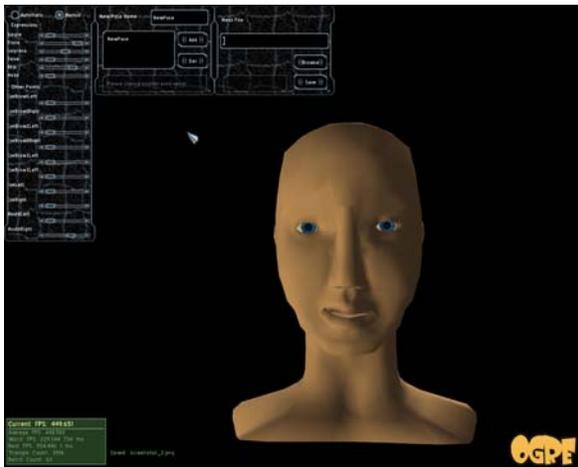


Figura 1: Interface do programa *Faces*.

Cada expressão básica foi construída no *Blender* por meio da alteração da malha original da face concebida com a expressão neutra. Para cada expressão distinta são guardados os vectores de deslocamento, um para cada vértice cuja posição tenha sido alterada. O conjunto de vectores deslocamento que usamos para produzir uma expressão é frequentemente denominado de pose. Aplicando os deslocamentos registados aos vértices correspondentes obtém-se a expressão básica respectiva. Este método pode ainda ser alargado de maneira a albergar a especificação de uma intensidade para cada expressão e, uma vez registadas as poses que codificam as expressões, podem assim gerar-se expressões intermédias. Seguindo este caminho, se se tiver definido, por exemplo, uma pose que codifique a expressão de surpresa, pode-se produzir muito facilmente uma expressão de “80% de sur-

presa” sem qualquer esforço adicional de modelação. Para isso, basta aplicar os deslocamentos resultantes da redução, em um quinto, dos vectores registados pela pose em questão.

No programa *Faces* este conceito é concretizado com recurso à API do *OGRE*. Através do simples maneja-mento de botões de *scroll*, um para cada expressão base definida, o utilizador instrui a aplicação para que sejam geradas expressões, com intensidades entre 0 e 1 correspondentes às posições dos *scrolls*. Quando se faz variar a posição de mais do que um botão de *scroll*, o resultado é a mistura das expressões deste modo activadas. A mistura de expressões resulta apenas da soma algébrica dos vectores subjacentes às poses correspondentes, ou seja, em rigor, o deslocamento aplicado a um vértice é o con-sequente da soma dos vectores de deslocamento, para cada pose activa, que lhe estão afectos, multiplicados pela intensidade especificada pela posição dos botões de *scroll*.

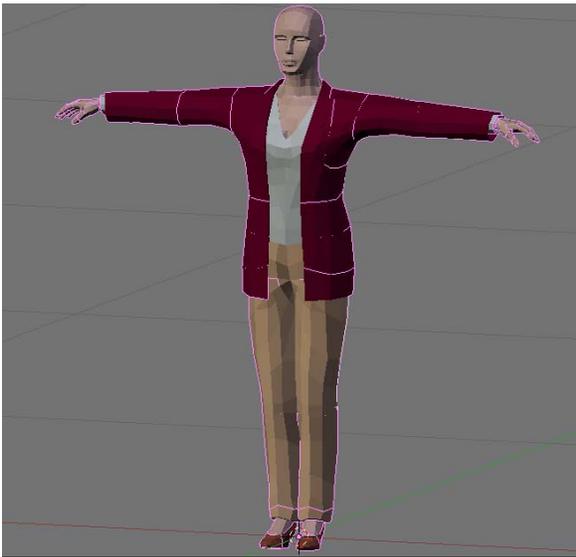
À medida que o utilizador faz variar as posições dos *scrolls*, os efeitos são gerados automaticamente, o que lhe confere a possibilidade de experimentar e aperfeiçoar as novas expressões que deseja criar.

