

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**PERSONAGENS VIRTUAIS EM AMBIENTES
VIRTUAIS (PLATAFORMA IVIHUMANS)**

João Miguel Pires de Carvalho

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2010

UNIVERSIDADE DE LISBOA
Faculdade de Ciências
Departamento de Informática



**PERSONAGENS VIRTUAIS EM AMBIENTES
VIRTUAIS (PLATAFORMA IVIHUMANS)**

João Miguel Pires de Carvalho

PROJECTO

Projecto orientado pela Prof. Doutora Ana Paula Boler Cláudio

MESTRADO EM ENGENHARIA INFORMÁTICA
Especialização em Sistemas de Informação

2010

Agradecimentos

À minha orientadora Ana Paula Cláudio pelo apoio dado ao longo deste ano lectivo, e pela sua dedicação e disponibilidade que permitiram a concretização desta etapa da minha formação académica.

Ao pessoal do LabMAg que devido a sua boa disposição e espírito de camaradagem, tornaram a realização de todo o projecto mais fácil e agradável.

A minha família e amigos pelo seu apoio incondicional não só durante este ano lectivo mas durante todo o meu percurso académico.

À minha família.

Resumo

Este projecto enquadra-se no desenvolvimento da plataforma *IViHumans*, que suporta a representação de Humanos Virtuais Inteligentes inseridos em ambientes virtuais. Neste contexto é fundamental conceber modelos de humanos virtuais credíveis e animados para povoar esses mundos sintéticos, uma tarefa que exige um esforço de modelação considerável e um consumo de tempo significativo. Adicionalmente, é importante dominar a ferramenta de modelação 3D com que se trabalha, e ter um bom desempenho artístico na sua utilização.

Com o intuito de tornar esta tarefa menos trabalhosa para o animador, foi concebida uma abordagem semi-automática para a aplicação de animações idênticas a diferentes modelos de humanos virtuais. A solução desenvolvida facilita a tarefa do animador, mas não dispensa a sua intervenção para proceder a pequenos ajustes, sendo por isso “semi-automática”. Foi implementada no Autodesk 3D Studio Max 2010, recorrendo às suas capacidades de modelação e à linguagem interna de *scripting* desta ferramenta.

Palavras-chave: Humanos virtuais, animação, captura de movimento.

Abstract

This project is part of the *IViHumans* platform, which supports the representation of Intelligent Virtual Humans inserted in virtual environments. One of our main concerns is to obtain credible and animated virtual human models to populate these synthetic worlds, a task that requires considerable modeling effort and significant time consumption. Furthermore, it is important to master the use of a 3D modeling tool and to possess good modeling skills.

Aiming to ease the animator's task, we implemented a semi-automatic approach that supports the application of identical animations to different virtual humans models. The developed solution eases the animator's task, but his intervention is still needed to perform some minor adjustments, it is therefore "semi-automatic". It was developed in Autodesk 3D Studio Max 2010, using its modeling capabilities and its scripting language.

Keywords: Virtual humans, animation, motion capture.

Conteúdo

Lista de Figuras	xvii
1 Introdução	1
1.1 Motivação	1
1.2 Objectivos	2
1.3 Contexto e Contribuições do Trabalho	2
1.4 Estrutura do documento	3
2 Fundamentos e Estado da Arte	5
2.1 Animação	5
2.2 Animação Facial	6
2.3 Fases de uma Animação 3D	6
2.4 Modelos de Deformação Facial	9
2.4.1 <i>Moving Picture Experts Group Layer 4 (MPEG-4)</i>	9
2.4.2 <i>Facial Action Coding System (FACS)</i>	10
2.5 Técnicas de Animação 3D	11
2.5.1 <i>Motion Capture (Mocap)</i>	11
2.5.2 <i>Morph Target Animation</i>	12
2.5.3 <i>Skeleton Animation</i>	13
2.6 Trabalho Relacionado com a Animação 3D	14
2.6.1 <i>Motion capture</i>	14
2.6.2 <i>Morph Target Animation</i>	15
2.6.3 <i>Skeleton Animation</i>	17
2.7 Conclusão	19
3 Plataforma IViHumans	21
3.1 Introdução	21
3.2 Arquitectura	21
3.3 Percepção, Movimento e Expressões Faciais	23
3.3.1 Percepção	23
3.3.2 Movimento	24

3.3.3	Expressões Faciais	25
3.4	Conclusão	25
4	Trabalho Desenvolvido	27
4.1	Introdução	27
4.2	Abordagens Desenvolvidas	27
4.2.1	Primeira Abordagem: Usando Vértices	29
4.2.2	Segunda Abordagem: Usando <i>Envelopes</i>	29
4.3	Interface Visual	32
4.3.1	Introdução	32
4.3.2	Construção da Interface	32
4.4	Modelos Virtuais Utilizados	35
4.5	Ferramentas Utilizadas	37
4.6	Ambiente de Desenvolvimento	38
4.7	Conclusão	38
5	Testes Efectuados	39
5.1	Introdução	39
5.2	Testes no 3D Studio Max	39
5.2.1	Inserir um objecto estático na plataforma	39
5.2.2	Inserir um humano virtual animado na plataforma	40
5.3	Comparação de Expressões Faciais	44
5.4	Conclusão	47
6	Conclusão e Trabalho Futuro	49
6.1	Conclusões	49
6.2	Perspectivas Futuras	50
A	Manual de Utilizador	51
A.1	Introdução	51
A.2	Instalação	51
A.3	Menu da Interface	52
A.4	<i>Phase 1 - Rigging & Skinning & Weighting</i>	53
A.4.1	Passo 1	54
A.4.2	Passo 2	54
A.4.3	Passo 3	54
A.4.4	Passo 4	54
A.4.5	Passo 5	55
A.4.6	Passo 6	55
A.4.7	Passo 7	55

A.5	Phase 2 - Animation	56
A.5.1	Painel de controlo	57
A.5.2	Animação	59
B	Ficheiros de Configuração	61
B.1	Ficheiro <i>.Mcharacter</i>	61
B.2	Ficheiro <i>.IViHuman</i>	62
C	Testes Efectuados no Blender	63
C.1	Inserir um objecto estático na plataforma	63
C.2	Inserir um humano virtual animado na plataforma	65
	Bibliografia	73

Lista de Figuras

2.1	Animação em <i>Stop Motion</i> [1].	5
2.2	As Seis Expressões Faciais Primárias [2].	6
2.3	Esqueleto <i>Biped</i> de um cão [3].	7
2.4	Esqueleto <i>Biped</i> ajustado à malha poligonal do personagem virtual [4].	7
2.5	Representação de dois tipos de <i>Weighting</i> [5] [6].	8
2.6	Pontos de referência representados na face de um modelo.	9
2.7	<i>Action Units</i> e respectivas interpretações [7].	10
2.8	<i>Motion Capture</i> [8] [9].	11
2.9	Utilização de <i>Morph Target Animation</i> no <i>3D Studio Max</i> [10].	12
2.10	Representação da malha poligonal e do esqueleto no <i>3ds Max</i>	13
2.11	Expressão facial final do modelo.	14
2.12	Transferência de expressões faciais.	15
2.13	<i>Morphed face</i> e face alvo respectiva.	16
2.14	Mapeamento de duas expressões faciais.	16
2.15	Representação do <i>Pinocchio</i>	17
2.16	Representação da deformação dos diferentes objectos alvo.	18
2.17	Vista geral do funcionamento do algoritmo.	18
3.1	Arquitectura da plataforma <i>IViHumans</i>	22
3.2	Diagrama de classes - Visão e Memória.	23
3.3	Diagrama de classes - <i>IViEntity</i>	24
4.1	Esqueleto <i>Biped</i> do <i>3ds Max</i> com ossos extra na face.	28
4.2	Funcionalidade <i>Load Envelopes</i> do <i>3ds Max</i>	30
4.3	Funcionalidade do <i>3ds Max</i> , <i>Load Motion Capture File</i>	31
4.4	Menu inicial da interface.	32
4.5	<i>Phase 1 - Rigging & Skinning & Weighting</i>	33
4.6	<i>Phase 2 - Animation</i>	34
4.7	Modelos de corpo inteiro utilizados.	35
4.8	Modelos de corpo inteiro utilizados.	36
4.9	Modelos de face utilizados.	36
5.1	O modelo da mulher inserido na plataforma <i>IViHumans</i>	40

5.2	<i>OgreMax Scene Exporter for 3DS Max.</i>	41
5.3	<i>Add Mesh Animation.</i>	42
5.4	Animação do modelo da mulher na plataforma.	43
5.5	Modelos utilizados.	44
5.6	Modelos distintos mostrando Alegria.	44
5.7	Modelos distintos mostrando Tristeza.	45
5.8	Modelos distintos mostrando Raiva.	45
5.9	Modelos distintos mostrando Medo.	46
5.10	Modelos distintos mostrando Nojo.	46
5.11	Modelos distintos mostrando Surpresa.	46
A.1	Menu inicial da interface.	52
A.2	Botão < <i>Phase 1 - Rigging & Skinning & Weighting</i> >.	52
A.3	Botão < <i>Phase 2 - Animation</i> >.	53
A.4	Menu da primeira fase da interface.	53
A.5	Botão < <i>Load Skeleton</i> >.	54
A.6	<i>Drop-box</i> < <i>Import Model</i> >.	54
A.7	Botão < <i>Add 'Skin' Modifier</i> >.	54
A.8	Botão < <i>Add Bones</i> >.	55
A.9	Botão < <i>Load Envelope File</i> >.	55
A.10	<i>Editbox</i> < <i>filename</i> >, botão < <i>Save & Exit</i> > e botão < <i>Save & Go to Phase2</i> >.	55
A.11	Menu da segunda fase da interface.	56
A.12	Controlo de Translação.	57
A.13	Controlo de Selecção.	57
A.14	Controlo de Rotação.	58
A.15	Controlo de Reset.	58
A.16	Controlo Hide Bones.	58
A.17	<i>Editbox</i> do número de <i>frames</i> .	59
A.18	Botão < <i>Auto Key</i> >.	59
A.19	<i>Time Slider</i> do número de <i>frames</i> .	59
A.20	Atalhos de navegação na animação.	60
A.21	<i>Dropbox</i> < <i>Load Animation</i> >.	60
A.22	<i>Editbox</i> do nome do ficheiro.	60
B.1	Exemplo de um ficheiro <i>.mcharacter</i> .	61
B.2	Exemplo de um ficheiro <i>.ivihuman</i> .	62
C.1	<i>Ogre Meshes Exporter</i> .	64
C.2	O modelo do homem de negócios inserido na plataforma <i>IViHumans</i> .	65
C.3	Esqueleto e face do modelo.	66

C.4 <i>Ogre Meshes Exporter</i>	67
---	----

Capítulo 1

Introdução

A animação digital é a arte de criar imagens em movimento utilizando computadores e é também um subcampo da computação gráfica e da animação. O desenvolvimento da animação digital aumentou muito a velocidade do processo de animar personagens virtuais, eliminando tarefas mecânicas e repetitivas.

A animação facial é o elemento crucial para transmitir emoções e personalidade a um personagem virtual. A indústria de entretenimento exige resultados de alta qualidade e conduz os esforços de investigação para automatizar o processo de caracterização, obtendo deste modo uma animação mais realista de personagens 3D. Com o intuito de tornar esta tarefa menos trabalhosa para o animador, concebemos uma abordagem semi-automática para a aplicação de animações faciais idênticas a modelos distintos de humanos virtuais. A nossa solução facilita a tarefa do animador, mas não dispensa a sua intervenção para proceder a pequenos ajustes, por isso a adjectivamos de “semi-automática”. Além disso a nossa solução tem a vantagem de também funcionar para animação corporal.

O trabalho insere-se no projecto IViHumans (Intelligent Virtual Humans), uma plataforma que tem vindo a ser desenvolvida no LabMAG para suportar o desenvolvimento de diversas aplicações com humanos virtuais inteligentes.

Este capítulo apresenta a motivação e objectivos do trabalho, assim como a organização do documento.

1.1 Motivação

Os ambientes virtuais habitados por agentes com forma e comportamento que, com diversos graus de realismo, simulam os humanos são cada vez mais comuns e têm muitas aplicações práticas hoje em dia, em áreas como o entretenimento, a educação, o treino industrial e a reconstituição de ambientes históricos. Estes ambientes são cada vez mais populares na internet, e o seu potencial como ferramentas úteis para estudos em ciências sociais, económicas e comportamentais, tem vindo a aumentar.

Assim sendo, a utilização de humanos virtuais inteligentes tem vindo a tornar-se habitual em diversos contextos em que a capacidade de transmitir emoções é de fundamental importância para tornar credível o comportamento de um humano virtual.

As emoções podem ser expressas através da expressão corporal e, sobretudo, através da expressão facial. Quer uma, quer outra, constituem verdadeiros desafios na modelação de humanos virtuais.

1.2 Objectivos

No seu estado actual de desenvolvimento, a plataforma *IViHumans* tem vindo a usar apenas dois modelos de humanos virtuais, dispondo apenas um deles de expressões faciais [11].

O objectivo deste trabalho foi, produzir mais modelos de humanos virtuais que pudessem ser inseridos na plataforma *IViHumans*. Estes modelos devem incluir animações que permitam tornar as simulações mais credíveis, como andar, correr, dançar, entre outras. Além disto, estes modelos devem incluir animações faciais capazes de transmitir estados emocionais e traços da personalidade. Procura-se que as ditas expressões faciais possam ser aplicadas a qualquer um dos personagens virtuais, conseguindo assim que personagens com faces diferentes tenham a mesma expressão facial. Em relação à deslocação dos humanos, qualquer animação corporal, como andar, correr ou dançar também deve poder ser aplicada aos diferentes humanos virtuais.

Com o decorrer do trabalho, e uma vez que se estava a fazer muito trabalho repetitivo com os modelos virtuais, tornou-se importante desenvolver uma interface visual em Maxscript, a linguagem de scripting interna do 3ds Max, que aumentasse a velocidade do processo de trabalho. Assim sendo implementou-se uma interface a partir da qual o animador pode, preparar vários modelos para serem animados, criar as ditas animações faciais e corporais, para de seguida se quiser poder inseri-las na plataforma *IViHumans*, com mais facilidade e rapidez.

1.3 Contexto e Contribuições do Trabalho

Este trabalho foi realizado no contexto do Projecto em Engenharia Informática, para o Mestrado em Engenharia Informática, na Faculdade de Ciências da Universidade de Lisboa, num dos grupos de investigação do Departamento de Informática, LabMAg - Laboratório de Modelação de Agentes [12].

Este trabalho visa facilitar o processo de trabalho de um animador 3D, aumentar a velocidade do processo de animação dos seus modelos humanos 3D, sempre que este utilizar como ferramenta de modelação o 3D Studio Max.

1.4 Estrutura do documento

Este documento está organizado da seguinte forma:

- Capítulo 1 - Introdução

Neste capítulo é apresentada a motivação, os objectivos desta tese e a sua organização.

- Capítulo 2 - Fundamentos e Estado da Arte

Neste capítulo são apresentados alguns fundamentos teóricos sobre animação, é feita uma breve descrição das várias fases que constituem uma animação 3D. São descritos também dois modelos de deformação facial, algumas das técnicas de animação 3D mais usuais, e trabalho relacionado com a animação 3D.

- Capítulo 3 - Plataforma *IViHumans*

Neste capítulo é apresentada a plataforma *IViHumans* na qual este projecto se insere.

- Capítulo 4 - Trabalho Desenvolvido

Neste capítulo é apresentado o trabalho desenvolvido, ou seja, as nossas abordagens para reutilizar animações em diferentes personagens virtuais e a interface visual que construímos, os modelos virtuais e ferramentas utilizadas, e também o ambiente de desenvolvimento.

- Capítulo 5 - Testes Efectuados

Neste capítulo são apresentados os vários testes efectuados utilizando uma ferramenta de modelação 3D, os resultados obtidos, e a comparação de expressões faciais entre os diferentes modelos virtuais utilizados.

- Capítulo 6 - Conclusão e Trabalho futuro

Neste capítulo são apresentadas as conclusões sobre o trabalho realizado e propõem-se etapas para o trabalho futuro.

- Apêndice A - Manual de Utilizador

Neste apêndice é apresentado o manual de utilizador da interface desenvolvida, as regras de instalação da ferramenta, assim como uma descrição pormenorizada de todas as suas funcionalidades.

- Apêndice B - Ficheiros de Configuração

Neste apêndice são apresentados os ficheiros necessários para inserir humanos virtuais da plataforma *IViHumans*.

- Apêndice C - Testes Efectuados no Blender

Neste apêndice são descritos os testes iniciais, realizados com o Blender como ferramenta de modelação 3D.

Capítulo 2

Fundamentos e Estado da Arte

Este capítulo apresenta alguns fundamentos teóricos sobre animação, com algum ênfase na animação facial. Faz uma breve descrição das várias fases que constituem uma animação 3D. Descreve também dois modelos de deformação facial, algumas das técnicas de animação 3D mais usuais, e trabalho relacionado com a animação 3D.

2.1 Animação

A palavra animação refere-se ao processo segundo o qual, cada fotograma de um filme é produzido individualmente. Pode ser gerado quer por computação gráfica, quer fotografando uma imagem desenhada, quer repetidamente fazendo-se pequenas mudanças a um modelo (*stop motion*) e fotografando o resultado. Quando os fotogramas são ligados entre si, e o filme resultante é visto a uma velocidade de 16 ou mais imagens por segundo, há uma ilusão de movimento contínuo devido à persistência de visão. A construção de um filme torna-se assim um trabalho muito intensivo e por vezes entediante.

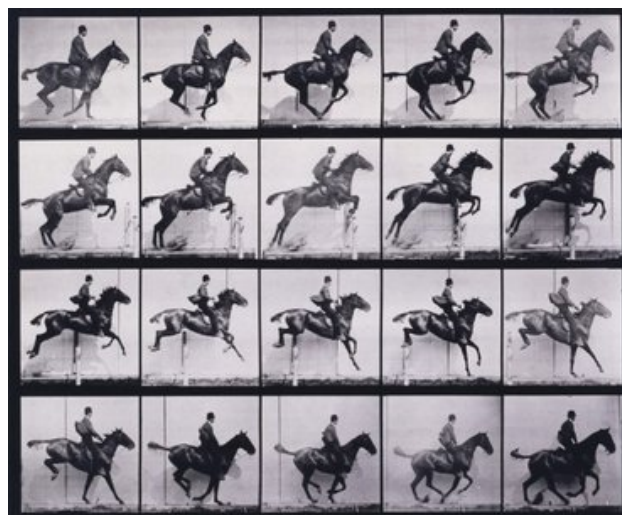


Figura 2.1: Animação em *Stop Motion* [1].

2.2 Animação Facial

A Animação Facial é uma área da computação gráfica que possui vários modelos e técnicas para gerar e animar faces e cabeças de personagens virtuais.

O psicólogo Paul Ekman afirma que o ser humano pode assumir seis expressões faciais primárias, universalmente reconhecidas: raiva, medo, nojo, surpresa, felicidade e tristeza (Figura 2.2) [13].



Figura 2.2: As Seis Expressões Faciais Primárias [2].

As expressões faciais humanas têm sido objecto de investigação científica há mais de 100 anos. Uma das mais importantes tentativas de descrever os movimentos faciais foi a *Facial Action Coding System (FACS)*, que foi introduzido por Ekman e Friesen em 1978 [14], e que descreveremos mais à frente na secção 2.4.2. Apesar de actualmente existirem psicólogos que contradizem esta abordagem, ela continua a ser a mais utilizada na investigação e desenvolvimento de agentes inteligentes sociais, bem como na animação de faces virtuais [13].

2.3 Fases de uma Animação 3D

Para construir uma animação 3D de um personagem virtual, são necessárias essencialmente duas coisas, uma malha poligonal que é a representação do corpo do personagem, e um esqueleto que irá estar associado à sua malha poligonal e que vai permitir ao animador, animar o dito personagem. É claro que depois de feita a animação, o esqueleto estará escondido por baixo da malha poligonal e não será visível no decorrer da animação.

