Mestrado em Engenharia Informática
Dissertação/Estágio
Relatório Final

SUIWA

Social User Interface for Web Applications

Hugo Alexandre Neves Sousa
hasousa@student.dei.uc.pt

Orientador:
Prof. Daniel Augusto Gama de Castro Silva
Engª. Susana Filipa de Noronha Boavida Fernandes
Data: 9 de Julho de 2012

FCTUC DEPARTAMENTO
DE ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

"The future is indeed bright for computer facial animation"
- Frederik I. Parke & Keith Waters (2008)

Resumo

A utilização de personagens virtuais animadas em páginas web pode ter várias aplicações, desde a transmissão personalizada de notícias até ao esclarecimento de dúvidas relacionadas com produtos ou serviços.

A utilização de personagens tridimensionais e com aparência e movimentos realistas vem acrescentar valor a essas aplicações, levando o utilizador a dar mais atenção àquilo que é dito pelas mesmas.

No decorrer deste estágio foi desenvolvida uma aplicação web onde é possível personalizar uma personagem tridimensional. Esta personagem, depois de criada, tem a capacidade de "ler" texto e simular expressões faciais consistentes com algumas emoções principais.

Palavras-Chave Animação facial, expressões faciais, candide3, MPEG-4, HTML5, javascript, visemas, emoções

Aos meus pais.

Agradecimentos

Nesta fase final do meu percurso académico não posso deixar de agradecer a todos aqueles que me ajudaram de uma ou de outra forma a chegar aqui.

Ao Professor Daniel Silva, por todo o seu trabalho, disponibilidade e empenho no sucesso deste estágio.

A toda a equipa do Grupo Inogate S.A, por me acolherem e darem as condições necessárias para o sucesso deste estágio.

Á Engenheira Susana Boavida Fernandes, do Grupo Inogate S.A, pelo acompanhamento e orientação.

Ao Doutor Jörgen Ahlberg, da Universidade Linköping na Suécia, pela pronta disponibilização de documentos relacionados com a sua investigação.

Ao Doutor Maher Ben Moussa, do MIRALab (Universidade de Genève), pelo interesse e disponibilidade demonstrada ao longo deste estágio.

À minha família, por todos os valores, apoio e ajuda que sempre me transmitiram. Por acreditarem que é sempre possível chegar mais longe.

Aos meu amigos, pela amizade e apoio que ao longo dos anos me foram transmitindo.

Ao meu irmão, pelo apoio, carinho e confiança que me transmite.

À minha namorada, por toda a sua ajuda, compreensão e dedicação. Pela paciência, que ao longo destes anos precisou. Por estar sempre presente.

Índice

Lista de Figuras

Lista de Tabelas

Tabela de Acrónimos

2D Duas dimensões

3D Três dimensões

AJAX Asynchronous JavaScript and XML (JavaScript e XML assíncrono)

API Application Programming Interface

AU Action Units (Unidades de acção)

CSS3 Cascading Style Sheets v.3

FAP Facial Animation Parameter (Parâmetros de animação facial)

FAPU Facial Animation Parameter Unit (Unidades dos parâmetros de animação facial)

FFP Face Feature Point (Ponto de interesse facial)

FPS Fotogramas por segundo

GUI Graphical User Interface (Ambiente gráfico de interacção com o utilizador)

HTML5 HyperText Markup Language v.5

IDE Integrated Development Environment (Ambiente de desenvolvimento integrado)

IIS Internet Information Services

I/O Input/Output (Entrada/saída)

IA Inteligência Artificial

I&D Investigação e desenvolvimento

MVC Model View Controler

OCC Ortony, Clore & Collins

TTS Text-to-Speech

XML Extensible Markup Language

Capítulo 1

Introdução

Neste capítulo será apresentado o estágio, dando algum destaque aos seus objectivos. Será também apresentada a entidade de acolhimento e será feita uma breve descrição da estrutura deste documento.