## 3. MODELAÇÃO DO CORPO

A modelação de humanos virtuais credíveis é um proces-so moroso e que requer habilidade artística. Consideran-do que a arte da criação, de raiz, de personagens virtuais não faz parte dos nossos objectivos, decidimos usar um modelo já existente como ponto de partida e refiná-lo. Após uma breve análise dos recursos publicamente dis-poníveis, acabámos por escolher uma ferramenta direc-cionada para a criação e animação de personagens vir-tuais – o *Poser*. Actualmente, esta ferramenta fornece, à partida, uma quantidade razoável de conteúdos necessá-rios à tarefa em questão e possibilita a sua parametriza-ção, adaptação e conjugação, ainda que de forma algo limitada. Embora o *Poser* disponibilize personagens humanas completas com grande qualidade e detalhe, optámos por adaptar e melhorar um modelo de baixa resolução. Esta opção foi tomada devido às restrições inerentes ao processamento gráfico em tempo real e ten-do em conta que o realismo final não seria prejudicado, uma vez que a cara – a zona que requer uma maior defi-nição – seria substituída. Partimos do modelo apresenta-do na figura 2, exportámo-lo para o formato do *Wave-front* (extensão .obj) e importámo-lo no *Blender*.

Numa primeira aproximação, estabelecemos objectivos mínimos para o aspecto do humano, tal como para o carácter de realismo do seu movimento. Durante a mode-lação, procurámos refinar o modelo, dando especial aten-ção ao melhoramento da geometria e à base da animação e relegando para segundo plano a definição de materiais e as reacções dinâmicas de componentes que tencionamos tornar deformáveis, como a roupa e o cabelo. Embo-ra tenhamos já definido, no *Blender*, materiais relativamen-te complexos para a pele e para os olhos estes não podem ser directamente exportados para o *OGRE* estan-do destinados a ser usados como guia para uma definição

futura de materiais efectivamente utilizáveis. Por agora, reduzimos a definição dos materiais a cores e à aplicação pontual de imagens em texturas.



**Figura 2: Visualização, no Blender, do modelo acabado de importar do Poser.**

Criámos, ainda no *Poser*, uma animação simples da acção de andar para a frente num estilo feminino, aplicada a um esqueleto adequado à malha poligonal, através da parametrização de um movimento base. Não obstante, decidimos, quanto ao movimento, usar o *Poser* apenas para a variação de parâmetros relativos, por exemplo, à amplitude do movimento dos braços ou à intensidade do balanceamento da cabeça. Tal como se passa com a edição do esqueleto e das malhas poligonais, com a criação de expressões faciais ou com a aplicação de texturas, o *Poser* é bastante limitado, relativamente ao *Blender* ou a outra ferramenta de modelação 3D, no que concerne à edição de animações a um baixo nível. Deparamo-nos ainda com frequentes *bugs* quando tentámos, no *Poser*, editar o movimento de conjuntos de polígonos entre *frames* ou estabelecer uma ligação do esqueleto à malha poligonal. Posto isto, estas foram tarefas que optámos por realizar no *Blender*. Para tal, foi necessário exportar o esqueleto, com a animação respectiva, para o formato *motion capture* (extensão .bvh) e importá-lo, subsequentemente, no *Blender*.

Para associar a malha poligonal ao esqueleto, seguimos o processo tradicional que consiste na ligação de cada vértice da malha poligonal a um ou mais ossos do esqueleto, caracterizando cada relação entre vértice e osso com um determinado peso. O deslocamento de cada vértice é então função do peso com que está associado aos ossos e do movimento destes. Uma vez concluído convenientemente este procedimento, o humano virtual passou a usufruir fluentemente do movimento do seu esqueleto (figura 3).



**Figura 3: Frame da animação do humano virtual, vista no Blender.**

As ligações entre ossos e vértices foram estabelecidas como resultado de um processo de tentativa e erro porque as decisões inerentes são intuitivas e não matematicamente determinadas. Procedendo iterativamente à correcção de movimentos disformes e de interpenetrações entre os componentes da malha – consequências indesejadas de escolhas impróprias de intensidades e posições relativas – foi alcançado um resultado considerado razoável. Além disto, as exportações com origem no *Poser*, não determinam uma malha poligonal e um esqueleto imediatamente aptos a ser associados. Com efeito, foi necessário aperfeiçoar as dimensões relativas da malha e do esqueleto e colocá-los previamente em posturas e orientações idênticas.