O esqueleto mais utilizado é o *Biped*. O *Biped* é uma figura de duas pernas, humano ou animal, com uma armadura criada especialmente para animação. Os seus ossos estão organizados numa hierarquia, e logo este esqueleto tem propriedades específicas que o tornam instantaneamente pronto para ser animado. Como os humanos, os *Biped's*, são criados para andarem erectos, mas também podem ser usados para criar criaturas quadrúpedes (Figura 2.3).

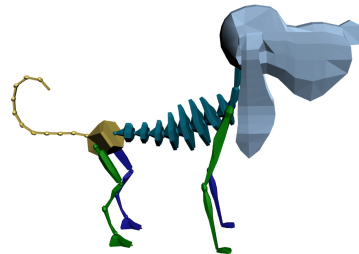


Figura 2.3: Esqueleto *Biped* de um cão [3].

Rigging

Esta é a primeira fase de uma animação 3D, e consiste em construir um *Skeleton* ou *Rig*, que nos permite controlar o personagem. Nesta fase, ajusta-se cada osso do esqueleto de forma a encaixar de forma correcta na malha poligonal do personagem virtual (Figura 2.4).

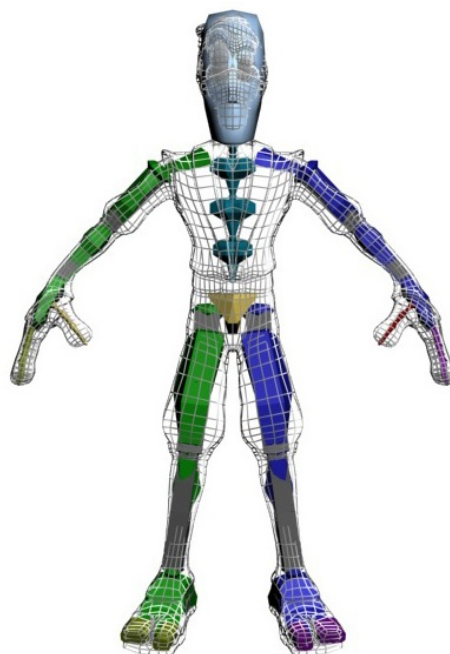


Figura 2.4: Esqueleto *Biped* ajustado à malha poligonal do personagem virtual [4].

Skinning

Consiste em aplicar uma ferramenta de deformação ao esqueleto (ex: *modifier skin*), esta funcionalidade permite-nos deformar um objecto usando um esqueleto. Depois é necessário adicionar-lhe os ossos que se pretendem animar, ou seja, todos os ossos do esqueleto *Biped*. A cada osso é associado automaticamente um *envelope*, um *envelope* é um volume em forma de cápsula, e todos os vértices que lhe são interiores são movidos com o respectivo osso.

Weighting

Consiste em atribuir a cada um dos vértices do modelo, um peso que é relativo ao osso a que está associado, a posição final do vértice é calculada consoante esse peso, um maior peso equivale a um movimento maior (Figura 2.5(a)). Mas, no entanto, existe outra maneira de fazer o *Weighting* do modelo, que é posicionar os *envelopes* de tal forma, que os vértices que lá estão dentro são aqueles que desejamos mover, quando alterarmos a posição do osso (Figura 2.5(b)).

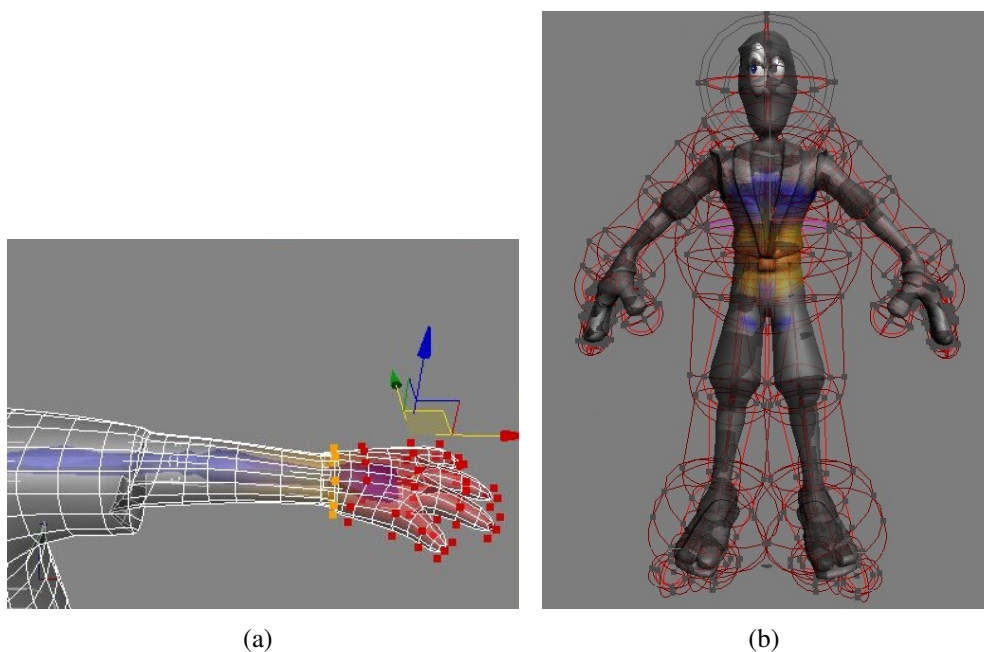


Figura 2.5: Representação de dois tipos de *Weighting* [5] [6].

Animation

Depois de passarmos por todas as fases anteriores o modelo está pronto para ser animado. A animação propriamente dita pode ser construída a partir de várias técnicas, as mais usuais estão explicadas na secção 2.5.

2.4 Modelos de Deformação Facial

Existem diversos modelos de deformação facial, que têm vindo a ser desenvolvidos ao longo dos anos. Muitos destes modelos são apenas adaptações de modelos generalistas de deformação facial, alguns baseiam-se meramente na criação de efeitos visuais credíveis, sem terem em consideração leis estruturais e físicas. E outros são derivados de princípios físicos relacionados com forças, contracções musculares e biomecânica associada à pele.

Faremos aqui uma breve descrição de dois modelos de deformação facial, são eles o MPEG-4 e o FACS.

2.4.1 *Moving Picture Experts Group Layer 4 (MPEG-4)*

Um dos modelos de deformação facial mais usado baseia-se na norma MPEG-4 [15], na qual se especifica:

- O modelo facial em estado neutro;
- Um conjunto de pontos de referência;
- Um conjunto de parâmetros de animação facial: os *Facial Animation Parameters (FAP)*.

A cada um dos *FAP* corresponde uma acção facial específica que modifica a face no estado neutro.

A norma *MPEG-4* especifica 84 atributos ou pontos de referência para o estado neutro. Estes estão organizados em pequenos grupos como o queixo, os olhos, a boca, nariz entre outros (Figura 2.6).

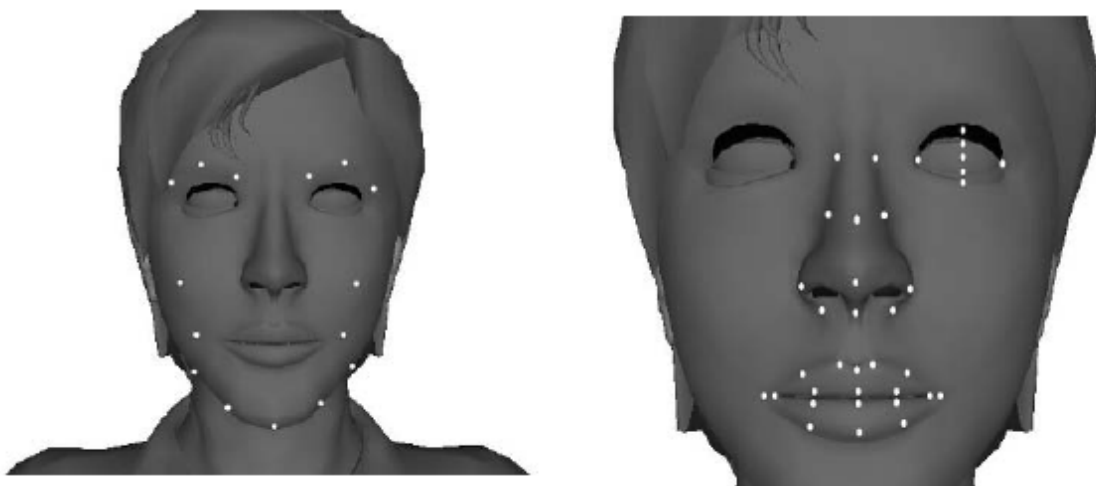


Figura 2.6: Pontos de referência representados na face de um modelo.

2.4.2 Facial Action Coding System (FACS)

FACS é um sistema muito útil para medir qualquer expressão facial que um humano possa fazer. É baseado na anatomia humana e descreve exaustivamente todos os movimentos faciais observados. Cada componente observável de um movimento facial é chamado de AU (*Action Unit*), e todas as expressões faciais podem ser decompostas nos AU's que as constituem (Figura 2.7). Este sistema, inicialmente desenvolvido por Friesen, classifica faces e acções faciais através de descrições musculares. Apesar de não descrever a que tipo de emoção uma face corresponde, este sistema veio representar através da descrição dos músculos, as expressões faciais comumente aceites como as que correspondem às seis emoções básicas, propostas pelos psicólogos Ekman e Friesen [16].

Através deste código qualquer expressão facial resulta de uma combinação dos 268 músculos da cara. Esta propostas de Ekman e Friesen define 44 tipos de acções básicas - *Action Units (AU's)* - em que cada acção é afectada por um conjunto de músculos que não são controlados conscientemente.

Upper Face Action Units					
AU 1	AU 2	AU 4	AU 5	AU 6	AU 7
*AU 41	*AU 42	*AU 43	AU 44	AU 45	AU 46
Lower Face Action Units					
AU 9	AU 10	AU 11	AU 12	AU 13	AU 14
AU 15	AU 16	AU 17	AU 18	AU 20	AU 22
AU 23	AU 24	*AU 25	*AU 26	*AU 27	AU 28

Figura 2.7: *Action Units* e respectivas interpretações [7].

2.5 Técnicas de Animação 3D

2.5.1 *Motion Capture (Mocap)*

É uma técnica que consiste em posicionar câmaras à volta de um actor, esta utiliza pequenos reflectores que são aplicados na face, ou então num fato completo, chamados marcadores (ou *markers*) que determinam precisamente a posição da pessoa no espaço (Figura 2.8(a)). Os movimentos dos marcadores são guardados pelas câmaras e são depois mapeados num modelo tridimensional do actor, de tal forma que o modelo executa o mesmo movimento do actor. Até há algum tempo atrás, o tamanho destes marcadores usados pelos sistemas de *motion capture* tornavam-nos inapropriados para captura facial. No entanto, hoje em dia já são bastante pequenos, o que faz desta técnica a ferramenta mais viável tanto para animação facial como corporal (Figura 2.8(a)).

No que toca a efeitos especiais não existe nada melhor que o *motion capture*. Esta técnica é bastante vantajosa pelo facto de as animações terem um aspecto bastante realista, uma vez que são feitas a partir de acções de humanos. Foi usada consideravelmente no filme *Avatar*, pela *Lightstorm Entertainment*, e foi uma inovação em termos de tecnologia cinematográfica devido ao seu desenvolvimento com visualização 3D e gravação com câmaras que foram feitas especialmente para a produção do filme (Figura 2.8(b)).

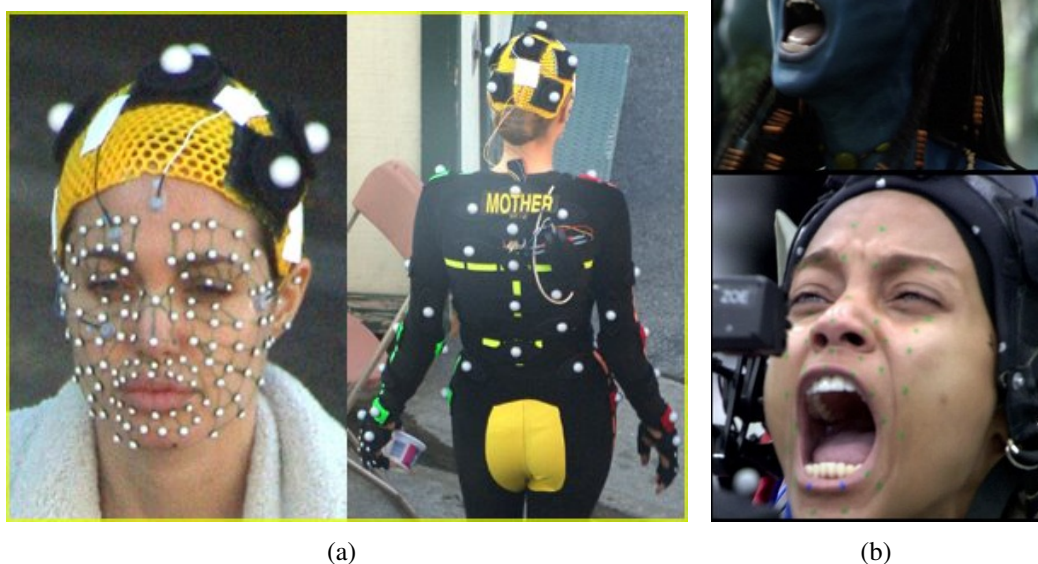


Figura 2.8: *Motion Capture* [8] [9].

2.5.2 *Morph Target Animation*

Morph Target Animation (ou *Per-Vertex Animation*) é outra técnica de animação 3D muito utilizada (Figura 2.9).

Nesta técnica a animação é guardada na forma de conjuntos de posições de vértices, e em cada *keyframe* os vértices são movidos para uma posição diferente, construindo desta forma uma animação.

Existem vantagens de usar esta técnica, uma delas é que o artista tem mais controlo sobre os movimentos, pois consegue definir posições individuais dos vértices numa determinada *keyframe*. Mas por outro lado é muito lenta, porque um modelo 3D tem em média cerca de 6000 a 15000 vértices, e cada posição de cada vértice tem que ser calculada.

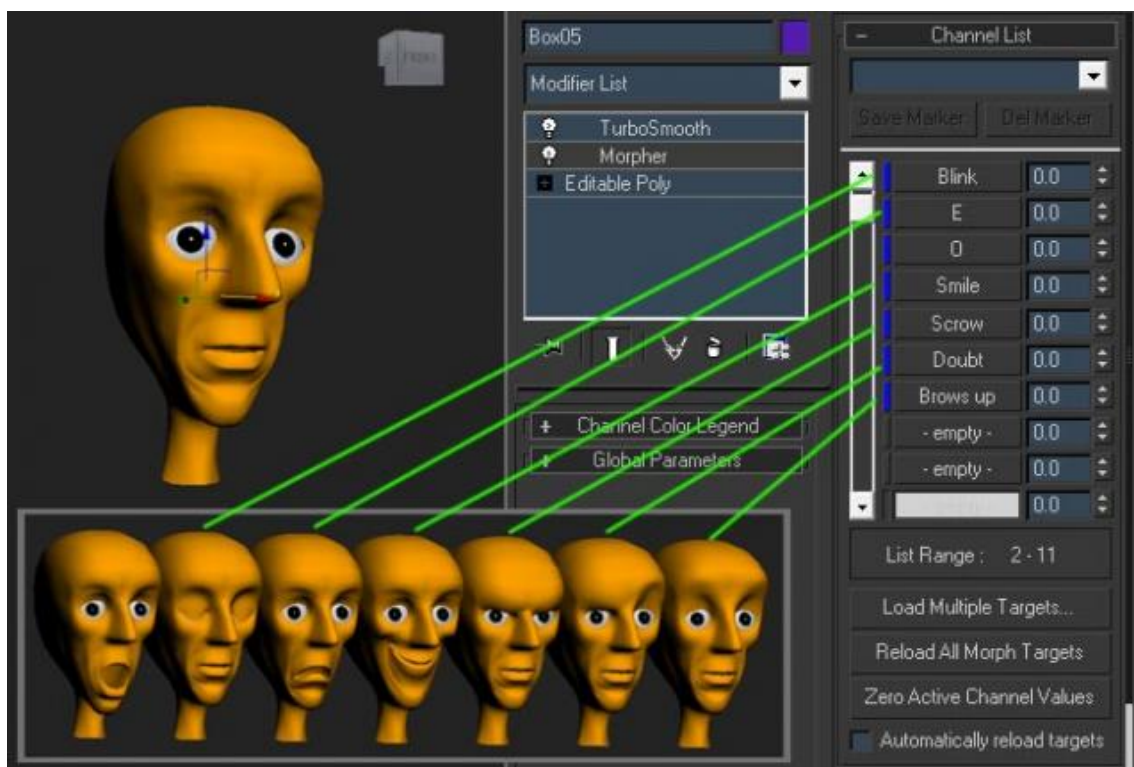


Figura 2.9: Utilização de *Morph Target Animation* no *3D Studio Max* [10].

2.5.3 *Skeleton Animation*

Nesta técnica de animação, um personagem é representado em duas partes, são elas a representação da superfície do modelo, chamada malha poligonal (*mesh*), e um conjunto de ossos usados para animar o dito modelo, chamado esqueleto (*skeleton*) (Figura 2.10). Esta técnica é usada construindo um conjunto de ossos (*rig*) que vão movimentar o modelo, a este processo chama-se *Rigging*. Estes ossos estão organizados numa hierarquia, isto quer dizer que ao movermos por exemplo a parte superior da perna, a parte inferior também se vai mover. Cada osso do *skeleton* é associado a um grupo de vértices da *skin*, a este processo de criar esta associação chama-se *Skinning*. A cada um destes vértices é atribuído um peso relativamente ao osso a que está associado, a posição final do vértice é calculada consoante esse peso, um maior peso equivale a um movimento maior. A esta fase chama-se *Weighting*, como vimos anteriormente.

Esta foi a técnica que escolhemos para utilizar neste projecto, e a vantagem desta técnica é o facto de se conseguir definir uma animação, com um simples movimento dos ossos do esqueleto, ao contrário da técnica anterior 2.5.2 que tem que ser feita vértice a vértice. A desvantagem é que esta técnica não fornece um movimento realista de músculos, e utiliza apenas os ossos.



Figura 2.10: Representação da malha poligonal e do esqueleto no 3ds Max.

2.6 Trabalho Relacionado com a Animação 3D

Identificamos aqui problemas e as respectivas soluções de alguns artigos relacionados com cada uma das técnicas de animação 3D descritas anteriormente.

2.6.1 *Motion capture*

Em 2005, Sifakis et al. [17] construíram um modelo anatomicamente correcto da musculatura facial, tecido e estrutura óssea subjacente, usando dados volumétricos adquiridos a partir de um sujeito vivo do sexo masculino. Modelos detalhados deste género podem ser difíceis de animar, e exigem um estímulo complexo e coordenado da musculatura subjacente. A solução que propuseram para este problema consistia em determinar automaticamente as activações musculares que controlavam um conjunto de marcadores da superfície, adquiridas a partir de dados de *motion capture*. Como a animação resultante é obtida através de um método de elementos finitos tridimensional não-linear, obtêm-se deformações anatomicamente correctas, com coerência espacial e temporal, o que proporciona robustez contra *outliers* nos dados de *motion capture*. Além disso, as activações musculares obtidas podem ser utilizadas numa plataforma de simulação, incluindo contacto e colisões da face com objectos externos. Na (Figura 2.11) podemos ver uma expressão facial criada pela acção de 32 músculos, simulados por uma malha tetraédrica de elementos finitos.