1.1 Enquadramento

O trabalho de estágio descrito neste relatório surge inserido no Mestrado em Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra. Decorreu sob orientação do Professor Daniel Silva, professor no Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra, e da Engª Susana Boavida da entidade de acolhimento, Innabler, S.A.

Innabler S.A.

A Innabler S.A é uma empresa do grupo Inogate. A Inogate, empresa mãe do grupo, foi fundada em 2004 com capital privado exclusivamente nacional. Tem a sua sede em Coimbra, no Instituto Pedro Nunes, tendo desde 2005 uma delegação em Lisboa.

A principal área de negócio é consultoria em inovação embora recentemente se tenha assumido no mercado como fornecedor de soluções de software colaborativo. No seguimento da aposta no desenvolvimento de software surge o produto weWant2, que embora esteja ainda em fase de desenvolvimento já se encontra instalado em alguns clientes, dos quais se podem destacar a Agência de Modernização Administrativa (AMA) e a farmacêutica AstraZeneca.

Além da Innabler, faz parte do grupo Inogate a empresa Move Mile. Durante o ano de 2010 o Grupo Inogate foi certificado com a norma ISO 9001:2008-Sistema de Gestão da Qualidade e foi também reconhecida a sua idoneidade em matéria de I&D[1] nas áreas de inovação e de tecnologias da informação.

[1] Investigação e desenvolvimento

1.2 Objectivos

Este estágio está integrado num projecto de I&D do grupo Inogate que pretende desenvolver um assistente pessoal virtual inteligente, que tenha a capacidade de interagir a vários níveis com o utilizador. Embora o projecto de I&D tenha várias vertentes, entre as quais a semântica e a inteligência artificial, este estágio está essencialmente focado na sua vertente visual. Assim, e de forma simplificada, pretende-se que seja desenvolvido um interface visual para o assistente pessoal.

Este objectivo principal pode ser decomposto em vários objectivos de menor dimensão e mais concretos.

Numa primeira fase pretende-se que seja criado um módulo que possibilite a aplicação de uma foto bidimensional sobre um modelo tridimensional representativo da face presente na foto. Para cumprir esse objectivo é necessário que o utilizador possa carregar imagens e tenha a possibilidade de fazer o *morphing* de um modelo, de modo a que este represente com a máxima veracidade possível a face presente na foto. É também importante o desenvolvimento de funcionalidades de *save/load*, que permitam a utilização do mesmo modelo sem ter a necessidade de o definir constantemente.

O segundo sub-objectivo da aplicação está relacionado com desenvolvimento de um sistema de animação que replique os movimentos da boca e que permita a sincronização desse movimento com áudio, de forma a simular que o avatar está efectivamente a falar. A transformação de texto em áudio não é uma funcionalidade a ser implementada neste projecto. Para a conseguir será utilizada uma aplicação externa de TTS[2] que irá criar um ficheiro com a informação do áudio e outro com informação fonética presente em cada instante de tempo.

Um outro objectivo é a simulação de emoções. Com o desenvolvimento deste módulo pretende-se que o utilizador seja capaz de identificar o *"estado de espírito"* da personagem tridimensional enquanto interage com ela.

Uma vez que o avatar desenvolvido neste estágio faz parte de um projecto de maior dimensão, importa focar que nesta fase o projecto inclui a transformação de fotos 2D[3] em 3D[4] e a sua animação, seja ao nível da fala, seja ao nível das emoções. Foi também implementada uma forma de testar as funcionalidades pretendidas. Neste caso concreto está disponível um local para introdução de texto de forma a testar a animação da fala e também um conjunto de comandos que permitem testar as emoções.

Para a aplicação desta tecnologia em ambiente web, existe um conjunto de requisitos que devem ser tidos em conta. São eles: a qualidade visual, a facilidade de utilização, geração automática de conteúdo e a inteligência artificial[12].