Quanto aos movimentos em si, começámos por definir apenas um, correspondente, como referido anteriormente, à acção de andar, e realizámos pequenas correcções na animação correspondente. Tencionamos concluir esta tarefa numa próxima iteração, bem como adicionar outro tipo de movimentos que possam, eventualmente, ser conjugados.

#### 4. INTEGRAÇÃO DO CORPO E DA FACE

Antes de associar a malha poligonal ao esqueleto, foi necessário concluí-la, nomeadamente através da adaptação mútua da face, que já tinha sido concebida, e do corpo, resultante das alterações efectuadas sobre o modelo original do *Poser*. Foi nesta etapa que surgiu um dos maiores contratemplos com que nos defrontámos e que consiste na indisponibilidade, no *Blender*, de um método de junção entre objectos em que pelo menos um deles tenha poses definidas. Esta funcionalidade seria da maior importância para conclusão do modelo do humano virtual e a sua inexistência representou a necessidade de dispender mais tempo para contornar o problema.

Uma vez que o *Blender* dispõe de uma interface de *scripting*, através de uma API em *Python*, pesquisámos por *scripts* que alguém com um problema análogo tivesse já desenvolvido e que nos pudessem ajudar a ultrapassar

este obstáculo. Não tendo encontrado nenhum que nos pudesse auxiliar nesta questão, considerámos ainda a hipótese de o criar, mas acabámos por a deixar de lado, por constituir um desvio excessivo face ao caminho inicialmente pensado, devido à complexidade inerente e por oferecer menos garantias de sucesso.

Para superar este impedimento dividimos a resolução do problema em três passos de complexidade crescente:

1. Posicionar e ajustar a face e o corpo de modo a juntá-los num único objecto. Como já foi referido, neste caso não se juntam as poses e, por isso, a face tem apenas a expressão base, mantendo-se a animação do esqueleto.
2. Posicionar e ajustar a face e o corpo de modo a que pudessem ser vistos como apenas um objecto, apesar de serem objectos distintos, alcançando assim uma solução de compromisso com algumas imperfeições ao nível do realismo. Deste modo mantêm-se as expressões faciais mas é necessário encapsular os dois objectos numa entidade abstracta, aquando da sua integração na plataforma *IViHumans*.
3. Realizar um procedimento que junte de uma forma coerente, num único objecto, a face e o corpo, mantendo as poses.

A execução dos passos anteriores foi conduzida pela preocupação de produzir modelos que possam vir a ser usados no *OGRE*. Ao longo daqueles procedimentos fomos desenvolvendo pequenos programas com recurso à API do *OGRE*, de modo a testar os resultados e a aperfeiçoá-los de acordo com o que então se revelasse conveniente. A solução do passo 2 seria adequada, ao ponto de possibilitar a criação de novas personagens apenas pela troca de caras (num estilo *plugin*) não fora o facto de o algoritmo de *shading* contar com a direcção das normais às faces adjacentes para realizar uma interpolação e colorir cada *pixel* da face considerada. Devido à estratégia seguida pelo algoritmo de sombreamento, que possibilita um efeito de *smoothing*, um humano virtual construído com recurso a malhas poligonais distintas evidencia o carácter descontínuo do limite entre os dois objectos correspondentes. Mesmo não sendo a falha entre os dois sub-objectos obtidos por este processo excessivamente vincada, considerámos que os resultados limitariam o realismo que poderia, de outro modo, ser alcançado e decidimos, portanto, procurar um processo alternativo para atingir o nosso objectivo (que corresponde ao passo 3). O procedimento inerente é resumidamente descrito na secção seguinte.

## 5. EXPORTAÇÃO E INTEGRAÇÃO DO HUMANO VIRTUAL NA PLATAFORMA *IVIHUMANS*

A transferência de modelos do *Blender* para o *OGRE* implica a realização de várias operações. Há um conjunto de operações básicas que são complementadas com operações adicionais para atingir um resultado final coerente.