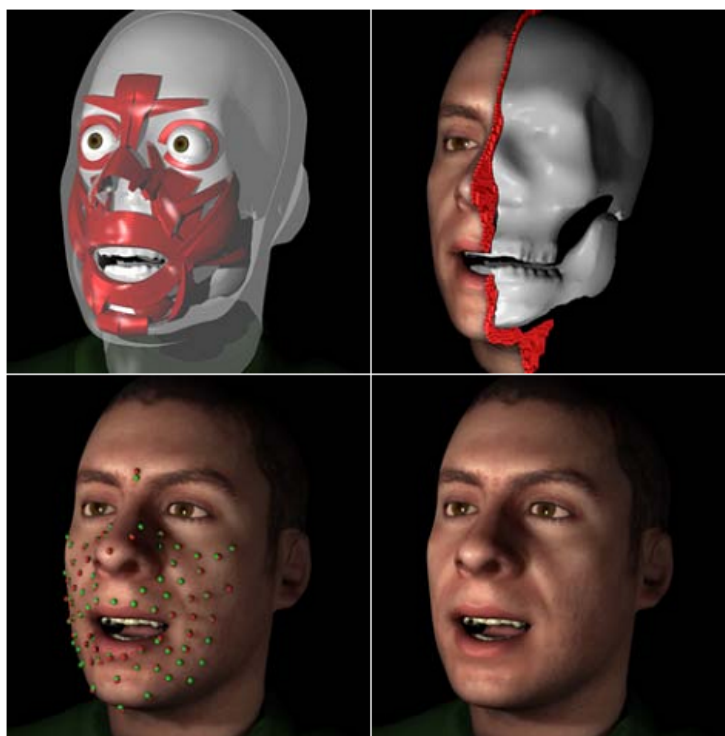


Figura 2.11: Expressão facial final do modelo.

Os gestos faciais humanos apresentam frequentemente variações estocásticas, como por exemplo a frequência do piscar dos olhos, quantas vezes as sobrancelhas e o nariz contraem e como a cabeça se movimenta durante a fala. Os movimentos estocásticos das características faciais são ingredientes cruciais para gerar expressões faciais convincentes e realistas. Embora tais pequenas variações tenham vindo a ser simuladas usando funções de ruído em muitas aplicações gráficas, modular as ditas funções de ruído para coincidir com variações naturais, induzidas por estados afectivos e personalidade dos personagens é difícil e não é intuitivo.

Em 2008, Ju e Lee [18] apresentaram uma técnica semi-automática para gerar gestos faciais expressivos (expressões faciais e movimentos de cabeça) a partir dos dados de *motion capture* (Figura 2.12). Esta abordagem é baseada em campos aleatórios de Markov, que são simulados em dois níveis. No nível mais baixo, os movimentos coordenados das características faciais são capturados, parametrizados e depois transferidos para faces sintéticas usando formas base. O nível superior representa comportamentos estocásticos independentes das características faciais. Os resultados experimentais mostram que este sistema gera gestos faciais expressivos sincronizados com o discurso de entrada.

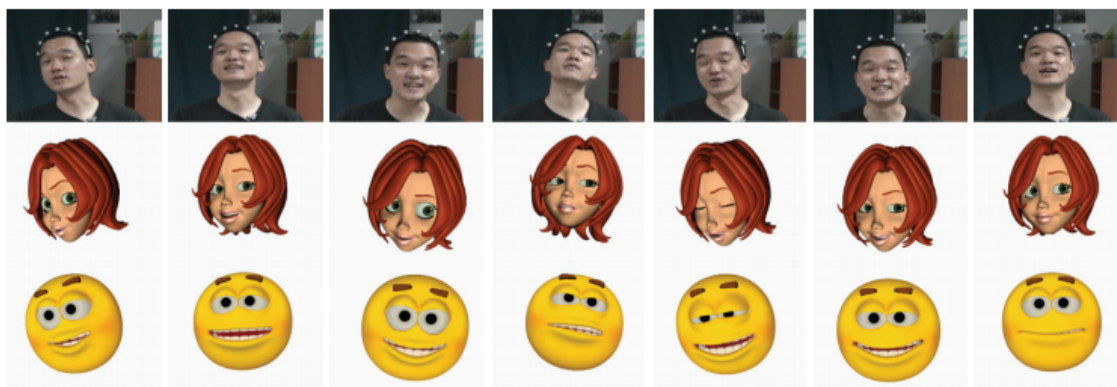


Figura 2.12: Transferência de expressões faciais.

2.6.2 *Morph Target Animation*

Em 2003, Bui et al. [19] apresentaram um novo método que localizava automaticamente um conjunto de treino de redes *RBF* (*Radial Basis Functions*) para fazer o *morphing* a uma face protótipo, para de seguida representar uma nova face (Figura 2.13). Isto era feito automaticamente, especificando e ajustando os pontos característicos correspondentes numa face alvo, como o nariz, orelhas, olhos e etc. As redes RBF eram então usadas para transferir os músculos da face protótipo para a *morphed face*. O ajuste automático destes pontos característicos na face alvo era feito através de algoritmos genéticos. A função de ajuste utilizada nos algoritmos genéticos, expressa a diferença entre a superfície da *morphed face* e da face alvo.

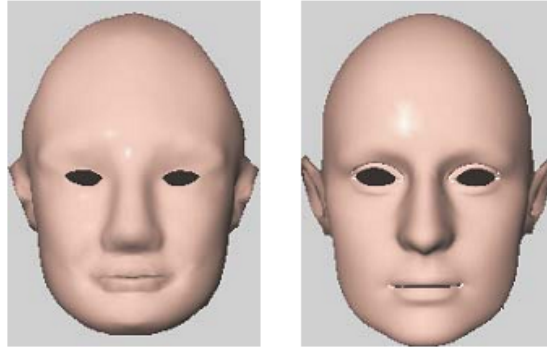


Figura 2.13: *Morphed face* e face alvo respectiva.

Em 2007, Cosker et al. [20] descrevem um método de re-mapeamento de parâmetros de animação, entre vários tipos de modelos faciais, para desempenho orientado à animação. A performance facial pode ser analisada automaticamente, em termos de um conjunto de trajetórias de acção facial, utilizando um modelo de aparência modificado com modos de variação, que codificam acções faciais específicas. Estes parâmetros podem ser usados para animar outros modelos de aparência, ou modelos faciais 3D. Assim, os parâmetros de animação analisados a partir do desempenho de vídeo podem ser reutilizados, para animar vários tipos de modelos faciais (Figura 2.14).



Figura 2.14: Mapeamento de duas expressões faciais.

2.6.3 *Skeleton Animation*

Animar um personagem 3D articulado requer que seja feito um *Rigging* manual, para se poder especificar a estrutura interna do esqueleto, e definir como é que os movimentos vão deformar a superfície do personagem. Em 2007, Baran e Popovic [21] apresentaram um método que lhes permitia animar um personagem automaticamente. Dada uma malha poligonal do personagem e um esqueleto genérico, o seu método adaptava o esqueleto à malha poligonal e ligava-o à superfície do personagem, permitindo animá-lo com dados de movimento do esqueleto (Figura 2.15). Como um único esqueleto podia ser usado para uma ampla gama de personagens, o seu método juntamente com uma biblioteca de movimentos para o esqueleto, permitia um sistema de animação *user-friendly* para iniciantes. O seu protótipo chamava-se *Pinocchio*, e demorava cerca de um minuto a fazer o *rig* de um personagem num pc de gama média.

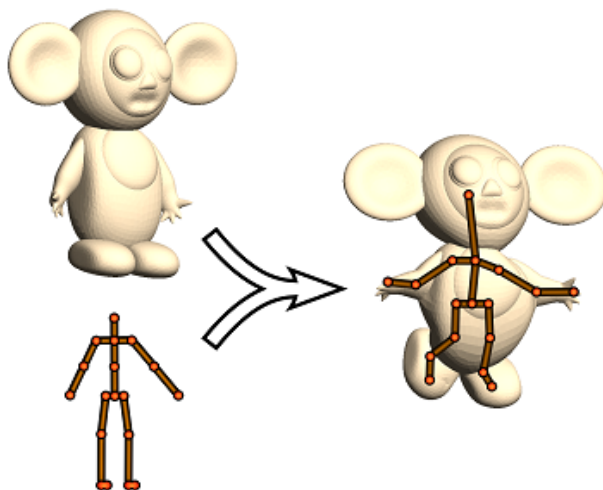


Figura 2.15: Representação do *Pinocchio*.

Também em 2007, Forstmann e Ohya et al. [22], apresentaram um novo sistema de *Skeleton Animation* baseado em deformações de *splines* para fornecer deformações de alta qualidade em tempo real.

O objectivo era permitir aos artistas a fácil criação de comportamentos de deformação, que pudessem ser atribuídos directamente a uma grande variedade de objectos alvo em simultâneo. Para atingir este objectivo, introduziram o uso de estilos de deformação e demonstraram a sua aplicabilidade no seu novo sistema de *Skeleton Animation*. Portanto, melhoraram a *Spline-Skinned Skeleton Animation* com duas variantes de deformação, livres de forma e baseadas em varrimento. As duas variantes são dependentes da postura, e conduzidas por três texturas e três curvas, que podem ser concebidas pelo artista.

Os estilos de deformação uma vez concebidos poderiam ser directamente aplicados a qualquer número de objectos alvo, para imitar o comportamento de materiais como roupa, metal ou mesmo músculos (Figura 2.16).

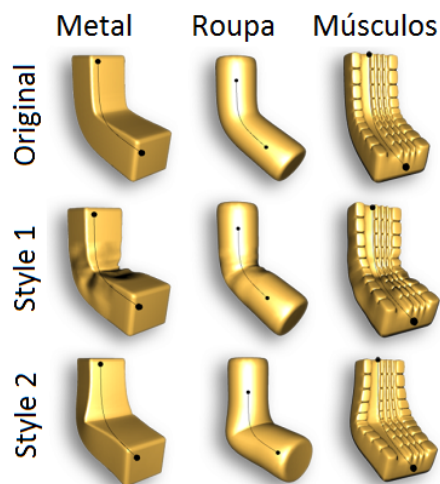


Figura 2.16: Representação da deformação dos diferentes objectos alvo.

Em 2008, tornava-se cada vez mais popular representar animações não por meio de um modelo clássico baseado no esqueleto, mas sim sob a forma de sequências de deformação da malha poligonal. A razão para esta tendência era que os métodos de deformação da malha poligonal dessa altura, bem como as técnicas de captura de superfície baseada na cena, ofereciam uma grande flexibilidade durante a criação da animação. Infelizmente, a representação da cena resultante é menos compacta do que as que usavam o esqueleto, e ainda não havia um conjunto de ferramentas disponível que permitisse um fácil pós processamento e modificação das *mesh animations*.

Para colmatar esta lacuna, Aguiar et al. [23], propunham um novo método que extraía automaticamente um esqueleto plausível, os parâmetros de movimento do esqueleto, bem como os *skinning weights* da superfície, de *mesh animations* arbitrárias. Deste modo, deformando sequências da malha poligonal, obtinham-se automaticamente sujeitos virtuais completos. De seguida, o *input* original seria rapidamente renderizado, com base nas novas representações dos ossos e malha poligonal do sujeito virtual, e facilmente modificado usando as ferramentas de animação já existentes (Figura 2.17).



Figura 2.17: Vista geral do funcionamento do algoritmo.

2.7 Conclusão

Neste capítulo apresentámos alguns fundamentos teóricos sobre animação e animação facial, uma breve descrição das várias fases que constituem uma animação 3D, ou seja, o *Rigging*, o *Skinning* e o *Weighting*. Descrevemos também dois modelos de deformação facial, o MPEG-4 e o FACS. Descrevemos também algumas das técnicas de animação 3D mais usuais, o *Motion Capture*, a *Morph Target Animation* e a *Skeleton Animation*. E apresentámos vários artigos relacionados com a matéria em estudo, a animação 3D.

Aprendemos que, representar emoções humanas em personagens virtuais é uma área de investigação que está em constante evolução e aperfeiçoamento. Cada vez se encontram mais métodos e técnicas para o fazer, de forma que os humanos virtuais sejam representados da forma mais realista possível. Esta representação vai sempre depender bastante da qualidade da modelação do humano virtual em si, e também da qualidade da técnica escolhida para o animar.

Capítulo 3

Plataforma *IViHumans*

Este capítulo descreve a plataforma *IViHumans*, a sua arquitectura e o modo como se tratam a percepção, o movimento e as expressões faciais.

3.1 Introdução

A plataforma *IViHumans* é destinada ao suporte de aplicações com humanos virtuais inteligentes. Foi desenvolvida em C++, recorrendo ao Microsoft Visual Studio, e possui uma camada para o processamento gráfico e outra para a inteligência artificial. As duas camadas comunicam entre si a partir de um protocolo Cliente-Servidor: a camada de processamento gráfico é o servidor e a camada de Inteligência Artificial é o cliente. Deste modo a camada do processamento gráfico é a base da plataforma, e fornece serviços aos agentes inteligentes que povoam a camada da inteligência artificial [11].

3.2 Arquitectura

Na plataforma *IViHumans* existe um balanço uniforme de responsabilidade entre as duas camadas, a camada de processamento gráfico (*GP - Graphical Processing*) é responsável por controlar rapidamente os aspectos de baixo nível, que caracterizam o processamento gráfico e físico, fornecendo uma interface abstracta à camada de Inteligência Artificial (*AI - Artificial Intelligence*) (Figura 3.1). A camada AI, por sua vez, lida com comportamentos cognitivos mais complexos usando representações simbólicas.

A camada GP usa como motor de *rendering* o *OGRE*, e o sistema Multi-Agente que é o *core* da camada AI, foi desenvolvido sobre a *Framework JADE*.

A principal tarefa da camada GP é representar todos os elementos contidos no mundo virtual, onde os humanos virtuais assumem o papel mais importante, reproduzindo as animações apropriadas que carregam o fluxo de ocorrências e consistentemente definem a evolução do mundo e seus componentes.

Por sua vez a camada AI apresenta e controla as mentes dos humanos virtuais, de forma a cada um ser controlado por um ou mais agentes que lhes conferem comportamento inteligente. Na camada AI existem essencialmente dois tipos de agentes, agentes de interface e agentes cognitivos. Os agentes de interface lidam com a comunicação, transformam dados em representações simbólicas significativas, e acções de translação de alto nível em comando apropriados. Componentes de interface estão também presentes na camada GP (Figura 3.1).

Em relação à comunicação é usado um protocolo TCP bastante simples, a camada GP fornece o servidor, e a camada AI liga-se a ela através de um cliente. Este protocolo é extensível pois novas mensagens podem ser adicionadas por qualquer aplicação que use a plataforma.

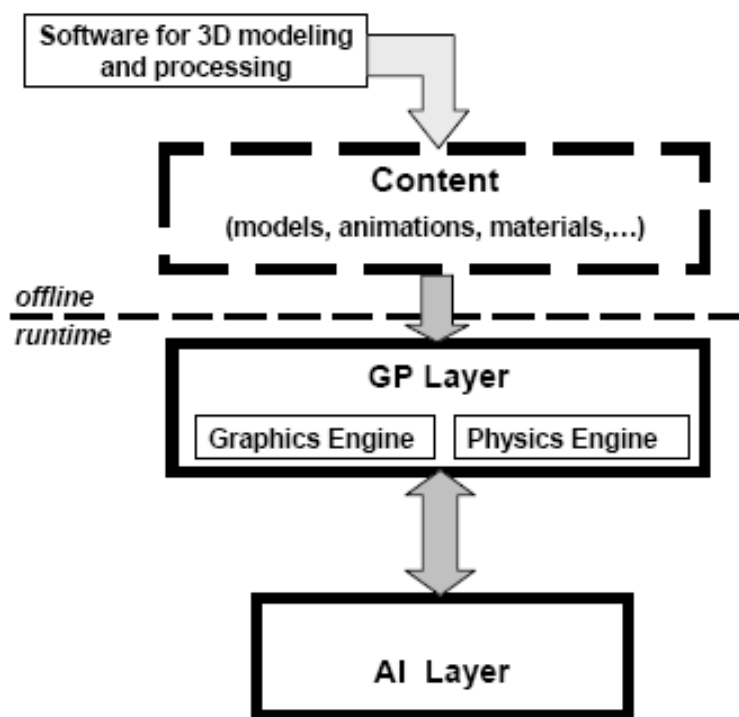


Figura 3.1: Arquitectura da plataforma *IViHumans*.

3.3 Percepção, Movimento e Expressões Faciais

3.3.1 Percepção

Na plataforma *IViHumans*, um humano virtual percebe o seu ambiente através de um algoritmo sintético de visão do tipo *ray-casting* [24] [25]. Este algoritmo permite a parametrização de diversas características, o que lhe dá bastante flexibilidade. Cada humano virtual tem uma câmara que herda o movimento da sua cabeça, esta é usada para mostrar o que o humano vê, e como referência para derivar a direcção dos raios que são lançados na cena.

A classe *SyntheticRayVision* implementa o algoritmo da visão, que pode fornecer informações sobre os objectos que o humano virtual vê num determinado momento. Num humano virtual (instância da classe *IViHuman*) pode também ser activado o modo de "visão automática", que consiste em executar periodicamente o seu algoritmo de visão. Para que este modo de visão funcione correctamente, o humano tem que ter memória, para gravar dados relevantes e fornecê-los quando necessário. A classe *VisionMemory* simula a memória do humano (Figura 3.2).

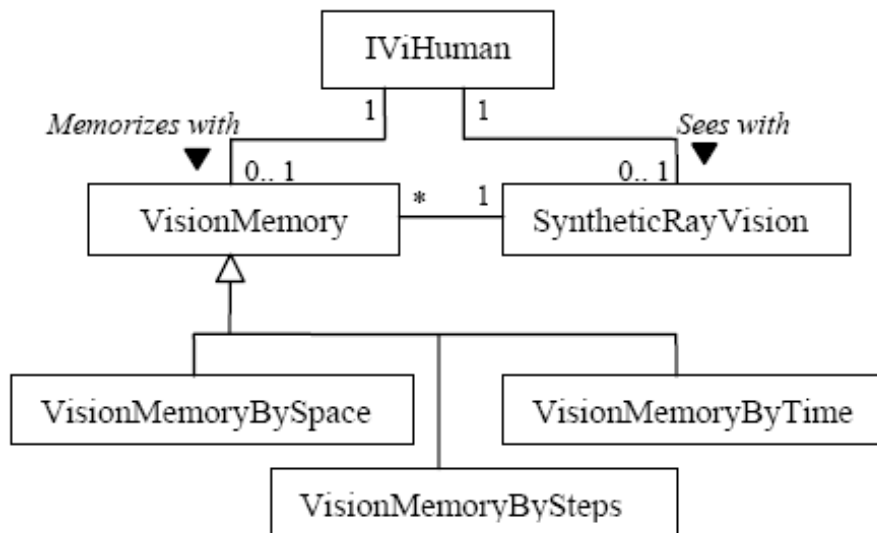


Figura 3.2: Diagrama de classes - Visão e Memória.

Existem também três subclasses da classe abstracta *VisionMemory*, são elas: a *VisionMemoryBySpace* que guarda informação de um determinado número de objectos, a *VisionMemoryByTime* que guarda toda a informação obtida num determinado espaço de tempo e a *VisionMemoryBySteps* que guarda toda a informação obtida num determinado número de iterações de visão.

3.3.2 Movimento

A deslocação de um humano virtual é suportada pela noção de *Steering Behaviour*. Foi adoptada a categorização hierárquica de movimento de Craig Reynold's nas três camadas de locomoção, direcção e selecção de acção. Alguns *Steering Behaviours* já estão incluídos, mas muitos mais podem ser implementados.

Os atributos mais óbvios que um humano virtual deve ter, são a posição, velocidade e a orientação, os valores destes atributos podem ser obtidos aplicando as leis da Física clássica. Usando estas leis um humano virtual pode actualizar o seu estado físico dado o tempo decorrido desde o último estado.

A classe *IViEntity* (Figura 3.3) implementa apenas uma representação abstracta de movimento, serve como classe base e fornece funcionalidades de movimento básico a outras classes.