Seguidamente descreve-se de uma forma sucinta o que se entende por cada um destes requisitos.

Qualidade visual

Qualidade visual não implica que os modelos sejam foto-realistas, implica sim que a animação seja o mais natural e atraente possível. Num modelo facial esta

[2]Text-to-Speech: transformação de texto em áudio
[3]Duas dimensões: largura e altura
[4]Três dimensões: largura, altura e profundidade

característica adquire especial importância porque, tal como refere Parke [13], o utilizador está muito familiarizado com o rosto humano e detecta facilmente movimentos ou expressões que não sejam naturais.

Facilidade de utilização

Aplicações de personagens virtuais não são o conteúdo principal da página web onde estão inseridos, são sim, uma 'atracção"que o utilizador pode ou não utilizar. Assim sendo, um utilizador, ao ser confrontado com a necessidade de instalar software adicional para utilizar esse bónus, pode ter dúvidas se o deve ou não fazer. Existe ainda a questão de algumas empresas não darem aos seus colaboradores a autonomia ou a autorização para instalar software nos seus postos de trabalho, o que dificulta a utilização de qualquer aplicação web que necessite da instalação de plugins de terceiros. Mesmo tendo essa possibilidade, a tarefa de instalação pode revelar-se mais ou menos complexa, consoante o sistema e os conhecimentos de quem o utiliza. Uma forma de suavizar essa barreira entre a tecnologia e o utilizador é tentar, sempre que possível, simplificar ou tornar transparente para o utilizador qualquer tarefa de instalação de suporte à aplicação.

Geração automática de conteúdo

Por geração automática de conteúdo entenda-se geração automática de discurso e correspondente animação facial. Esta é uma componente chave neste tipo de sistemas. Nenhum sistema de personagens virtuais se torna realmente interessante e apelativo se apenas tiver a capacidade de reproduzir sons e animações anteriormente definidas. Embora se torne um sistema claramente mais complexo, a sua complexidade é compensada pelas avançadas funcionalidades oferecidas. Um sistema que tenha a capacidade de processar texto escrito transformando-o em áudio e que seja também capaz de animar uma personagem sincronamente com a reprodução desse áudio é claramente mais apelativo que um sistema com respostas pré-definidas e que apenas abra e feche a boca, sem qualquer tipo de sincronismo ou aproximação à realidade humana.

Inteligência artificial

Embora não esteja directamente relacionada com este estágio, a componente de inteligência artificial é também de grande importância numa personagem virtual. Este sistema, quando anexado ao sistema de sintetização de fala, que posteriormente está ligado ao sistema de animação, tem a capacidade de transformar uma simples personagem virtual num verdadeiro assistente pessoal virtual. As possibilidades são imensas, desde a simples pesquisa na web até à análise da agenda para marcação de reuniões.

1.3 Motivação

Segundo Pandzic [12], personagens virtuais falantes são simulações gráficas de pessoas reais ou imaginárias capazes de simular alguns comportamentos humanos, sendo os mais importantes a fala e os gestos. Com o crescimento exponencial de

serviços na web e com a massificação do acesso a esses mesmos serviços a partir de dispositivos móveis, podem ser identificadas novas aplicações para essa tecnologia, como por exemplo a transmissão personalizada de noticias[14].

Segundo Pandzic [15] uma personagem falante pode ser mais persuasiva do que apenas texto. Adicionalmente também se podem considerar essas personagens como "user entertainers", na medida em que estas podem ser utilizadas de forma a aliviar a espera do utilizador por uma resposta do servidor ou pelo processamento de dados.

Pandzic [16] refere ainda que os utilizadores, quando confrontados com dois serviços (um deles com uma personagem virtual e o outro sem essa personagem) dão clara preferência àquele em que a personagem está presente em detrimento do outro.