### 5.1 Processo base de transferência do modelo do *Blender* para o *OGRE*

A criação de humanos virtuais tem por fim a sua integração na plataforma *IViHumans*, construída sobre a API do *OGRE*. Para que as animações associadas ao esqueleto estejam disponíveis, é necessário que este seja exportado do *Blender* conjuntamente com o modelo a que se aplica. Tanto para a malha poligonal do humanóide como para o seu esqueleto é assim produzido um ficheiro XML que deve ainda ser processado previamente por uma ferramenta distribuída em conjunto com o *OGRE* – *OGREXMLConverter*. Esta, por seu turno, gera novos ficheiros num formato binário em função do conteúdo dos ficheiros XML fornecidos como input. Estes ficheiros são rapidamente processados pela plataforma.

Os ficheiros XML definem, entre outras coisas, a geometria, as poses e as animações dos modelos, no caso das malhas poligonais, e a hierarquia dos ossos, bem como as suas animações, no caso dos esqueletos. Estes ficheiros são resultantes da execução de um *script* que comunica com o *Blender* (através da sua interface em *Python*) e que obtém a informação necessária. Neste processamento inclui-se um conjunto de transformações indispensáveis que lidam com a diferença de referenciais usados pelo *Blender* e pelo *OGRE*. A figura 4 mostra um esquema que ilustra o processamento base de exportação.



Figura 4: Processo de transferência de um modelo do *Blender* para o *OGRE*.

### 5.2 Processamento complementar do modelo exportado

Infelizmente, durante os processos envolvidos na modelação, brevemente descritos anteriormente, não nos apercebemos de uma questão de extrema importância: a informação espacial a que o *script* de exportação acede não contempla transformações de escala, rotações ou translações que tenham sido aplicadas ao modelo como um todo, sem que explicitamente se tenha dado a instrução de alargar os seus efeitos directamente às primitivas que os compõem. Na realidade, o *Blender* separa a informação das características da malha poligonal, como sejam as posições dos seus vértices ou os deslocamentos que codificam as suas poses, da informação que é efectivamente usada para o *rendering*. Com efeito, uma qualquer transformação aplicada à totalidade de um objecto não tem influência sobre, por exemplo, as posições registadas para cada um dos vértices. Em vez disso, estas transformações são tidas como ulteriores à modelação *per se* e guardadas independentemente. Os seus resultados são só calculados durante a geração da imagem, sendo então guardados em *buffers* cuja persistência é volátil. Por conseguinte, é natural que o resultado da exportação de um modelo apresente uma posição, uma dimensão e uma orientação incongruentes com o que é visualizado

no *Blender*. Apesar de este software disponibilizar um meio de estender os efeitos das referidas transformações directamente à definição da malha poligonal, esta funcionalidade nem sempre opera correctamente. Em particular, no caso do nosso modelo, o seu uso tem efeitos nefastos sobre o sentido das normais a uma grande parte dos milhares de faces que o constituem. Por esta razão, e porque a alteração manual de milhares de normais é impraticável, fomos obrigados a abandonar este método.

Mais uma vez, teria sido possível resolver este problema através de um outro *script* que modificasse correctamente todos os aspectos que compõem a malha poligonal. Não encontramos, no entanto, nenhum *script* já desenvolvido que executasse eficazmente esta operação e optámos, mais uma vez, por não o criar, pelas mesmas razões que acima referimos. Além disso, as modificações que tínhamos, de qualquer modo, de levar a cabo sobre a malha podiam ser realizadas ao nível do XML. Criámos então um programa *Java* que processa os ficheiros XML produzidos e que os altera em função de parâmetros que permitem especificar as transformações geométricas a realizar. Este programa contempla ainda as consequências da operação sobre um referencial diferente do usado no *Blender* e realiza as transformações adicionais que este facto acarreta, para que o modelo processado tenha uma aparência análoga à que é observada no *Blender*. Executando este programa sobre os ficheiros XML criados imediatamente após a exportação, com parâmetros correspondentes às transformações aplicadas no *Blender* sobre os objectos respectivos, conseguimos gerar novos ficheiros que correspondem ao que é visualizado no programa de modelação, a menos de pequenas imperfeições que surgem como consequência da propagação de erros resultante da baixa precisão com que o *Blender* disponibiliza determinados dados numéricos – apenas com 4 dígitos significativos. Felizmente, estes detalhes são imperceptíveis. Assim, enquanto antes tínhamos um humanóide de proporções irrealistas, cujo esqueleto estava rodado, dimensionado e deslocado relativamente à malha poligonal, obtemos então um modelo plenamente coerente, pronto a ser integrado na plataforma *IViHumans*. O passo 1, apresentado na secção anterior ficou assim completo.