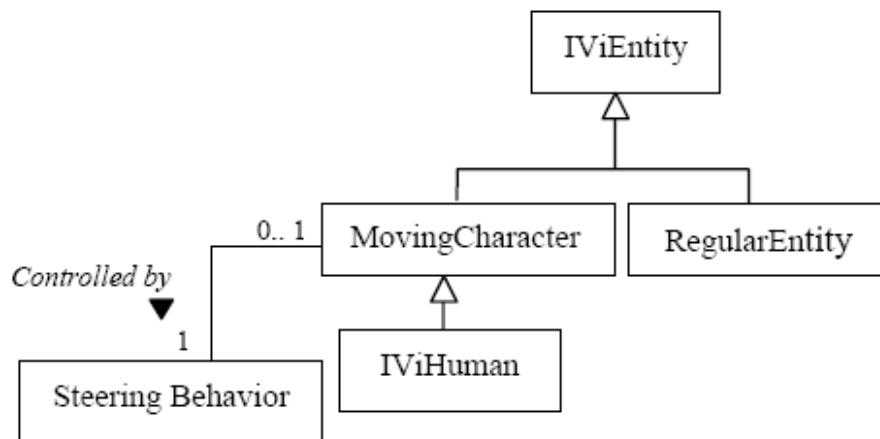


Figura 3.3: Diagrama de classes - *IViEntity*.

Na plataforma *IViHumans*, um *MovingCharacter* pode ter uma instância da classe *SteeringBehavior* a gerir o seu movimento, e este comportamento pode ser básico ou então um *CombineBehavior*, que é na verdade um comportamento composto [26].

3.3.3 Expressões Faciais

Na plataforma *IViHumans*, os humanos virtuais transmitem emoções através de expressões faciais. Qualquer número de expressões básicas pode ser modelado por deformações do modelo original, e estas podem ser misturadas para criar expressões compostas.

As expressões básicas são animações de 1 segundo, e são compostas por duas *keyframes*, uma no instante 0s que não altera em nada o modelo 3D, e outra no instante 1s que regista a forma do modelo 3D que corresponde à expressão na sua máxima intensidade. Desta forma, quando se manipula mais do que uma expressão, o humano virtual exhibe uma expressão composta.

3.4 Conclusão

Neste capítulo descrevemos de forma breve, algumas das características e funcionalidades mais importantes da plataforma *IViHumans*. Resumidamente esta plataforma oferece suporte a aplicações com humanos virtuais inteligentes.

Um vez que o trabalho realizado está inserido na plataforma *IViHumans*, achámos que era de todo o interesse darmos a conhecer as propriedades da plataforma, que mais se relacionavam com o trabalho. São elas; a arquitectura, que possui duas camadas, a camada de processamento gráfico (GP) e a camada de inteligência artificial (AI); a percepção, que possui um algoritmo sintético de visão, e uma memória que simula a de um humano; o movimento, que é suportado pela noção de *SteeringBehaviors* de Craig Reynold's; e principalmente as expressões faciais, que é onde este trabalho vai incidir, construindo animações faciais que permitem aos humanos virtuais, transmitir emoções.

Capítulo 4

Trabalho Desenvolvido

Este capítulo descreve o trabalho desenvolvido, ou seja, as nossas abordagens para reutilizar animações em diferentes personagens virtuais e a interface visual que construímos, apresenta os modelos virtuais e ferramentas utilizadas, e também o ambiente em que o trabalho foi desenvolvido.

4.1 Introdução

Na plataforma *IViHumans* a ferramenta de modelação 3D utilizada foi sempre o Blender, um programa de software de domínio público. Por este motivo, esta ferramenta foi a escolhida neste trabalho. Contudo, uma fase de testes iniciais revelou grandes dificuldades, tendo-se obtido resultados insatisfatórios, como descreve o Apêndice C. Deste modo, foi adoptada outra ferramenta de modelação 3D, o 3D Studio Max, um produto comercial que a par com o Maya, é dos melhores programas de modelação 3D que existem actualmente.

Tornou-se então importante aprender a trabalhar com esta nova ferramenta, e explorar as suas funcionalidades. Depois de muitos tutoriais e de já possuir algum domínio nesta ferramenta, comecei a investigar qual seria a melhor forma de conseguir resolver o principal objectivo deste projecto, ou seja, conseguir reutilizar animações iguais em diferentes modelos de humanos virtuais.

Tendo em conta que o trabalho foi desenvolvido no contexto do 3ds Max, optei por indicar ao longo do texto, as funcionalidades e opções desta ferramenta que foram sendo utilizadas

4.2 Abordagens Desenvolvidas

No meu trabalho implementei duas abordagens distintas para transferir animações de um modelo para o outro, cada uma delas com granularidade diferente. O desenvolvimento destas abordagens é descrito em seguida.

Os primeiros testes foram feitos com um esqueleto do tipo *Biped* disponibilizado pelo 3ds Max. Este tipo de esqueleto permite-nos animar o corpo de um humano virtual utilizando os vários controlos do *motion panel* do 3ds Max. No entanto, este esqueleto é limitado no que diz respeito à face: por omissão não contém ossos que permitam animar esta parte do corpo. Comecei por adicionar ossos extra ao esqueleto *Biped* através da componente 'Xtra group' para conseguir assim animar também a face do modelo (Figura 4.1). Estes ossos extra são associados à cabeça do esqueleto usando o comando 'parent'. De notar que estes ossos extra que adicionei à face, não existem no esqueleto de um humano real, são usados apenas com o intuito de animar a face do personagem virtual. Desta forma obtive um novo esqueleto capaz de suportar animação facial e corporal (Figura 4.1). Este novo esqueleto é guardado num ficheiro de extensão *.max* e carregado no 3ds Max sempre que pretendermos animar um modelo humano.

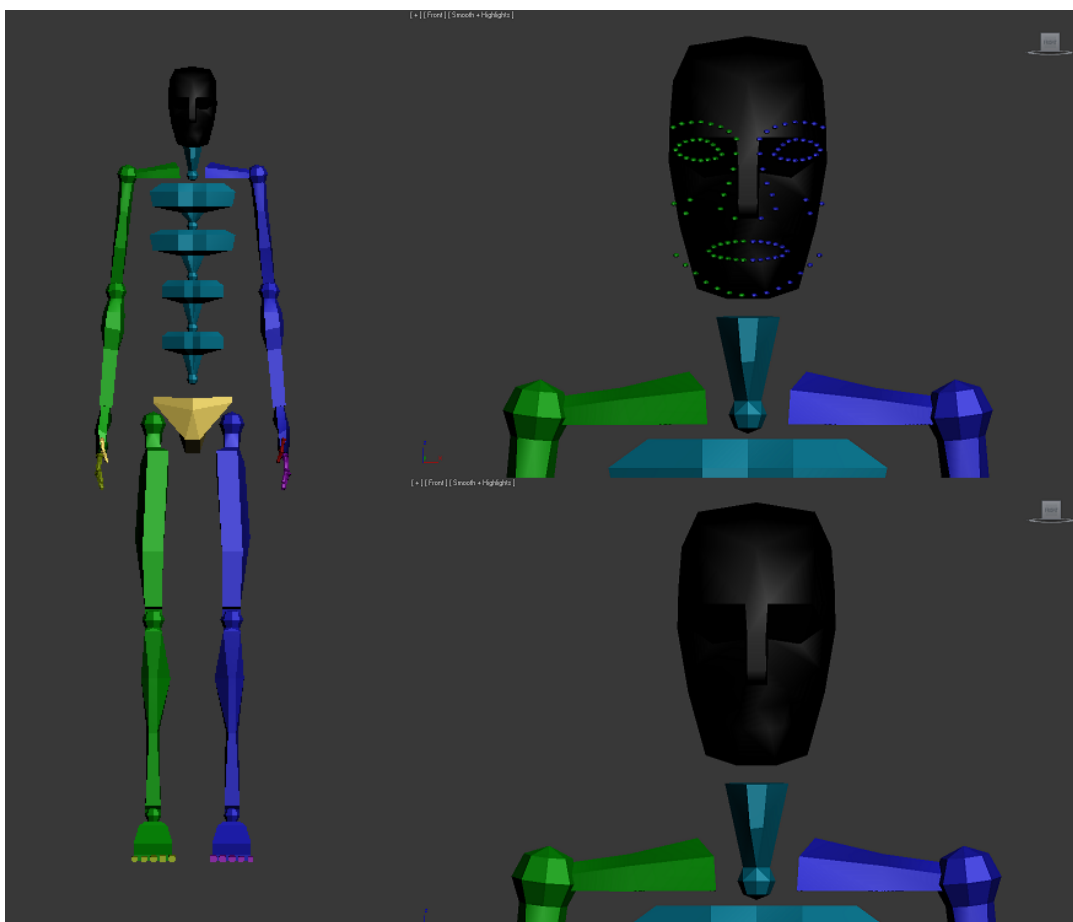


Figura 4.1: Esqueleto *Biped* do 3ds Max com ossos extra na face.

Depois de já ter o esqueleto final que iria ser utilizado em todos os modelos de forma genérica, deparei-me com um problema, mesmo utilizando o mesmo esqueleto em dois modelos diferentes, para que a animação fosse semelhante em ambos, o *Skinning* e o *Weighting* teriam também que ser semelhantes em ambos os modelos.

Surgiu então o seguinte problema:

Como passar o Skinning e o Weighting de um modelo para o outro?

Para resolver este problema usei duas abordagens. A primeira consistiu em utilizar os vértices do modelo para passar o *skinning* e o *weighting* de um modelo para o outro. A segunda abordagem consistiu em utilizar, em vez dos vértices do modelo, os *envelopes* dos ossos do esqueleto. Na secção seguinte apresento as duas abordagens em mais pormenor.

4.2.1 Primeira Abordagem: Usando Vértices

A primeira ideia que tive para abordar este problema foi, depois de ter o *skinning* e o *weighting* feito num modelo, desenvolver um *script* em Maxscript (linguagem de *scripting* interna do 3ds Max) que percorria todos os vértices de um modelo, e guardava o peso atribuído a cada vértice num ficheiro. A ideia base era, para um segundo modelo, ler do ficheiro o valor do peso de cada vértice e, atribuí-lo ao vértice do segundo modelo que tivesse as coordenadas mais próximas do vértice do modelo original. Esta ideia teoricamente fazia algum sentido, mas depois ter desenvolvido o *script* e após alguns testes realizados concluí que, não era a melhor forma de abordar o problema, pois existiam sempre problemas de diferença da escala dos modelos e até diferenças a nível da estrutura facial dos modelos. Outro dos problemas desta abordagem era o facto de ser muito lenta, pois num modelo de 15.000 vértices demorava bastante tempo a calcular qual o vértice com as coordenadas mais próximas e a atribuir-lhe o peso respectivo. A lentidão também se deve ao facto de estarmos a usar uma linguagem interpretada.

4.2.2 Segunda Abordagem: Usando *Envelopes*

A segunda abordagem utiliza os *envelopes* dos ossos. O primeiro passo foi, fazer o *skinning* e o *weighting* a um personagem virtual posicionando os *envelopes* nos sítios correctos. Este trabalho pode ser guardado sobre a forma de um ficheiro de extensão *.env*. Desta forma, ficamos com um ficheiro que contém o *skinning* e o *weighting*, ou seja, as definições dos *envelopes* dos ossos do modelo. Logo, este ficheiro pode, potencialmente ser usado para animar outros personagens virtuais desde que se use o esqueleto que defini. Basta para isso usar uma funcionalidade do 3ds Max chamada *Load Envelopes* (Figura 4.2), que nos permite fazer o *load* do ficheiro de extensão *.env* para um outro modelo, desde que use o mesmo esqueleto.

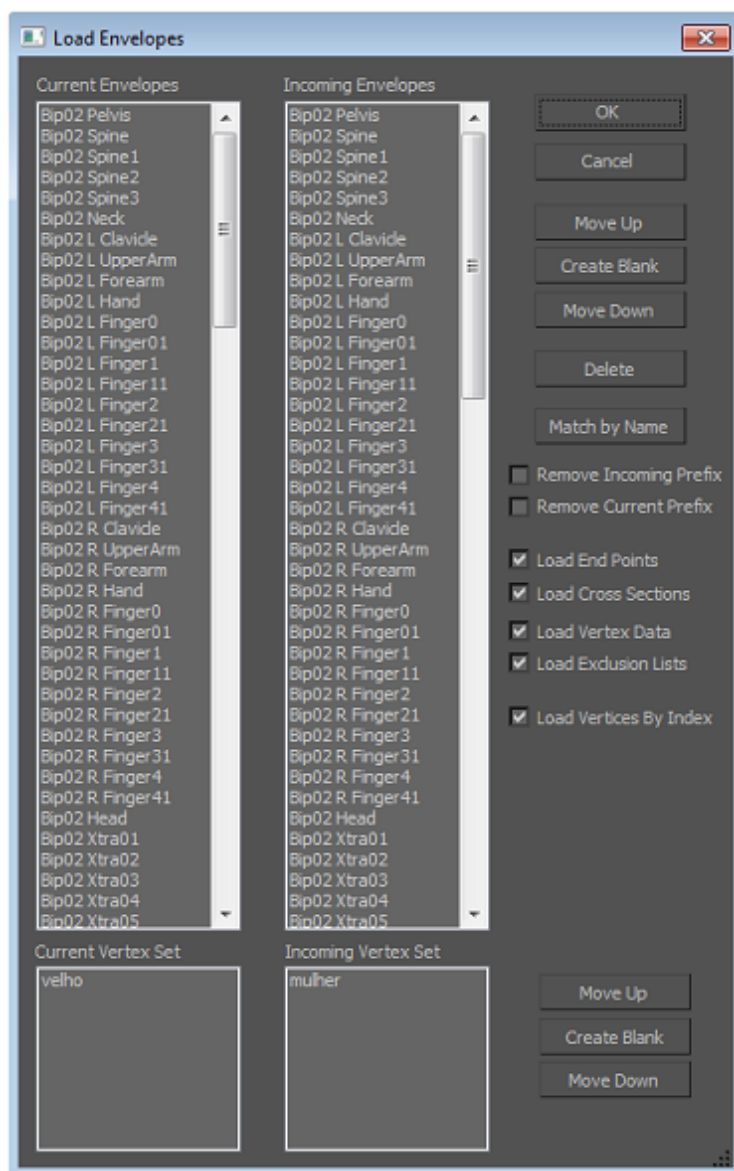


Figura 4.2: Funcionalidade *Load Envelopes* do 3ds Max.

A Figura 4.2, mostra à esquerda uma tabela com os *envelopes* dos ossos do nosso modelo, e à direita uma tabela com os *envelopes* que foram guardados para o ficheiro que queremos importar. Deste modo, só temos que escolher a opção '*Match by Name*', e as definições dos *envelopes* serão copiadas dos ossos de uma tabela para a outra. Assim ficamos com o *skinning* e *weighting* do modelo exactamente igual ao que estava guardado no ficheiro.

Desta forma, conseguimos aplicar o mesmo *skinning* e *weighting* a diferentes modelos e animá-los de forma bastante semelhante, concretizando assim o objectivo de conseguir em reutilizar animações em diferentes modelos.

Finalmente, para animar um modelo, basta utilizar outra funcionalidade do 3ds Max chamada *Load Motion Capture File*, esta permite fazer *load* de um ficheiro de extensão *.bip*, que possui animações faciais e corporais (por exemplo de *MoCap*) e podemos aplicar essas animações a diferentes modelos virtuais (Figura 4.3).

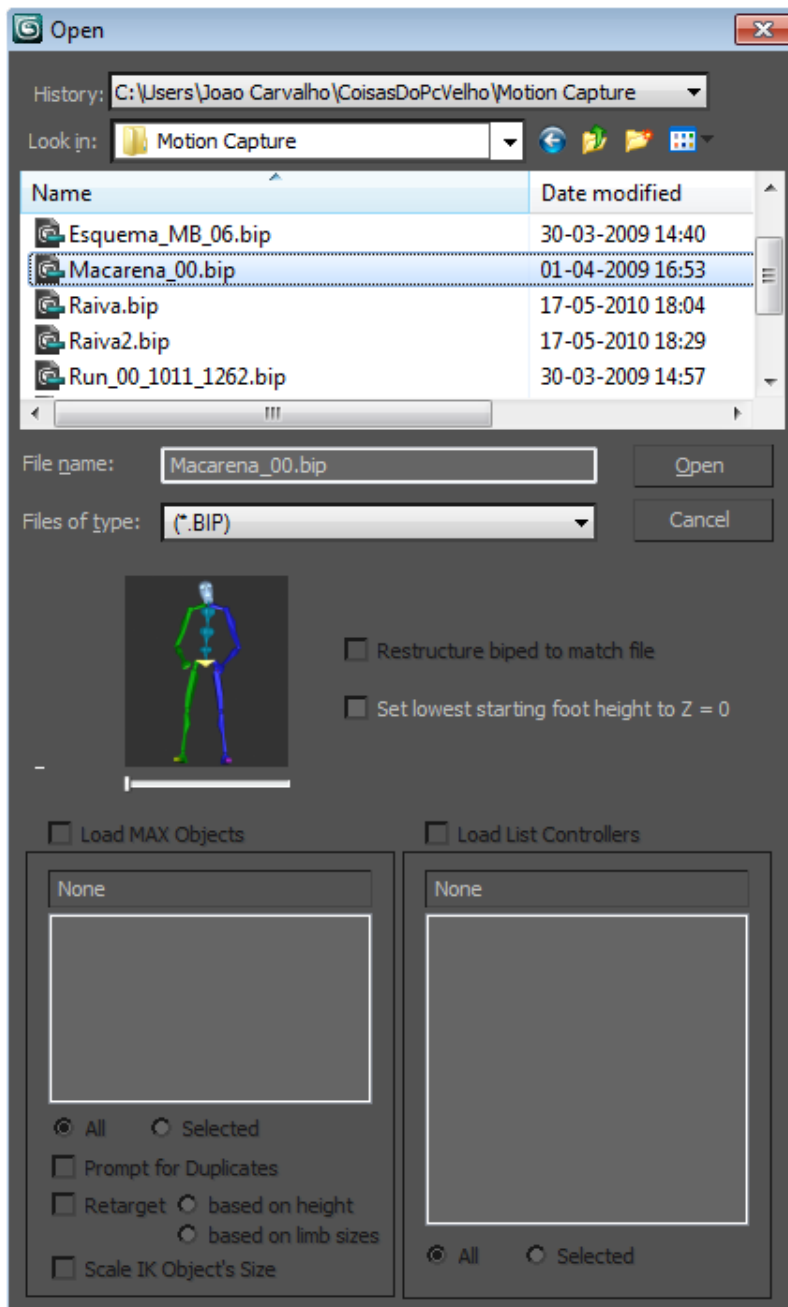


Figura 4.3: Funcionalidade do 3ds Max, *Load Motion Capture File*.

4.3 Interface Visual

4.3.1 Introdução

Com o decorrer do trabalho, considerei interessante e vantajoso desenvolver uma interface visual em Maxscript. Ou seja, construir uma interface que guiasse o animador nas diferentes fases do processo de *Rigging*, *Skinning* e *Weighting* dos modelos e na construção das animações propriamente ditas. De tal forma que todo este processo que foi explicado anteriormente, se tornasse automático e bastante mais rápido.

Uma vez que o 3ds Max era uma ferramenta nova, o primeiro passo consistiu em fazer vários tutoriais de Maxscript, a sua linguagem de *scripting* interna, para poder começar a desenvolver a dita interface visual.

4.3.2 Construção da Interface

A ideia inicial foi, uma vez que tinha duas fases no processo que são independentes uma da outra, decidi associá-las a duas janelas distintas, ou seja, numa janela o utilizador faz o *Rigging*, *Skinning* e *Weighting* ao seu modelo, e na outra janela é que se procede à construção da animação em si. Apresentamos então a janela inicial, que permite ao utilizador navegar para ambas as fases do processo (Figura 4.4).