Ao nível da empresa de acolhimento, o desenvolvimento de um assistente pessoal virtual teve como objectivo inicial a libertação do utilizador de tarefas de pesquisa sobre a utilização da aplicação em que este assistente vai ser embutido. Por exemplo, caso o utilizador pretenda saber como fazer determinada tarefa poderá inquirir o assistente, em linguagem natural, e este dar-lhe-á a resposta, não tendo o utilizador que efectuar a pesquisa.

Ao dar a possibilidade de criar um assistente pessoal com um aspecto ao gosto do utilizador pretende-se que a aplicação se torne mais atractiva e mais fácil de utilizar.

1.4 Organização do documento

Este trabalho está dividido em sete capítulos, sendo que este primeiro capítulo apresenta uma introdução ao trabalho realizado.

O capítulo seguinte apresenta uma ligeira introdução à anatomia humana e o estado da arte no que respeita à animação facial, sincronização labial e sintetização de emoções.

No terceiro capítulo são apresentadas e fundamentadas algumas das escolhas tecnológicas realizadas no projecto.

No quarto capítulo é apresentado de forma detalha o planeamento do estágio, dividido por semestres.

O capítulo cinco apresenta a arquitectura do sistema SUIWA segundo vários pontos de vista (cliente, programador e utilizador).

No capítulo seis está patente a apresentação de todo o trabalho realizado durante o primeiro e o segundo semestres respectivamente.

No capítulo final são feitas algumas considerações acerca do trabalho desenvolvido, e apresentadas propostas de desenvolvimento futuro para a aplicação em causa.

Capítulo 2

Estado da arte

Neste capítulo serão introduzidos alguns temas que têm especial relevância na animação facial, dos quais importa destacar a anatomia facial, como sendo a área de estudo principal para se conseguir uma animação passível de ser considerada realista.

O *mind map* apresentado na figura 2.1 representa o conjunto de áreas de interesse para a realização com sucesso deste projecto. Neste capítulo será feita uma passagem pelos três ramos não tecnológicos presentes no mind map, anatomia, processamento de fala e emoções . Por sua vez, no Capítulo 3, serão abordadas as áreas mais relevantes dos ramos ligados à tecnologia, como é o caso de *"Node.js"* e *"Modelação 3D"*.