Para realizar o passo 2, ajustámos as duas malhas no *Blender* mas não as juntámos numa única malha. Pelo mesmo processo de exportação são produzidos dois ficheiros XML, um para a face e outro para o corpo. Ambos os ficheiros sofreram os processamentos complementares descritos anteriormente. Ao nível do *OGRE*, as duas malhas são encapsuladas numa única entidade abstracta.

### 5.3 Extensão do processamento complementar

Para alcançar os objectivos do passo 3, partiu-se dos ficheiros XML obtidos com o passo 2 para a face e para o corpo e criou-se um único ficheiro, através de uma sequência de operações que se expõe a seguir.

Um ficheiro XML que conserve a definição de uma malha poligonal tem uma estrutura que compreende a

divisão da malha em submalhas, de acordo com os materiais associados a cada região. As várias submalhas são referenciadas em diferentes zonas do ficheiro através de índices que se prendem com a ordem da sua declaração, tal como se passa para grande parte das restantes propriedades. Foi portanto fácil adicionar as submalhas dos olhos, dos lábios, dos dentes e do cabelo à malha do corpo. Dado que os seus materiais são diversos de qualquer dos materiais do corpo, bastou copiar as secções correspondentes entre os dois ficheiros XML (do da face para o do corpo), corrigir as referências às submalhas e acrescentar as declarações necessárias para associar os novos vértices ao osso da cabeça, tendo cada uma das novas associações um peso de 100%.

A etapa que nos criou mais dificuldades foi a da adição, ao modelo do corpo, da submalha da face coberta pelo material concebido para a pele. Sendo este o mesmo material já usado por uma das submalhas do corpo, e uma vez que o algoritmo de sombreamento deveria encarar a ligação dos polígonos adjacentes da face e do corpo como contínua, não seria suficiente seguir o mesmo procedimento que para as restantes submalhas da face. Foi, antes, imprescindível fundir as submalhas da pele, dos lados do corpo e da cara, numa única. Além da substituição de referências e da associação dos vértices da face ao esqueleto, tivemos de conferir um carácter de continuidade aos limites adjacentes da face e do corpo, através da “fusão” dos vértices fronteiros, de forma a que a imagem gerada não evidenciasse uma falha entre os polígonos. Para atingir este propósito era forçoso identificar os vértices limite. Como o *Blender* não permite aplicar materiais a vértices de modo independente das faces que os compõem, nem exportar para XML malhas poligonais que contenham apenas vértices e arestas mas que não englobem nenhum polígono, tivemos de arranjar uma alternativa. Decidimos então criar um *script* elementar, cuja única função é imprimir para um ficheiro as coordenadas dos vértices que compõem um modelo. Pensamos que esta opção se justificou devido à sua simplicidade, que contrasta com a dificuldade das hipóteses de desenvolvimento de *scripts* anteriormente apresentadas. Eliminando, das duas malhas, todos os vértices excepto os que queríamos identificar nos ficheiros XML e correndo o *script*, alcança-se a informação pretendida.