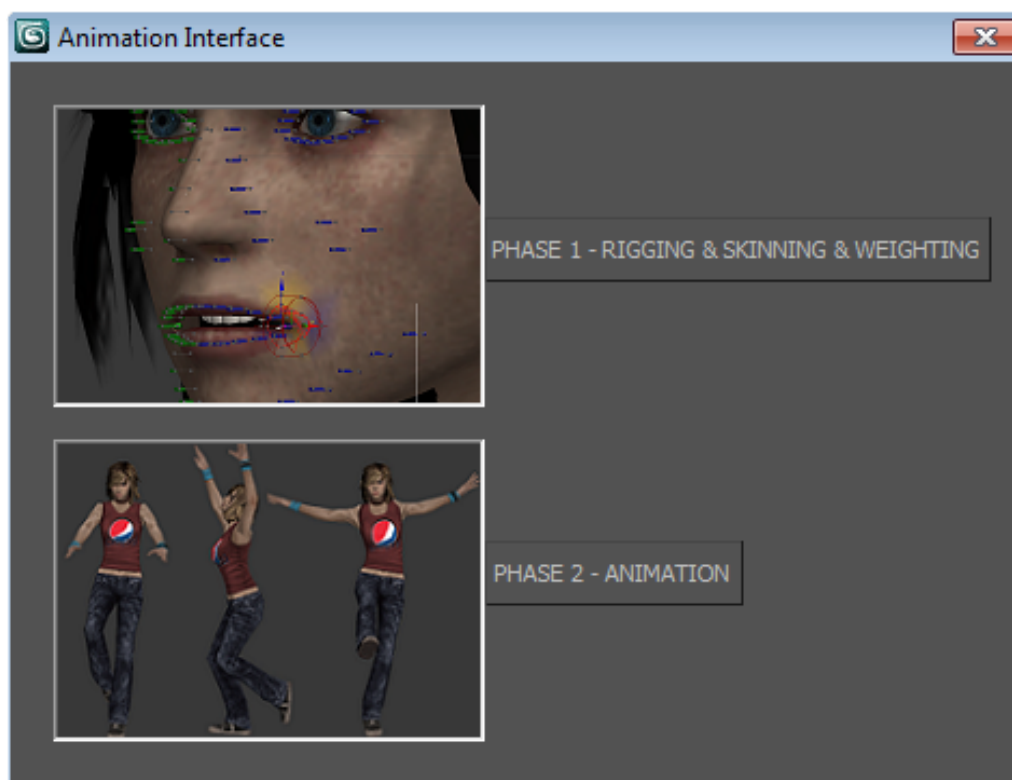


Figura 4.4: Menu inicial da interface.

Relativamente à primeira fase, a partir do script já desenvolvido identifiquei 6 passos necessários para preparar o modelo para ser animado:

- Os passos 1, 2 e 3 correspondem ao *Rigging*. Faz-se *load* do esqueleto, importa-se o modelo, e ajustam-se os ossos do esqueleto de forma a encaixar de forma correcta na malha poligonal do modelo.
- Os passos 4 e 5 correspondem ao *Skinning*. Adiciona-se o *modifier skin* ao modelo e de seguida adicionam-se os ossos do esqueleto.
- O passo 6 corresponde ao *Weighting*. Ou seja, faz-se *load* do ficheiro de extensão *.env* para o modelo que queremos, e obtemos assim um modelo pronto a animar.
- O último passo permite ao utilizador guardar o trabalho feito. Este é guardado sob a forma de um ficheiro de extensão *.max*.

De seguida apresentamos então a janela desenvolvida, onde o utilizador é guiado por todos os passos deste processo, recorrendo a simples botões (Figura 4.5).

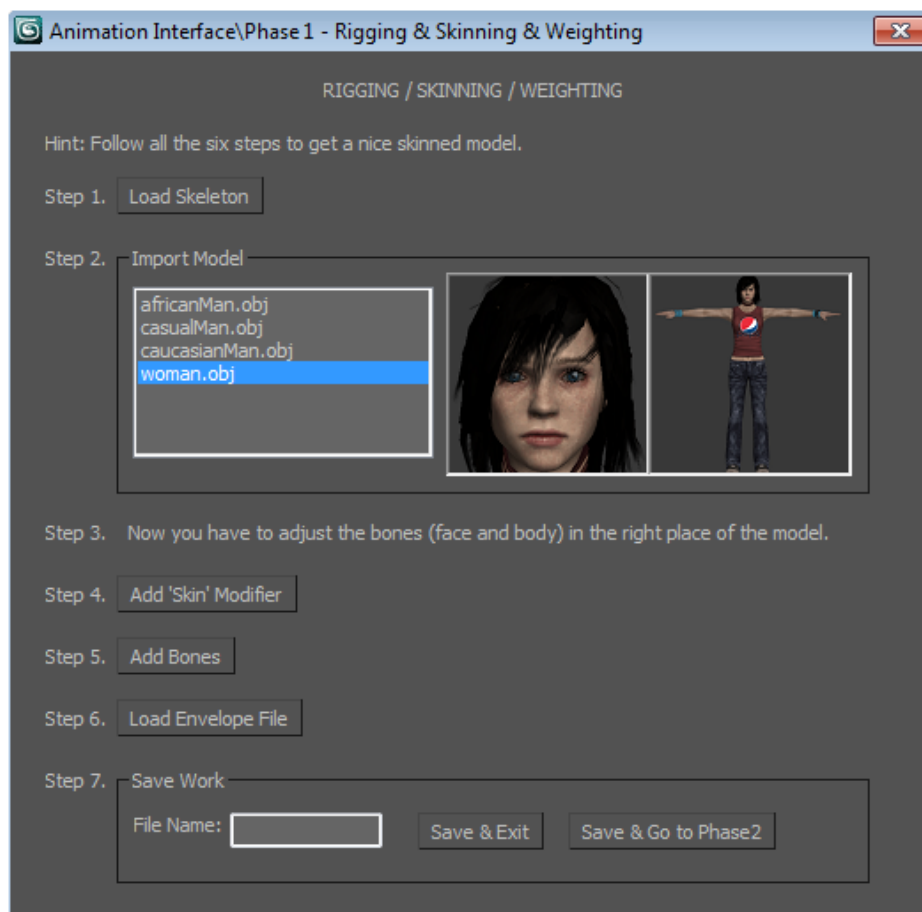


Figura 4.5: Phase 1 - Rigging & Skinning & Weighting.

Na janela correspondente à segunda fase do processo, a animação (Figura 4.6). Surgem as imagens da face e do esqueleto, e um conjunto de comandos e funcionalidades, nomeadamente:

- *Checkbuttons* em ambas as imagens, no sítio respectivo ao osso que vão mover.
- Comandos do painel de controlo, entre os quais: *translation*, *selection*, *rotation*, *reset* e *hide bones*.
- Funcionalidades para, construir, fazer *load* e gravar uma animação.

Todas estas funcionalidades da interface, estão explicadas em mais pormenor no manual de utilizador, Apêndice A.

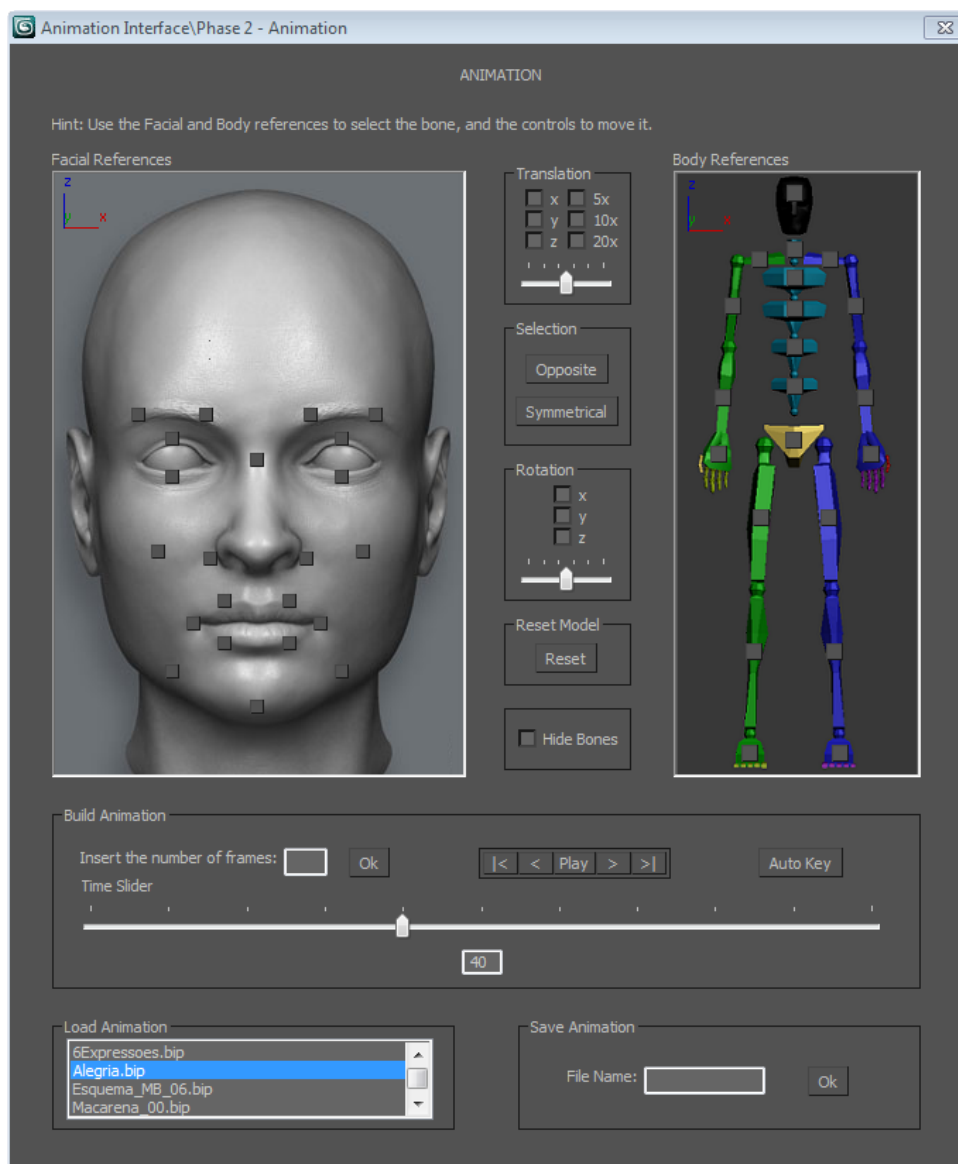


Figura 4.6: Phase 2 - Animation.

4.4 Modelos Virtuais Utilizados

Todos os modelos utilizados foram obtidos gratuitamente na internet. Encontrar estes modelos gratuitos implicou uma pesquisa demorada e exaustiva, uma vez que a esmagadora maioria dos modelos encontrados, são comercializados. Para conseguir rapidamente identificar os modelos fui-lhes atribuindo nomes sugestivos:

- Um homem de negócios (Figura 4.7(a)) [27].
- Um doutor (Figura 4.7(b)) [27].
- Uma mulher (Figura 4.8(a)) [28].
- Um homem casual (Figura 4.8(b)) [29].
- A face de um homem caucasiano (Figura 4.9(a)) [28].
- A face de um homem africano (Figura 4.9(b)) [30].

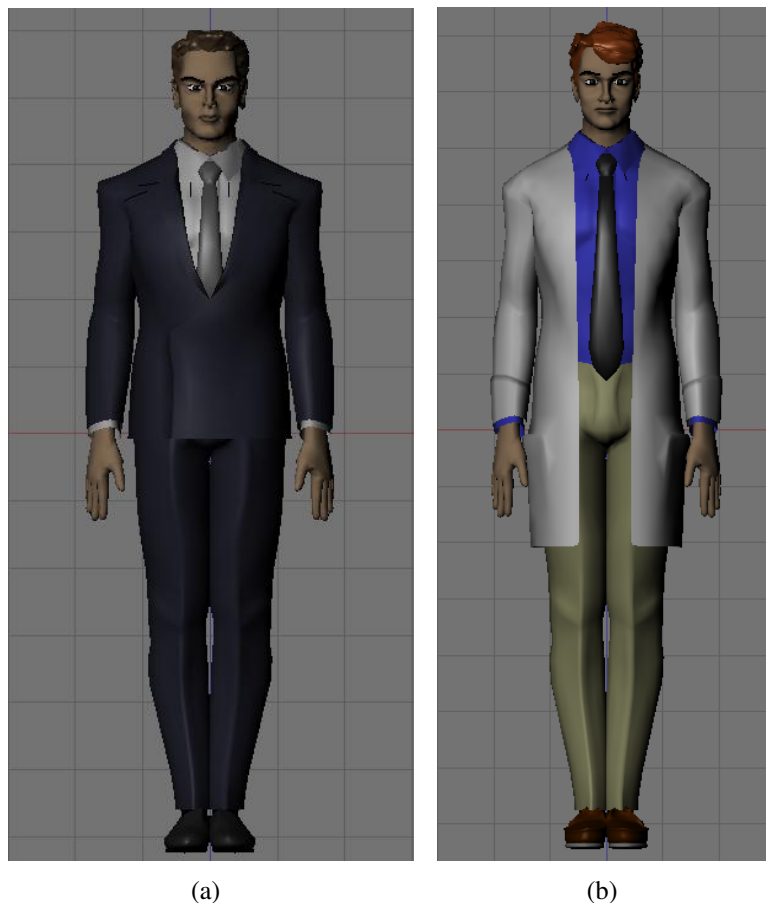


Figura 4.7: Modelos de corpo inteiro utilizados.

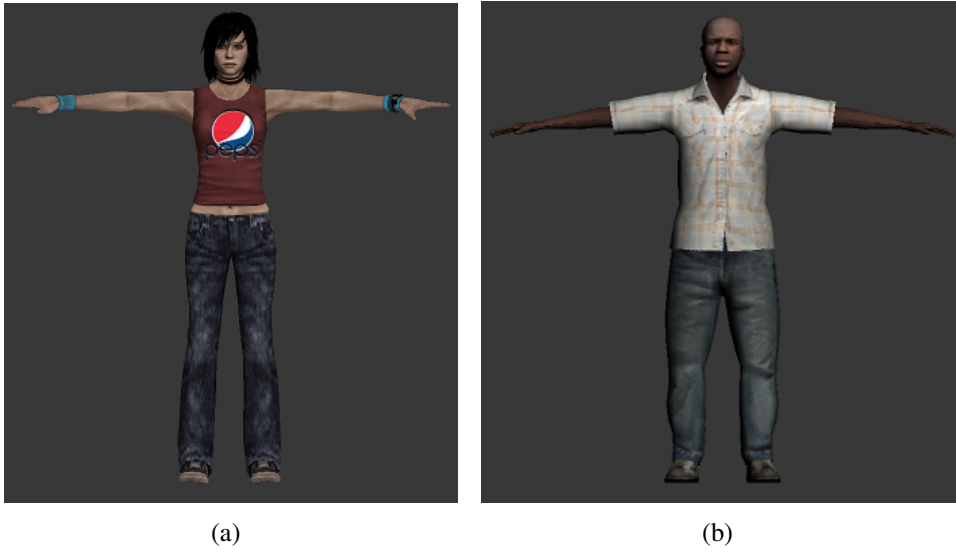


Figura 4.8: Modelos de corpo inteiro utilizados.

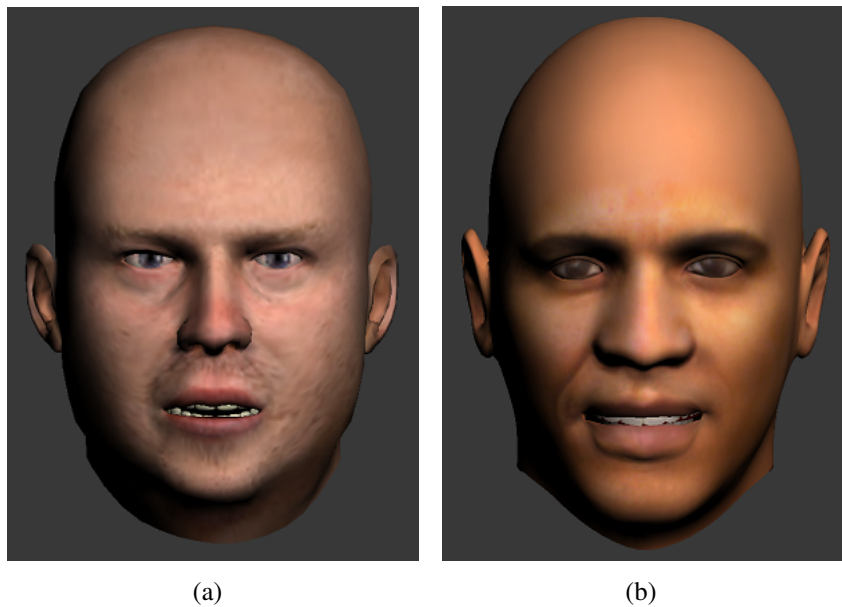


Figura 4.9: Modelos de face utilizados.

4.5 Ferramentas Utilizadas

Este projecto foi desenvolvido com as seguintes ferramentas:

- **3D Studio Max 2010** - ferramenta de modelação 3D, que permite fazer o *rendering* de imagens e animações. É bastante usado em produção de filmes de animação, criação de personagens de jogos 3D, publicidade, maquetas electrónicas e na criação de qualquer mundo virtual. O 3ds Max é considerado um dos principais softwares, na geração de animação 3D, e possui uma opção para integrar nos modelos 3D dados obtidos por *motion capture*, produzindo animações mais credíveis [31].
- **OGRE 1.6** - *Object-Oriented Graphics Rendering Engine* é um motor gráfico 3D orientado a objectos. É considerado por muitos, um dos melhores motores disponíveis de código aberto que existem. Foi escrito por uma pequena equipa, mas possui um grande número de colaboradores. A principal linguagem que usa é C++, porém existem versões de teste para Python, Java e .NET [32].
- **Microsoft Visual Studio 2008** - pacote de programas da Microsoft, para desenvolvimento de software, especialmente dedicado, ao Framework .NET e às linguagens Visual Basic (VB), C, C++, C# e J#. Também é um grande produto de desenvolvimento na área Web, usando a plataforma do ASP.NET. As linguagens com maior frequência nessa plataforma são: VB.NET (Visual Basic .Net) e o C# [33].
- **Blender 2.49b** - ferramenta gratuita de modelação 3D, desenvolvido pela Blender Foundation, para modelação, animação, textura, composição, renderização, edição de vídeo e criação de aplicações interactivas em 3D, tais como jogos, apresentações e outros, através de seu motor de jogo integrado, o Blender Game Engine. É um programa multi-plataforma, estando portanto disponível para diversos sistemas operacionais. Inclui suporte a Python como linguagem de script, que pode ser usada tanto no Blender, como no seu motor de jogo [34].

4.6 Ambiente de Desenvolvimento

- Sistema Operativo: Microsoft Windows 7 Professional 64-bit.
- Processador: Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz
- Memória RAM: 4,00 GB
- Placa Gráfica: NVIDIA Quadro FX 1700

4.7 Conclusão

Neste capítulo descrevemos todo o trabalho desenvolvido, em particular, duas abordagens para reutilizar animações em diferentes personagens virtuais. Concluimos que, a primeira, utilizando vértices, não era escalável para modelos virtuais com muitos vértices, e por isso muito lenta. A segunda abordagem utilizando *envelopes*, não era perfeita, mas conseguia cumprir o nosso objectivo, reutilizar animações em diferentes modelos virtuais.

Relativamente à interface visual desenvolvida, concluimos que facilita o trabalho do animador, mas que não prescinde completamente da sua intervenção para obter um resultado final realista. Apresentámos também os modelos dos humano virtuais e ferramentas que utilizámos durante todo o trabalho, e também o ambiente em que o mesmo foi desenvolvido.