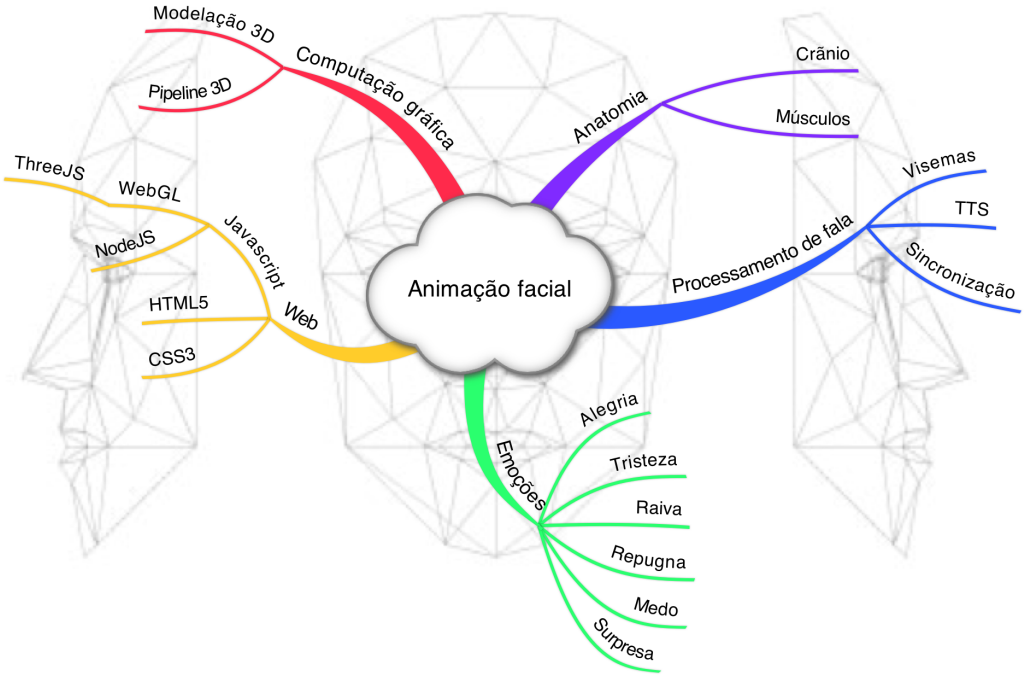

Figura 2.1: Mind map de referência ao projecto

2.1 Anatomia facial

O corpo humano tem sido alvo de intenso estudo ao longo dos séculos. No período decorrido entre os séculos XIII e XVII, denominado de Renascimento, o movimento artístico começou a dar extrema importância ao realismo na reprodução do corpo humano.

Leonardo da Vinci, por exemplo, para melhor compreender o corpo humano, dissecava cadáveres. Uma das suas obras mais conhecidas, *Homem Vitruviano*, vide figura 2.2, demonstra o interesse pela forma humana e pelo rigor na sua representação.

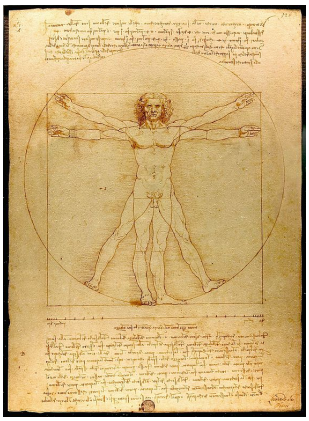

Figura 2.2: Homem Vitruviano.©Leonardo da Vinci

Um dos objectivos da animação é "modelar"faces que sejam realistas tanto estaticamente como em movimento animado. Para atingir esse objectivo, à semelhança dos artistas renascentistas, importa adquirir algum conhecimento em anatomia, mais concretamente, em anatomia facial.

Crânio

O crânio tem particular interesse para a sintetização de faces, uma vez que uma das suas funções é o suporte dos músculos e da pele. É também a base para o formato da face[2]. Do ponto de vista prático, o crânio é uma caixa protectora do cérebro[1]. Na figura 2.3 pode ser visto o crânio segundo várias perspectivas.

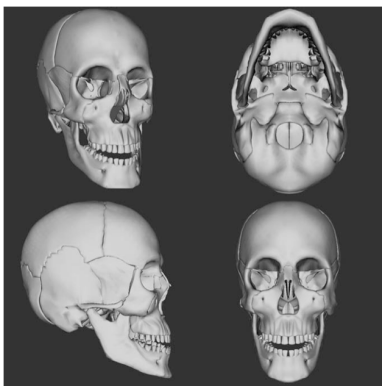

Figura 2.3: Quatro vistas do crânio[1]

O crânio pode ser dividido em duas partes, o crânio propriamente dito e o esqueleto facial, como se verifica na figura 2.4. Todos os ossos estão ligados entre si de forma fixa por uniões denominadas suturas, excepto a mandíbula, que está ligada por duas articulações.

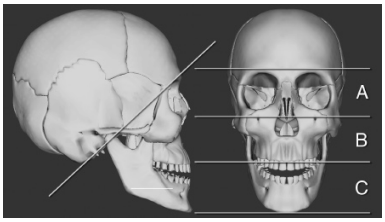

Figura 2.4: Divisão aproximada do crânio. Esquerda: divisão do crânio em duas partes. Direita:divisão do esqueleto facial em três partes[1]

O esqueleto facial pode também ser subdividido, como se pode ver na imagem 2.4 à direita. O primeiro terço é constituído pelas órbitas oculares e pelas fossas nasais. A parte central pelo maxilar e as cavidades nasais. No último terço apenas se encontra a mandíbula.