Dispondo das coordenadas dos vértices que queríamos encontrar, no seio de milhares, pudemos criar outro programa *Java* para os identificar. A execução deste programa passa por percorrer uma submalha, num ficheiro XML, calculando a distância entre cada vértice aí definido e cada um dos vértices constantes de um outro ficheiro, fornecido como input, que contenha as coordenadas dos vértices que se pretenda identificar. Consideram-se quaisquer dois vértices como idênticos quando a distância euclidiana entre eles é inferior a um dado limiar. O programa mantém um contador que indica a posição da declaração de cada vértice no ficheiro XML. Assim, quando um vértice é encontrado, sabe-se o seu identificador numérico que, tal como no caso das submalhas, não é mais do que a posição em que ocorre. Conhecendo

o emparelhamento próprio entre os vértices que definiam a fronteira entre a cara e o corpo, modificámos as faces do lado do corpo, de maneira a que fossem construídas com base nos vértices limite da face. Pudemos então adicionar as expressões da cara ao ficheiro XML, corrigindo os índices das submalhas referidas e, para a região da pele, alterando também os identificadores dos vértices a deslocar. Esta última edição resumiu-se à soma do número de vértices da pele, do lado corpo, ao índice de cada vértice da face usado para a definição das poses. Isto foi feito, novamente, através de um pequeno programa *Java*. A figura 5 mostra *screenshots* do humano virtual com animação de expressões, no ambiente *OGRE*.



Figura 5: *Screenshots* do humano virtual com animação de expressões, no ambiente *OGRE*.

## 6. TRABALHO FUTURO

Com o modelo do nosso humanóide completo, de forma coerente e funcional, perspectivamos, para um futuro próximo, integrá-lo na plataforma *IViHumans*, associando à sua representação gráfica a capacidade de percepção do mundo e um módulo de inteligência artificial. A percepção do ambiente será realizada através de um algoritmo de visão sintética já referido anteriormente. Este

algoritmo permite detectar quais os objectos visíveis por cada agente, num determinado instante ou durante um curto intervalo de tempo. Através da camada de inteligência artificial, concebida de um ponto de vista abrangente, cada agente será capaz de planejar trajectos e de exprimir emoções adequadas às circunstâncias percebidas.

Uma vez concluída esta primeira iteração, apresentam-se-nos várias ideias para um aprofundamento posterior. Destas, destacamos como incontornáveis o refinamento de materiais e a adição de texturas, assim como a concepção de mais movimentos pré-definidos, de modo a construir um leque suficientemente abrangente para que os agentes possam operar realisticamente sobre o mundo virtual através de um conjunto adequado de acções compostas. Consideramos também imprescindível a introdução de novas personagens.

Outros possíveis desenvolvimentos envolvem a extensão das funcionalidades da camada de inteligência artificial, nomeadamente no que toca à interacção entre agentes, entre agentes e *smart objects* [Abaci05], e à geração automática de animações em tempo real, tanto ao nível das expressões faciais como ao dos movimentos do esqueleto.

## 7. REFERÊNCIAS

- [Abaci05] T. Abaci, J. Ciger and D. Thalmann. Planning with Smart Objects. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, WSCG'05, Czech Republic, February 2005
- [Faustino06] J. Faustino, A.P. Cláudio, M.B. Carmo. *Faces*- Biblioteca de Expressões Faciais. Actas da 2ª Conferência Nacional em Interacção Pessoa-Máquina, Interacção 2006, pp 139-142, Braga, 16-18 de Outubro de 2006
- [Semião06] P.M. Semião, M.B. Carmo, A.P. Cláudio. Algoritmo de Visão para Humanos Virtuais. Actas da 2ª Conferência Nacional em Interacção Pessoa-Máquina, Interacção 2006, pp 133-138, Braga, 16-18 de Outubro de 2006
- [Silvestre04] M. Silvestre, M. Pinto-Albuquerque, M.B. Carmo, A.P. Cláudio, J.D. Cunha, H. Coelho. Arquitectura de Suporte à Geração de Cenas Animadas com Agentes Inteligentes. Relatório Técnico do Departamento de Informática da Faculdade de Ciências da Universidade de Lisboa, DI-FCUL TR-04-7, Julho 2004
- [httpBlender] <http://www.blender3d.org/>
- [httpJade] <http://jade.tilab.com/>
- [httpLabMAG] <http://labmag.di.fc.ul.pt/>
- [httpMakeHuman] <http://www.dedalo-3d.com/>
- [httpOGRE] <http://www.ogre3d.org/>
- [httpPoser] <http://www.e-frontier.com/>