Capítulo 5

Testes Efectuados

Este capítulo apresenta os testes efectuados utilizando o 3D Studio Max, e comparações de expressões faciais entre os diferentes modelos virtuais utilizados.

5.1 Introdução

Os objectivos do projecto eram, como já disse anteriormente, inserir mais modelos na plataforma *IViHumans* e reutilizar animações entre modelos diferentes. Os testes descritos neste capítulo foram efectuados para saber até que ponto os objectivos foram cumpridos.

Para cada teste efectuado explicarei todo o processo passo a passo e de seguida mostrarei os resultados que obtive utilizando como ferramenta de modelação o 3ds Max.

5.2 Testes no 3D Studio Max

5.2.1 Inserir um objecto estático na plataforma

Este teste é bastante simples e consiste em inserir um modelo estático na plataforma *IViHumans*. É necessário começar por exportar o modelo do 3ds Max para o Ogre 3D, o motor gráfico que se usa na plataforma. Para o fazer usamos o *OgreMax Scene Exporter for 3DS Max*, que foi retirado da internet no seguinte site [35], e é um *plugin* que exporta ficheiros para usar com o Ogre 3D.

Correndo a plataforma *IViHumans* e passando-lhe o nome da malha poligonal do objecto estático que queremos lá inserir, ou seja, uma *Entity* que neste caso é a mulher, o resultado obtido está representado na Figura 5.1.



Figura 5.1: O modelo da mulher inserido na plataforma *IViHumans*.

5.2.2 Inserir um humano virtual animado na plataforma

Para inserir um modelo virtual animado na plataforma, neste caso um humano virtual, já vamos poder tirar partido das funcionalidades da interface desenvolvida no contexto do 3D Studio Max. De seguida basta seguir os passos da fase 1 da interface, que guiam o utilizador nos processos de *Rigging*, *Skinning* e *Weighting*, obtendo assim um modelo capaz de ser animado na fase 2. Depois de na fase 2 construirmos a dita animação, vamos então proceder à exportação do modelo e da respectiva animação.

Recorremos de novo ao *OgreMax Scene Exporter for 3DS Max*, mas desta vez antes de podermos exportar seja o que for, temos que ir à opção '*Object Settings*', e no separador '*Mesh Animations*' seleccionamos as opções pretendidas, na Figura 5.2. No *drop down list* '*Mesh Animation Type*' escolhemos a opção '*Skeleton*', de seguida nos '*Skeleton Settings*', temos vários *checkbox*'s, são eles: '*Always Export Skeleton*', '*Remove Bones With no Influence*', '*Export Animations to Separate Skeleton Files*' e '*Link Separate Animations to Main Skeleton*', vamos seleccionar todos estes *checkbox*'s.

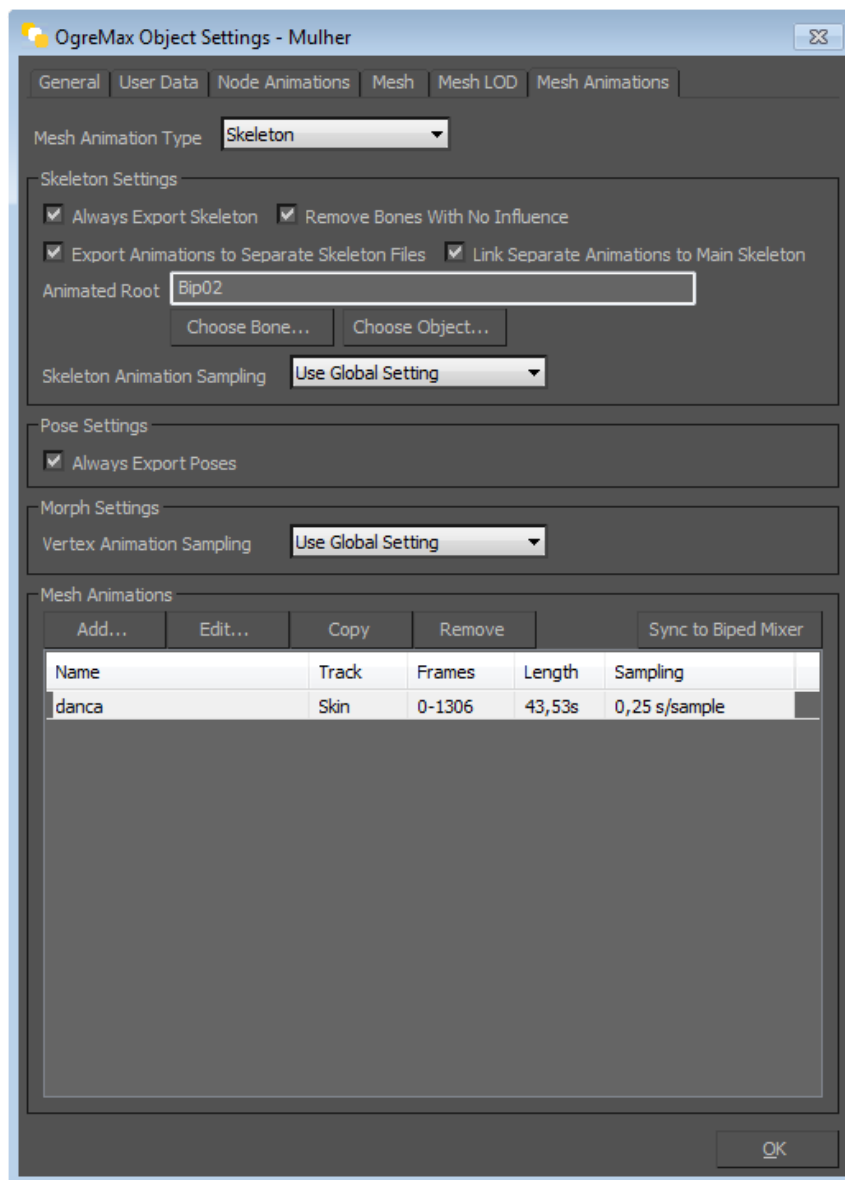
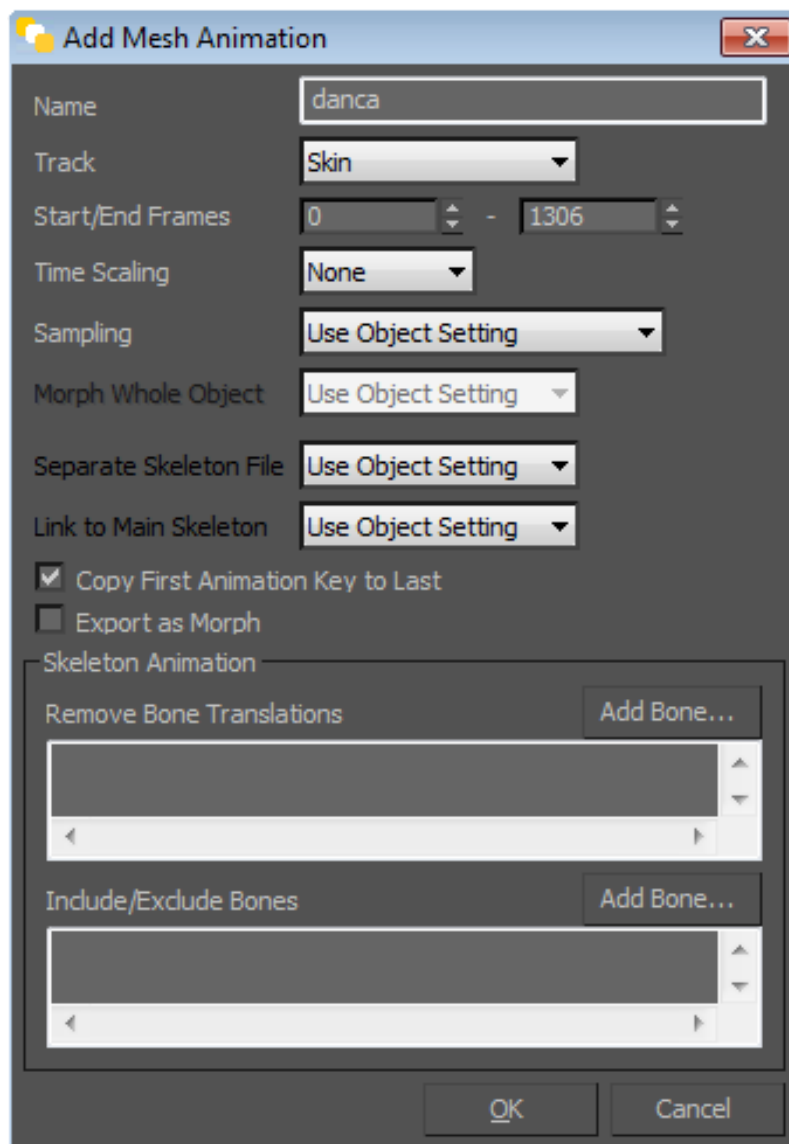


Figura 5.2: *OgreMax Scene Exporter for 3DS Max.*

De seguida na opção '*Mesh Animations*' vamos adicionar a dita animação, ao escolher a opção '*Add...*' vai surgir uma nova janela (Figura 5.3), nesta inserimos o nome da animação, este vai ter que coincidir rigorosamente com o nome que está no ficheiro *.ivihuman* na secção *TranslateAnims* da Figura B.2. De seguida escolhemos '*Skin*' no *drop down list 'Track'*, na opção '*Start/End Frames*' definimos o tamanho da animação inserindo a *frame* inicial e final, seleccionamos o *checkbox 'Copy First Animation Key to Last'* e premimos '*OK*'. Podemos agora finalmente passar à exportação do dito modelo animado.

Figura 5.3: *Add Mesh Animation*.

Corremos então a plataforma *IViHumans* e passamos-lhe o nome da malha poligonal do modelo animado que queremos lá inserir, ou seja a mulher, e em vez de ser uma *Entity* como no teste anterior é agora um *IViHuman*.

Logo vamos ter que criar dois ficheiros adicionais, tal como fizemos para o Blender, que são cruciais para que tudo funcione correctamente, e são eles o *.mcharacter* B.1 e o *.ivihuman* B.2, em cada um deles temos que definir os atributos respectivos, que podemos ver representados em ambas as figuras.

O resultado desta animação aparece representado na seguinte Figura 5.4 sobre a forma de uma sequência de 14 *frames*.



Figura 5.4: Animação do modelo da mulher na plataforma.

5.3 Comparação de Expressões Faciais

Estes testes de comparação foram feitos utilizando a ferramenta de animação desenvolvida no 3D Studio Max, e visam comparar as 6 expressões primárias de Paul Ekman, nomeadamente a felicidade, tristeza, raiva, nojo, surpresa e medo, para 3 modelos distintos.

As expressões faciais foram produzidas com o modelo do homem caucasiano à esquerda, e de seguida transferidas para os outros dois modelos, o modelo feminino e o modelo do homem africano (Figura 5.5).



Figura 5.5: Modelos utilizados.

Inicialmente comecei por seguir todos os passos da fase 1 da interface de animação, que guiam o utilizador nos processos de *Rigging*, *Skinning* e *Weighting*, obviamente que no passo 3 tentei para os vários modelos, colocar os ossos da face da forma mais semelhante possível, para que depois o resultado da expressão fosse obtido em boas condições.

Já na fase 2 da interface foram criadas e guardadas, uma animação de 10 *frames* para cada uma das 6 expressões faciais, e de seguida aplicadas a cada um dos 3 modelos, os resultados obtidos estão representados nas seguintes figuras.



Figura 5.6: Modelos distintos mostrando Alegria.



Figura 5.7: Modelos distintos mostrando Tristeza.



Figura 5.8: Modelos distintos mostrando Raiva.



Figura 5.9: Modelos distintos mostrando Medo.



Figura 5.10: Modelos distintos mostrando Nojo.



Figura 5.11: Modelos distintos mostrando Surpresa.

Estes testes foram feitos com o intuito de demonstrar que, utilizando o mesmo esqueleto genérico e aplicando a mesma animação facial a cada um dos modelos, conseguimos com a nossa abordagem obter expressões faciais bastante semelhantes nos 3 diferentes modelos testados.

5.4 Conclusão

Neste capítulo apresentámos todos os testes realizados. O primeiro teste consistia em inserir um objecto estático na plataforma *IViHumans* com o 3ds Max. O resultado foi apresentado sob a forma duma imagem que retrata o modelo da mulher inserido na plataforma, concluímos que o modelo tinha um aspecto bastante realista, e bastante mais qualidade que o modelo do homem de negócios inserido na plataforma usando o Blender Figura C.2.

O segundo teste, consistia em inserir um modelo virtual animado na plataforma *IViHumans* também com o 3ds Max. Esta animação foi apresentada sob a forma de uma imagem com uma sequência de 14 *frames*, obtivemos resultados bastante bons mas, infelizmente pela imagem não se consegue ver a fluidez da dança da animação.

Finalmente, em relação às comparações de expressões faciais, e este sim é o teste mais importante, foi um teste bastante positivo pois conseguimos cumprir o objectivo, e obter seis expressões faciais bastante semelhantes nos três diferentes modelos testados.

Não foram feitas comparações de animações corporais, pois não achámos que fosse relevante testar as animações se que obteriam, com um simples esqueleto *Biped* do 3ds Max.

Capítulo 6

Conclusão e Trabalho Futuro

Neste capítulo apresenta-se um resumo do trabalho realizado, tecem-se algumas conclusões e discutem-se perspectivas futuras de evolução do trabalho.

6.1 Conclusões

Representar emoções humanas em personagens virtuais é uma área de investigação que está em constante evolução e aperfeiçoamento, cada vez se encontram mais métodos e técnicas para o fazer, de forma a que os humanos virtuais sejam representados com um aspecto o mais realista possível. Esta representação vai depender bastante da qualidade da modelação do humano virtual em si, e também da qualidade da técnica utilizada para o animar.

Neste relatório apresentou-se uma interface visual que permite ao utilizador, não só representar emoções em personagens virtuais à sua escolha, como reutilizar essas animações em diferentes modelos.

A solução proposta não é perfeita visto que não dispensa por completo a tarefa do animador, pois na primeira fase do processo de animação é o utilizador que tem de, ajustar a posição do modelo com o esqueleto e posicionar os ossos no sítio certo do modelo, para que mais tarde o *Skinning* fique bem feito. Ou seja, esta solução apesar de diminuir claramente o esforço do animador, não prescinde completamente da sua intervenção para obter um resultado final realista.

Contudo, tem o mérito de ser simples de implementar e de usar, permitindo uma poupança significativa de tempo no processo de animação facial e corporal de humanos virtuais.

Relativamente aos prazos de entrega do PEI, o meu relatório final não foi entregue dentro do prazo limite. Isto deve-se simplesmente ao facto de nos primeiros 2 meses, ter estado a trabalhar com o Blender como ferramenta de modelação 3D. E uma vez que só passado esse tempo, é que foi adquirida a licença do 3D Studio Max, todo o trabalho realizado anteriormente foi em vão. E só passados dois meses comecei realmente

a trabalhar na tese, meses esses que eram essenciais agora na escrita do relatório final.

6.2 Perspectivas Futuras

No que respeita ao trabalho futuro, será acrescentado um motor de física à camada gráfica da plataforma *IViHumans*. Este motor simula a física de Newton em modelos 3D, usando variáveis como massa, velocidade, fricção, e resistência ao ar, tornando possível simular condições de vida real. O motor de física será o *OgreODE* [36].

ODE (Open Dynamics Engine) é uma biblioteca *open-source* que simula dinâmica de corpo rígido, que é muito útil para simular veículos, objectos em ambientes virtuais e personagens virtuais. O *OgreODE* é um intermediário que permite que o *ODE* corra no motor do *Ogre*.

Relativamente à ferramenta de animação que foi desenvolvida no 3D Studio Max, o ideal seria de alguma forma conseguir automatizar o terceiro passo da fase 1, *Rigging & Skinning & Weighting*. Não será uma tarefa fácil de realizar, mas desta maneira o utilizador não teria que ajustar o modelo e posicionar todos os ossos do esqueleto no sítio certo manualmente, e a ferramenta tornar-se-ia completamente automática. Isto seria bastante benéfico, pois pouparia ainda mais tempo ao animador na utilização desta ferramenta.

Apêndice A

Manual de Utilizador

Neste apêndice apresenta-se o manual de utilizador da interface desenvolvida, as regras de instalação da ferramenta, assim como uma descrição pormenorizada de todas as suas funcionalidades.

A.1 Introdução

Esta ferramenta desenvolvida, basicamente é um script que é executado no contexto do 3D Studio Max, e permite ao utilizador animar um modelo à sua escolha. Para isso é essencial que faça primeiro o *rigging*, *skinning* e *weighting* do dito modelo na fase 1, e depois sim avance para a fase 2 para proceder à animação.

Esta interface oferece ao utilizador, um menu onde podemos aceder às duas fases do processo de animação de um modelo, a primeira fase é "*Rigging & Skinning & Weighting*" e a segunda fase é "*Animation*".

A.2 Instalação

A instalação desta interface é extremamente simples, basta fazer download do script "*interfaceMaxScript.ms*" no sítio "<http://labmag.di.fc.ul.pt/n/>", copiá-lo para a pasta dos *scripts* do 3D Studio Max na seguinte localização "*C:\Program Files\Autodesk\3ds Max 2010\Scripts*", e de seguida no 3D Studio Max seleccionar o separador '*MAXScript*' e fazer '*Run Script*' para abrir o dito *scrip*.

A.3 Menu da Interface

Aqui é descrito o menu da interface. Esta primeira janela da interface possui dois botões, cada um deles permite-nos aceder a uma das fases do processo de animação de um modelo humano (Figura A.1).

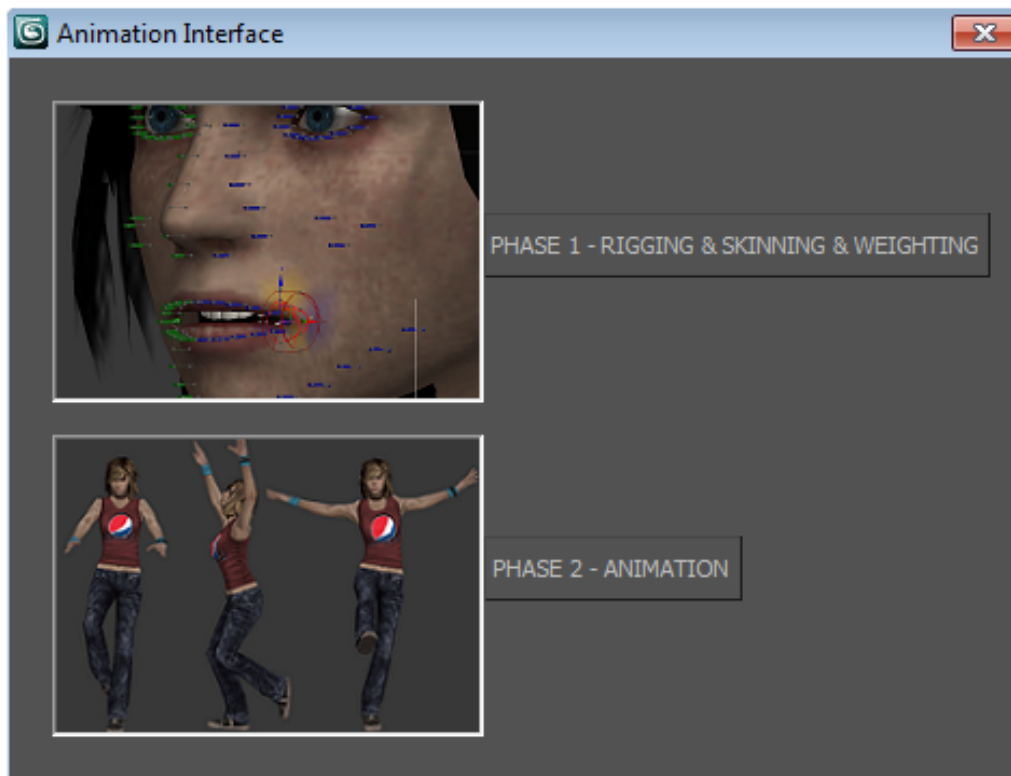


Figura A.1: Menu inicial da interface.