A parte do crânio em si é formada por um conjunto relativamente pequeno de ossos. São apenas oito, no entanto ao nível da dimensão estes podem ser considerados de tamanho grande. Seguindo uma linha da frente para trás, o nosso crânio é então formado pelo frontal, o esfenóide, dois parietais e, por último, o occipital. De cada um dos lados existe um temporal[2].

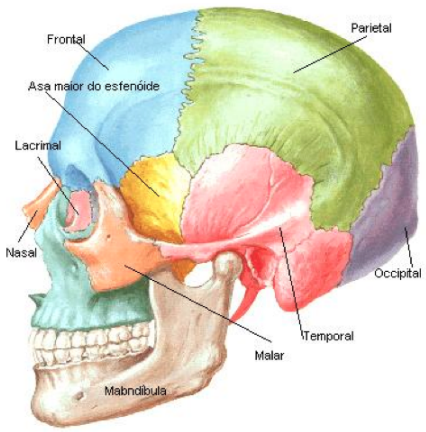

Figura 2.5: Vista lateral do crânio[2]

O osso frontal é grande e fino e forma a região da testa e a parte superior das cavidades oculares[17]. Os parietais estão ligados ao frontal e ao occipital e formam o que pode ser chamado de "telhado do crânio"[1]. O occipital está localizado na parte posterior do crânio. É por uma cavidade que tem, denominada *foramen magnum*, que a espinal medula passa para se ligar ao cérebro[17].

Os ossos do esqueleto facial são 14, agrupados aos pares, excepto no caso da mandíbula e do vomer. Esta característica confere à face uma aparência simétrica[2].

Um par de ossos que, apesar do seu tamanho, tem grande importância no formato da face são os ossos nasais. Têm formato convexo e estão situados na parte superior do esqueleto facial, entre as cavidades oculares[2].

Outro conjunto de ossos muito importante é o par de zigomáticos ou "ossos da bochecha"[1]. Estes ossos, em conjunto com o frontal e com o maxilar, são responsáveis pela definição das cavidades oculares[2].

O maxilar é o segundo maior osso do esqueleto facial, sendo apenas superado pela mandíbula. O formato deste osso influencia grandemente o formato e a dimensão de toda a face, devido à sua localização central. Este osso, além de servir de suporte aos dentes superiores, tem também uma importância acrescida por servir de âncora a muitos músculos faciais[2].

O maior osso do esqueleto facial, como já foi referido anteriormente, é a mandíbula, também commumente denominado maxilar inferior. É o mais forte e mais pesado de todos os ossos que fazem parte do esqueleto facial. Serve de suporte aos dentes inferiores e também de ponto de fixação dos músculos temporais[17].

Os dentes, embora não façam parte da estrutura óssea do crânio, são um elemento de extrema importância ao nível da animação, principalmente se um dos objectivos for a simulação de discurso. Os dentes, em conjunto com a língua e os lábios são os responsáveis por dar forma à boca para produzir o som pretendido.

Como já foi referido anteriormente, os dentes estão fixados quer ao maxilar quer à mandíbula. Nas crianças, como estes ossos são mais pequenos, os dentes são também eles mais pequenos e em menor número. Enquanto um adulto tem trinta e dois dentes, uma criança até aos cinco anos de idade tem apenas vinte. Estes dentes, normalmente entre os seis e os doze anos, vão sendo substituídos por dentição definitiva. Neste período nascem também os primeiro molares da dentição permanente sem que haja troca de dentes decíduos.

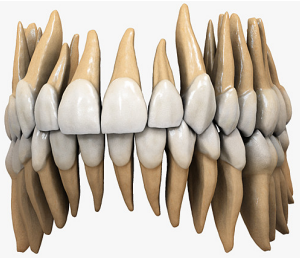

Figura 2.6: Dentição de aldulto[1]

Músculos

A grande maioria dos músculos do corpo humano está associada a alguma forma de movimento, devido ao facto de, em geral, estarem suspensos entre dois ossos ou órgãos. Contudo, alguns músculos faciais, em particular aqueles associados às expressões faciais, estão ligados ao osso numa das extremidades, mas na outra está ligada à pele.