PHASE 1 - RIGGING & SKINNING & WEIGHTING

Figura A.2: Botão <Phase 1 - Rigging & Skinning & Weighting>.

Este botão permite ao utilizador ir para a primeira fase onde pode fazer o *Rigging*, *Skinning* e *Weighting* de um modelo humano à sua escolha (Figura A.2).

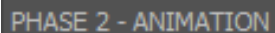


Figura A.3: Botão <Phase 2 - Animation>.

Este botão permite ao utilizador ir para a segunda fase onde pode animar o seu modelo humano (Figura A.3).

A.4 Phase 1 - Rigging & Skinning & Weighting

Aqui descreve-se a fase 1 da interface, o *Rigging & Skinning & Weighting*. Esta é a segunda janela da interface, e surge quando o utilizador clica no botão da fase 1 do menu inicial, esta janela guia o utilizador através dos vários passos do *Rigging*, *Skinning* e *Weighting* de um modelo humano (Figura A.4).

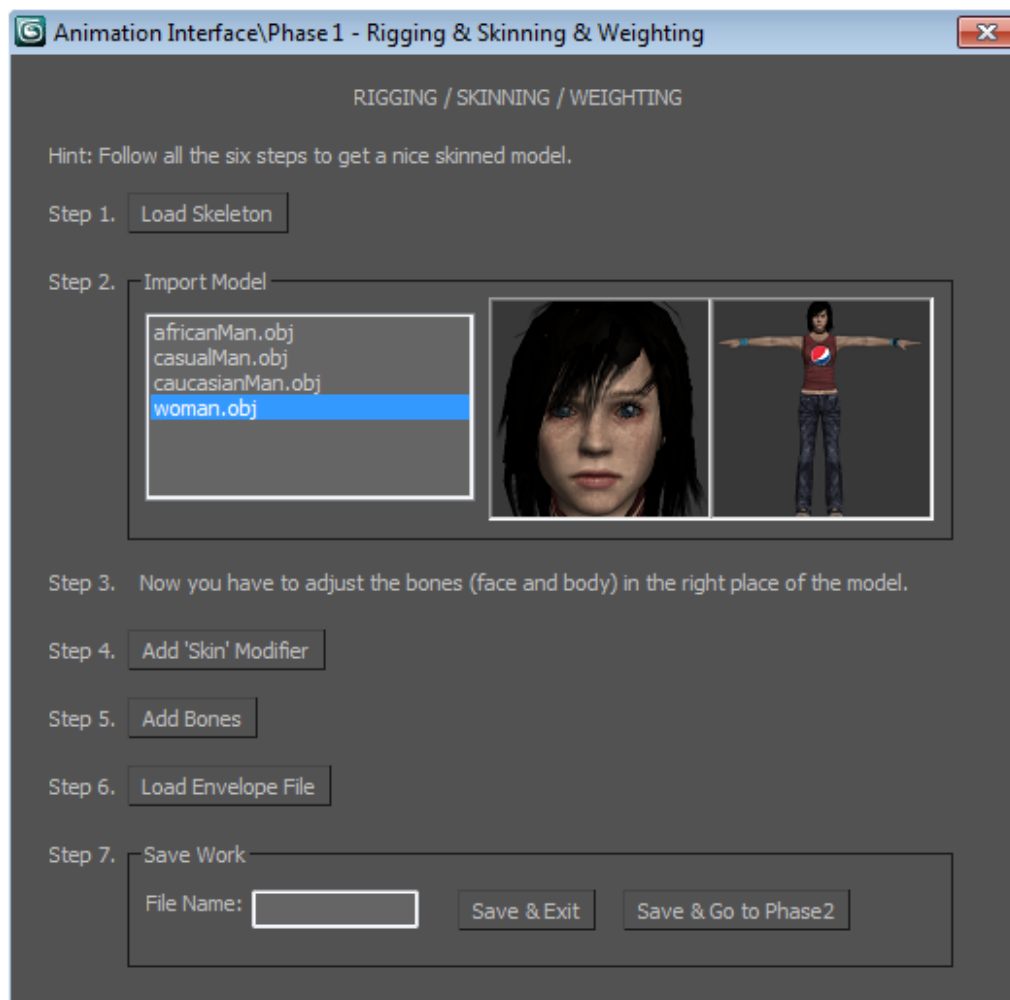


Figura A.4: Menu da primeira fase da interface.

A.4.1 Passo 1

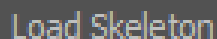
A rectangular button with a dark grey background and the text "Load Skeleton" in a light grey, sans-serif font.

Figura A.5: Botão <Load Skeleton>.

Neste primeiro passo podemos fazer *load* do nosso esqueleto *Biped* que irá animar o modelo humano(Figura A.5).

A.4.2 Passo 2



Figura A.6: Drop-box <Import Model>.

No passo 2, temos um *drop-box* com os modelos disponíveis que podemos escolher, se clicarmos num deles podemos ver a sua cara e corpo do lado direito, se fizermos duplo clique importamos o respectivo modelo humano(Figura A.6).

A.4.3 Passo 3

Neste passo cabe ao utilizador ajustar o modelo do humano e posicionar os ossos do esqueleto no sítio certo do modelo, é o único passo desta fase que não é feito automaticamente.

A.4.4 Passo 4

A rectangular button with a dark grey background and the text "Add 'Skin' Modifier" in a light grey, sans-serif font.

Figura A.7: Botão <Add 'Skin' Modifier>.

No passo 4, temos um botão que adiciona um *modifier skin* ao modelo, este *modifier* permite deformar um objecto, neste caso o modelo do humano escolhido, usando um esqueleto (Figura A.7).

A.4.5 Passo 5

A rectangular button with a dark grey background and the text "Add Bones" in a light grey font.

Figura A.8: Botão <Add Bones>.

Neste passo temos botão <Add Bones> que como o nome indica, adiciona todos os ossos do esqueleto ao *modifier skin* do modelo humano (Figura A.8).

A.4.6 Passo 6

A rectangular button with a dark grey background and the text "Load Envelope File" in a light grey font.

Figura A.9: Botão <Load Envelope File>.

No passo 6, temos um botão que faz *load* do ficheiro *.env* que contém as propriedades dos *envelopes* de cada osso do esqueleto, ou seja contém o *skinning* e *weighting* do modelo humano (Figura A.9).

A.4.7 Passo 7

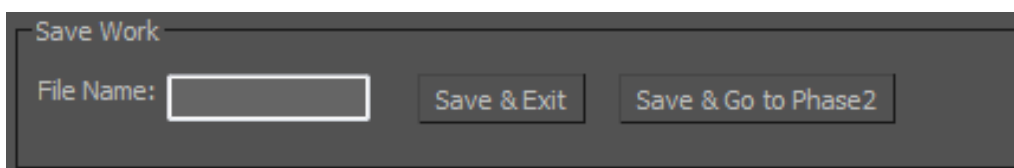


Figura A.10: *Editbox* <filename>, botão <Save & Exit> e botão <Save & Go to Phase2>.

No último passo, temos uma *editbox* onde podemos inserir o nome com que pretendemos gravar todo o nosso trabalho feito até aqui num ficheiro *.max*. De seguida temos dois botões, o botão <Save & Exit> onde podemos escolher gravar e voltar ao menu inicial. E o botão <Save & Go to Phase 2> onde podemos gravar e ir directamente para a fase 2, para começarmos a animar o nosso modelo humano (Figura A.10).

A.5 Phase 2 - Animation

Aqui descreve-se a fase 2 da interface, a *Animation* (Figura A.11). Esta é a terceira janela da interface, e surge quando o utilizador clica no botão da fase 2 do menu inicial, esta janela permite ao utilizador não só construir animações faciais e corporais como também guardá-las e fazer *load* das mesmas. A primeira coisa que se salta à vista ao utilizador nesta janela são as duas imagens, uma corresponde às referências faciais e outra às referências corporais, ambas as imagens têm vários *checkbox* posicionados nos sítios dos ossos do modelo.

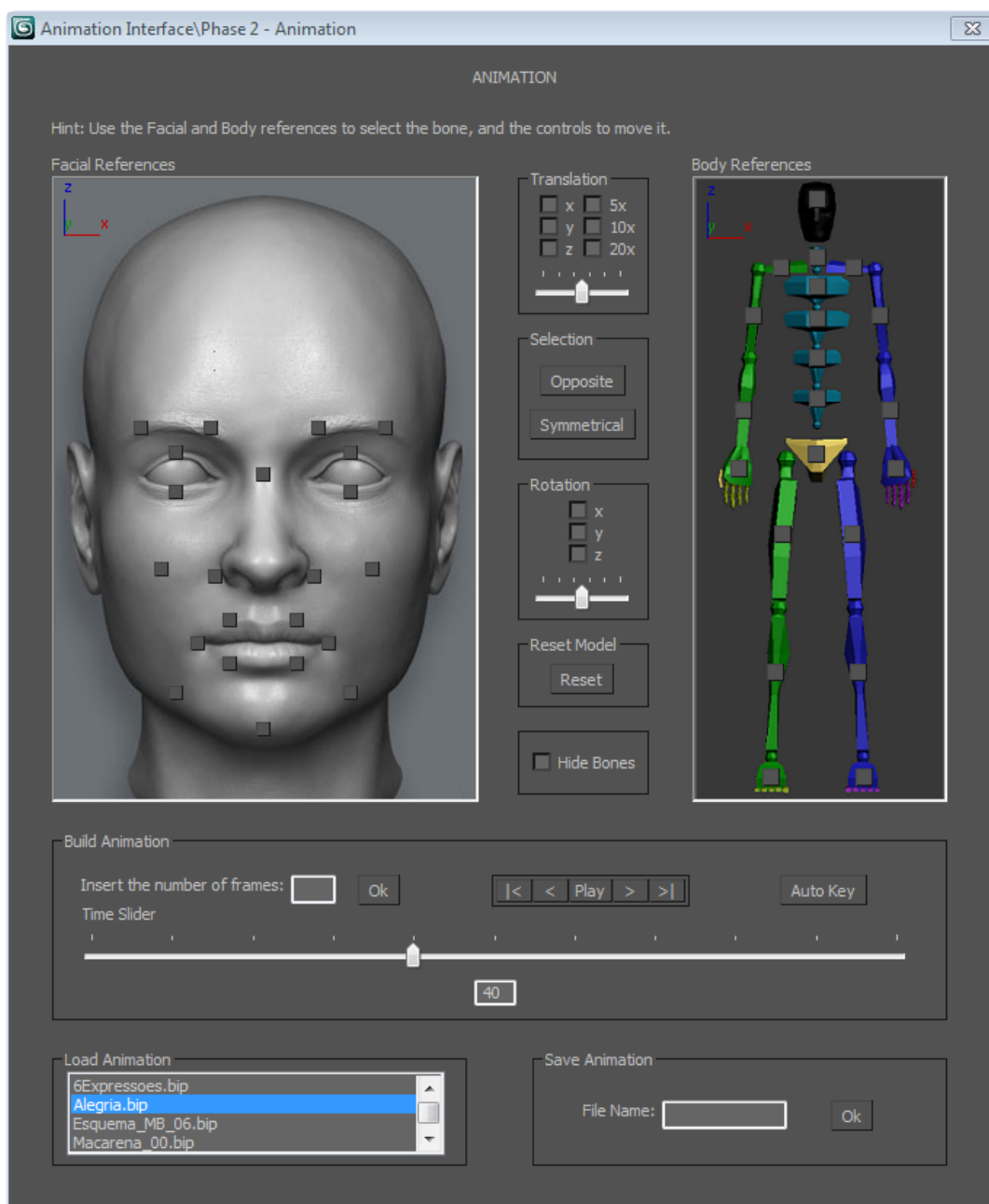


Figura A.11: Menu da segunda fase da interface.

Quando queremos seleccionar um determinado osso ou grupo de ossos, clicamos no *checkbox* respectivo e automaticamente seleccionamos o(s) osso(s) do modelo que pretendemos mover.

A.5.1 Painel de controlo

Relativamente aos controlos, a interface oferece vários tais como translação, rotação, selecção entre outros.

Translação

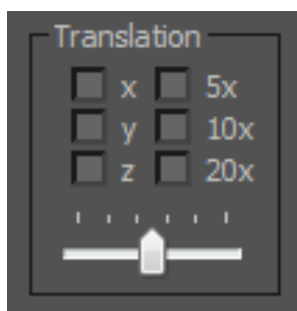


Figura A.12: Controlo de Translação.

- Os controlos de translação possuem 3 *checkbox*'s com os 3 eixos do referencial, são eles X, Y e Z, tem também três *checkbox*'s para aumentar o factor da translação 5, 10 e 20 vezes, e finalmente um *slider* para aplicar a translação aos ossos seleccionados (Figura A.12).

Seleccção

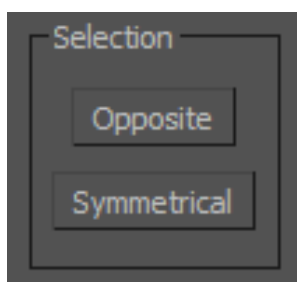


Figura A.13: Controlo de Seleccção.

- Os controlos de selecção possuem 2 botões, são eles o comando *Opposite* e o *Symmetrical*. A função do botão <Opposite> é seleccionar o osso oposto ao que está seleccionado. A função do botão <Symmetrical> é tendo um determinado osso selecci-

onado, seleccionar também o osso simétrico, logo ficam ambos os ossos seleccionados (Figura A.13).

Rotação

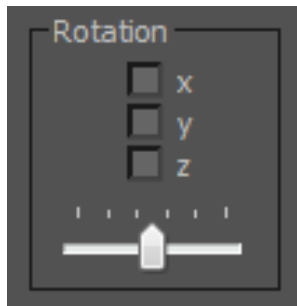


Figura A.14: Controlo de Rotação.

- Os controlos de rotação possuem 3 *checkbox*'s com os 3 eixos do referencial, são eles X, Y e Z, e um *slider* para aplicar a rotação aos ossos seleccionados (Figura A.14).

Reset

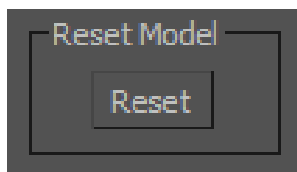


Figura A.15: Controlo de Reset.

- A função do botão <Reset> é como diz o nome fazer *reset* ao modelo, ou seja, depois de o utilizador alterar a posição do modelo, se desejar voltar à posição inicial do modelo pode fazê-lo (Figura A.15).

Hide Bones

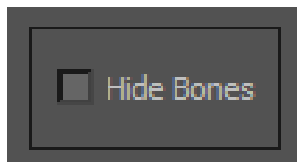


Figura A.16: Controlo Hide Bones.

- A função do *checkbox* <Hide Bones> é como o nome diz, esconder os ossos do modelo humano (Figura A.16).

A.5.2 Animação

Relativamente à animação em si, a interface oferece ao utilizador várias funcionalidades, são elas, construir uma animação, guardar uma animação e também fazer *load* de uma animação.

Construir Animação:

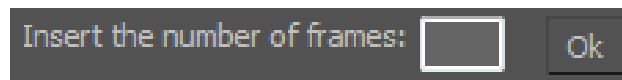


Figura A.17: *Editbox* do número de *frames*.

- É possível ao utilizador definir o número de *frames* que terá a sua animação, para isso deve inserir o número de *frames* na *editbox* e clicar "Ok" (Figura A.17).

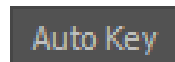


Figura A.18: Botão <Auto Key>.

- O botão <Auto Key> permite ao utilizador criar uma *key* na *frame* em que se encontra (Figura A.18).



Figura A.19: *Time Slider* do número de *frames*.

- O *Time Slider* permite ao utilizador navegar na animação *frame* a *frame*, para isso basta-lhe arrastar o *slider* para a *frame* desejada, a *editbox* em baixo dá ao utilizador feedback sobre a *frame* em que se encontra (Figura A.19).

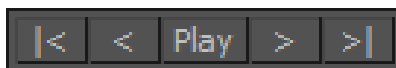


Figura A.20: Atalhos de navegação na animação.

- Estes 5 botões formam um grupo de atalhos de navegação na animação, da esquerda para a direita; ir para a *frame* inicial, ir para a *frame* anterior, fazer *play* à animação, ir para a *frame* seguinte e ir para a *frame* final (Figura A.20).

Fazer *Load* de uma Animação:

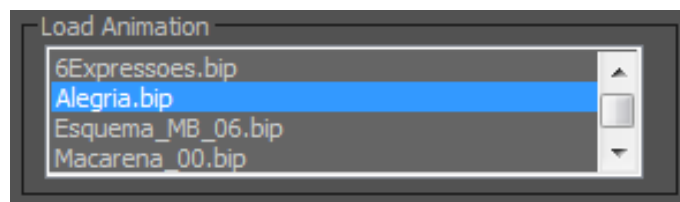


Figura A.21: *Dropbox* <Load Animation>.

- O *Dropbox* <Load Animation> permite ao utilizador fazer *load* de uma animação, fazendo um duplo clique na mesma (Figura A.21).

Guardar Animação:

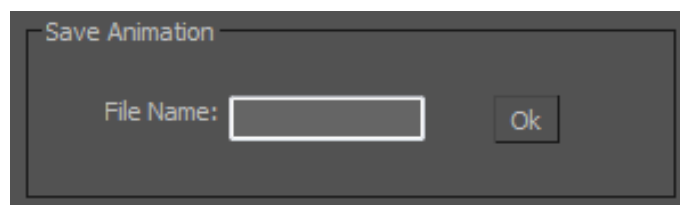


Figura A.22: *Editbox* do nome do ficheiro.

- Existe também a possibilidade de o utilizador poder gravar as suas animações, basta introduzir o nome do ficheiro na *editbox* e clicar "Ok" (Figura A.22).

Apêndice B

Ficheiros de Configuração

Os seguintes ficheiros são necessários e indispensáveis para poder inserir humanos virtuais na plataforma *IViHumans*. Para cada um desses humanos virtuais tem que existir um ficheiro *.mcharacter* (Figura B.1), e um ficheiro *.ivihuman* (Figura B.2).

B.1 Ficheiro *.Mcharacter*

Este ficheiro divide-se em duas secções, são elas a *[General]* e a *[SteeringBehaviorsSettings]*. A secção *General* tem três atributos, são eles a massa, velocidade máxima e a máxima força aplicável. A secção *SteeringBehaviorsSettings* contém as definições para os *SteeringBehaviors* que vão ser usados (Figura B.1).

```
1  ## Parameter definition file for MovingCharacter Male
2
3  [General]
4  mass=0.3
5  maxSpeed=10.0
6  maxForce=100.0
7
8  [SteeringBehaviorsSettings]
9  walkMinSpeed=5.0
10 arriveDecDist=20.0
11 arriveStopDist=5.0
12 arriveSpeed=9.0
13 arriveDecel=6.0
```

Figura B.1: Exemplo de um ficheiro *.mcharacter*.

B.2 Ficheiro *.IViHuman*

Este ficheiro contém os quatro seguintes atributos: o nome do ficheiro que contém a malha poligonal deste humano virtual, a direcção para onde o humano virtual está virado, o nome do osso da cabeça do humano virtual e a altura dos olhos do humano virtual quando os seus pés estão à altura zero. Contém ainda duas secções, são elas a *[TranslateAnims]* e a *[Transform]*. A secção *TranslateAnims* contém as animações para este respectivo humano virtual. A secção *Transform* contém as operações de transformação, *scale*, *translate* e *rotate* pela ordem definida no ficheiro (Figura B.2).

```
1  ## Parameter definition file for IViHuman Male
2  #TODO documentar
3
4  meshFileName=Male.mesh
5  defaultFacing=<default>
6  headBone=look
7  eyeHeight=14.2
8
9  # Translation Animations
10 [TranslateAnims]
11 animacao1=-1 0.00001 0
12
13 # Transformation operations to be applied at load time.
14 # The transformations are applied in alphabetical order of the keys.
15 [Transform]
16 a=scale 3
17 b=translate 20 18.5 20
18 c=rotate 0.999 -0.0445 0 0
```

Figura B.2: Exemplo de um ficheiro *.ivihuman*.

Apêndice C

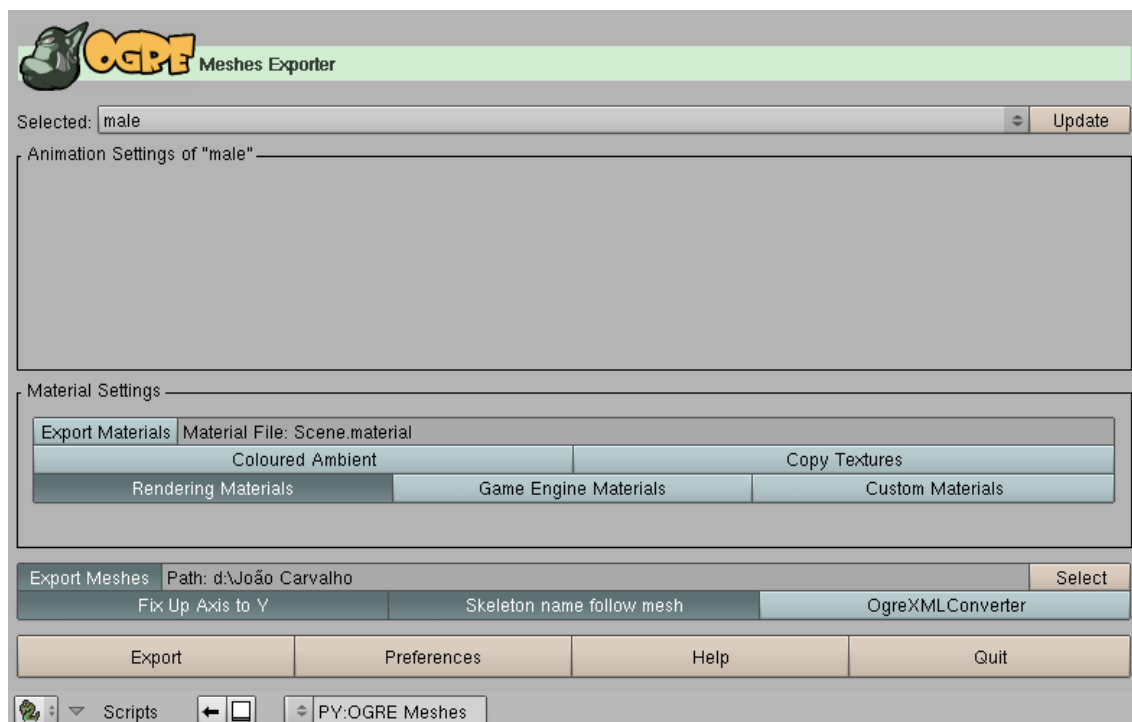
Testes Efectuados no Blender

Aqui são descritos os testes realizados com o Blender como ferramenta de modelação 3D. Este testes iniciais foram realizados nos primeiros dois meses do PEI, antes de adquirirmos uma licença para trabalhar com o 3ds Max. E por isso, não contribuíram de forma alguma para o desenvolvimento deste trabalho, a não ser como forma de comparação de resultados obtidos, com os testes de inserção de objectos e humanos virtuais animados na plataforma *IViHumans* realizados com o 3ds Max.

C.1 Inserir um objecto estático na plataforma

Inserir um objecto estático na plataforma veio a revelar-se um grande desafio, pois nada funcionava à primeira. O primeiro passo a fazer era exportar o modelo dos humanos do Blender, para isto foi usado o *Ogre Meshes Exporter* Figura C.1, que é um conjunto de *scripts* em Python e que correm directamente no Blender. Depois de actualizar a versão do Blender de 2.48a para a última 2.49b, o Python de 2.5.2 para a versão 2.6.3 e já com os *scripts* instalados procedi então aos testes de exportação.

Os primeiros testes foram feitos apenas com o objectivo de inserir um objecto estático na plataforma, e foram feitos utilizando o modelo do homem de negócios Figura 4.7(a).

Figura C.1: *Ogre Meshes Exporter*.

Depois de várias tentativas e alguns tutoriais de exportação, o melhor resultado foi obtido com as seguintes opções da Figura C.1 seleccionadas, a opção ”*Rendering Materials*” que serve para exportar os materiais da cena, a opção ”*Fix Up Axis to Y*” para mudar o referencial e colocar verticalmente o modelo na plataforma e a opção ”*Skeleton name follow mesh*” serve apenas para no ficheiro *.skeleton* (esqueleto do humano) usar o nome da malha poligonal (*mesh*) original.

Depois de exportar o objecto para *ogre mesh*, é necessário converter o ficheiro para XML, esta conversão é feita a partir do *OgreXmlConverter* que é uma ferramenta do OGRE que serve para manipular *mesh*'s, consegue-se assim converter os seguintes ficheiros em binário, *.mesh* e o *.skeleton* para XML.

Correndo a plataforma *IViHumans* e passando-lhe o nome da malha poligonal do objecto estático que queremos lá inserir, ou seja, uma *Entity* que neste caso é o humano, o resultado obtido quando comparado com o homem de negócios da Figura 4.7(a) deixa um pouco a desejar, pois a qualidade do modelo diminui de forma drástica desde a fase em que está no Blender até à fase em que está na plataforma, como podemos ver representado na seguinte Figura C.2.



Figura C.2: O modelo do homem de negócios inserido na plataforma *IViHumans*.

C.2 Inserir um humano virtual animado na plataforma

Para inserir um humano virtual animado na plataforma foi necessário antes de mais nada modelar a expressão facial do modelo humano no Blender, para depois esta poder ser exportada para a plataforma. Esta animação é feita aplicando rotações e translações aos ossos do esqueleto do modelo representado Figura C.3(a).

Primeiro é preciso no modo '*Weight Paint*' associar o peso de cada osso do esqueleto, ou seja, quais os vértices que vão estar associados a um determinado osso, por ex, quando deslocar um osso os vértices que estão associados a ele vão também ser deslocados. Depois de todos os ossos terem um peso associado, vamos então passar à modelação da expressão facial.

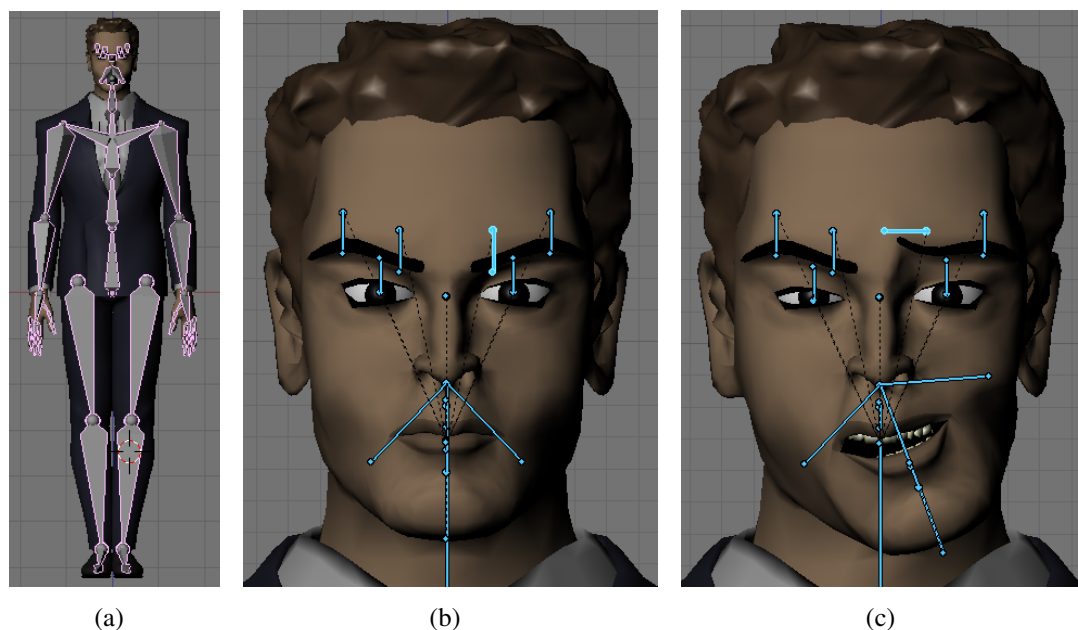


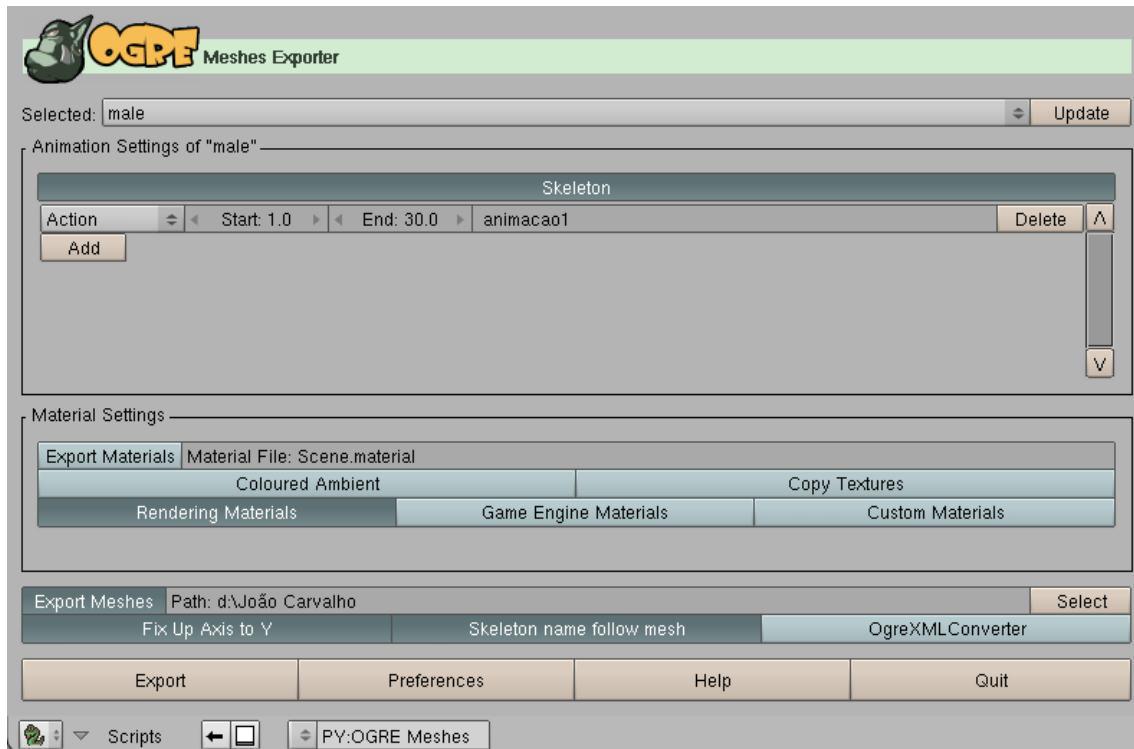
Figura C.3: Esqueleto e face do modelo.

A modelação é feita quando alteramos os ossos do esqueleto de forma a obter a expressão pretendida, na Figura C.3(c) conseguimos ver as modificações causadas na face do modelo resultantes de simples rotações dos ossos da sobrancelha esquerda, queixo, boca e olho direito.

Em relação à animação, é obtida quando alteramos o humano virtual de forma diferente em diferentes *frames*, por exemplo, se na *frame* 1 for mantida a posição neutra Figura C.3(b), e na *frame* 30 for alterado o modelo de forma a criar uma expressão facial Figura C.3(c), o próprio Blender calcula as *frames* intermédias originando assim uma animação de 30 *frames*.

De seguida vamos então exportar o humano virtual e a respectiva animação facial, recorreremos de novo ao *Ogre Meshes Exporter*, mas desta vez já existe na secção *Animation Settings of 'male'* uma *action* que começa na *frame* 1 e acaba na *frame* 30, esta animação tem o nome de 'animação1' Figura C.4.

Convertemos então os ficheiros com o *OgreXmlConverter* da mesma forma como fizemos no teste com o objecto estático.

Figura C.4: *Ogre Meshes Exporter*.

Corremos a plataforma *IViHumans* e passamos-lhe o nome da malha poligonal do humano virtual animado que queremos lá inserir, e que em vez de ser uma *Entity* como no teste anterior é agora um *IViHuman*. Desta forma vamos ter que criar dois ficheiros adicionais que são cruciais para que tudo funcione correctamente, e são eles o *.mcharacter* B.1 e o *.ivihuman* B.2, em cada um deles temos que definir os atributos respectivos, que podemos ver representados em ambas as figuras citadas anteriormente.

Bibliografia

- [1] [url-stopmotion] Animação em Stop Motion. (data de último acesso 26-09-2010). [Online]. Available: <http://newronio.espm.br/index.php/2010/03/03/nessa-semana-historia-da-animacao/>
- [2] [url-expressoes] As Seis Expressões Faciais Primárias. (data de último acesso 26-09-2010). [Online]. Available: <http://cesarcalejon.wordpress.com/category/uncategorized/>
- [3] [url-bipeddog] Esqueleto Biped de um cão. (data de último acesso 26-09-2010). [Online]. Available: <http://rage.iameven.com/wp-content/gallery/roger/biped-dog.jpg>
- [4] [url-biped] Esqueleto Biped ajustado à malha poligonal do personagem virtual. (data de último acesso 26-09-2010). [Online]. Available: <http://www.darttinstitute.ca/arts-3D-behind-the-scenes.html>
- [5] [url-weightvertices] Representação de dois tipos de Weighting. (data de último acesso 26-09-2010). [Online]. Available: <http://www.pouet.net/topic.php?which=6110>
- [6] [url-weightenvolpes] Representação de dois tipos de Weighting. (data de último acesso 26-09-2010). [Online]. Available: <http://www.darttinstitute.ca/arts-3D-behind-the-scenes.html>
- [7] [url-facs] Action Units e respectivas interpretações. (data de último acesso 26-09-2010). [Online]. Available: <http://nullmodel.egloos.com/tag/FACS/page/1>
- [8] [url-motion] Motion Capture. (data de último acesso 26-09-2010). [Online]. Available: [http://frederatorblogs.com/channel\\$_\\$frederator/files/2008/04/jolie-beowulf-captours.jpg](http://frederatorblogs.com/channel$_$frederator/files/2008/04/jolie-beowulf-captours.jpg)
- [9] [url-avatar] Motion Capture. (data de último acesso 26-09-2010). [Online]. Available: <http://www.dickers.com/blog/wp-content/uploads/avatar-foto-dal-set11.jpg>

- [10] [url-morph] Utilização de Morph Target Animation no 3D Studio Max. (data de último acesso 26-09-2010). [Online]. Available: <http://www.scriptspot.com/3ds-max/scripts/tags/morpher>
- [11] A. Cláudio, L. Moniz, G. Gaspar, B. Carmo, R. Abreu, D. Policarpo, M. Lourenço, and N. Martins, “A Layered Approach to Animate Intelligent Characters, Actas do 17º Encontro Português de Computação Gráfica, Covilhã,” pp. 121–130, 2009.
- [12] [url-labmag] Página do LabMAg. (data de último acesso 26-09-2010). [Online]. Available: <http://labmag.di.fc.ul.pt/n/>
- [13] Mandler, G. *The Psychology of Facial Expression*. Cambridge University Press, 1997.
- [14] P. Ekman and W. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Palo Alto: Consulting Psychologists Press, 1978.
- [15] N. Magnenat-Thalmann and D. Thalmann, Eds., *Handbook Of Virtual Humans*. Chichester: John Wiley, 2004. [Online]. Available: <http://www.wiley.com/WileyCDA/WileyTitle/productCd-0470023163.html>
- [16] P. Ekman, “Facial Expression and Emotion.” *American Psychologist*, vol. 48, pp. 384–392, 1993.
- [17] E. Sifakis, I. Neverov, and R. Fedkiw, “Automatic Determination of Facial Muscle activations from sparse Motion Capture marker data,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 417–425, 2005.
- [18] E. Ju and J. Lee, *Expressive Facial Gestures From Motion Capture Data*. *Computer Graphics Forum*, 27, 2008.
- [19] T. D. Bui, M. Poel, D. Heylen, and A. Nijholt, “Automatic Face Morphing for Transferring Facial Animation,” pp. 19–24, 2003. [Online]. Available: <http://doc.utwente.nl/60388/>
- [20] D. Cosker, S. Roy, P. L. Rosin, and D. Marshall, “Re-Mapping Animation Parameters between multiple types of facial model,” in *MIRAGE’07: Proceedings of the 3rd international conference on Computer vision/computer graphics collaboration techniques*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 365–376.
- [21] I. Baran and J. Popović, “Automatic Rigging and Animation of 3D Characters,” p. 72, 2007.
- [22] S. Forstmann, J. Ohya, A. Krohn-Grimberghe, and R. McDougall, “Deformation Styles for Spline-Based Skeletal Animation,” pp. 141–150, 2007.

- [23] E. de Aguiar, C. Theobalt, S. Thrun, and H.-P. Seidel, “Automatic Conversion of Mesh Animations into Skeleton-based Animations,” *Computer Graphics Forum (Proc. Eurographics EG’08)*, vol. 27, no. 2, pp. xx–xx, 4 2008.
- [24] P. M. Semião, M. B. Carmo, and A. P. Cláudio, “Implementing Vision in the IViHumans Platform. In Ibero-American Symposium on Computer Graphics, SI-ACG 2006,” pp. 56–59, 2006.
- [25] P. M. Semião, , A. P. Cláudio, and M. B. Carmo, “Algoritmo de Visão para Humanos Virtuais. In Actas da 2a Conferência Nacional em Interação Pessoa-Máquina, Interação 2006,” pp. 133–138, 2006.
- [26] G. Larman, “Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design. Prentice-Hall,” 1998.
- [27] [url-kator] Site de modelos 3D. (data de último acesso 26-09-2010). [Online]. Available: [http://www.katorlegaz.com/3d\\$_models/](http://www.katorlegaz.com/3d$_models/)
- [28] [url-studio] Site de modelos 3D. (data de último acesso 26-09-2010). [Online]. Available: <http://www.the3dstudio.com/>
- [29] [url-axyz] Site de design de modelos humanos 3D animados. (data de último acesso 26-09-2010). [Online]. Available: <http://www.axyz-design.com/>
- [30] [url-facepro] Site de animação facial. (data de último acesso 26-09-2010). [Online]. Available: <http://www.facepro.net/>
- [31] [url-3dsmax] Site do 3D Studio Max. (data de último acesso 26-09-2010). [Online]. Available: <http://usa.autodesk.com/adsk/servlet/pc/index?id=13567410&siteID=123112>
- [32] [url-ogre3d] Site do OGRE - Open Source 3D Graphics Engine. (data de último acesso 26-09-2010). [Online]. Available: <http://www.ogre3d.org/>
- [33] [url-visualstudio] Site Microsoft Visual Studio. (data de último acesso 26-09-2010). [Online]. Available: <http://msdn.microsoft.com/en-us/vstudio/default.aspx>
- [34] [url-blender] Site do Blender. (data de último acesso 26-09-2010). [Online]. Available: <http://www.blender.org/download/get-blender/>
- [35] [url-ogremax] Site do exporter OgreMax. (data de último acesso 26-09-2010). [Online]. Available: <http://www.ogremax.com/downloads>
- [36] [url-ogreode] Site do OgreODE. (data de último acesso 26-09-2010). [Online]. Available: <http://www.ogre3d.org/tikiwiki/OgreODE>