A actividade muscular é estimulada por um ou mais nervos que causam a contracção dos músculos. Outra forma de estimular a actividade muscular é através de algum tipo de acção exterior, como na experiência de Duchenne(1862), em que os músculos faciais foram estimulados por choques eléctricos.

Embora os músculos faciais sejam muitas vezes associados a músculos das expressões faciais, eles desempenham outras tarefas, como por exemplo abrir e fechar os olhos ou mastigar.

Podemos dividir os músculos faciais em dois grupos: os músculos superficiais e os profundos. No que respeita às expressões faciais, estas são responsabilidade quase total dos músculos superficiais. Os músculos profundos estão mais relacionados com o movimento em si, como rodar a cabeça ou abrir a boca[17].

Posteriormente será feita uma pequena apresentação de alguns músculos considerados importantes.

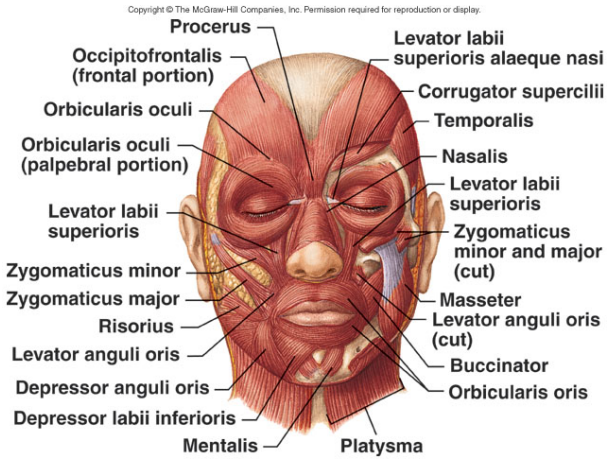

Figura 2.7: Representação dos músculos faciais[2]

Além desta divisão por macro-funções, os músculos estão categorizados segundo a orientação das suas fibras musculares. Dessa divisão surgem três grupos: os lineares ou paralelos, os elípticos ou circulares e os músculos em lençol.

Orbicularis oculi é um músculo circular que está localizado em redor dos olhos e é o responsável pela sua abertura e fecho[17].

Corrugator supercilii são um pequeno par de músculos piramidais localizados aproximadamente ao meio de cada sobrancelha. Este músculo puxa o centro das sobrancelhas, originando rugas verticais na testa[17].

Levator palpebrae superioris é um pequeno músculo, espalmado e triangular, cuja função é abrir e fechar a pálpebra superior[17].

Orbicularis oris é um músculo circular que circunda a boca. É responsável por uma quantidade muito grande de movimentos, desde auxiliar no processo de mastigação até às formas da boca quando se fala[17].

Zigomaticus major é um músculo que liga o comissura labial ao osso zigomático. Quando contraído puxa para o lado e para cima a comissura labial, dando origem a expressões de riso[17].

Messeter é um músculo forte que liga o osso zigomático à mandíbula. É utilizado no fecho da boca e tem uma especial utilidade na mastigação[17].

Lateral pterygoid é o músculo responsável pela abertura da mandíbula[17].

Embora não faça parte dos músculos faciais a língua é um músculo de extrema importância. É um músculo muito versátil visto que tem grande capacidade de movimento e de modificação do seu formato. Desempenha um papel de muito relevo na mastigação, deglutição e na fala[17].

2.2 Voz e processamento de fala

Na animação tradicional, o processo de sincronização labial é geralmente muito lento e envolve a análise do áudio de forma a identificar os tempos em que estão localizados eventos significativos que tenham que ser animados. Depois desta análise, são então desenhadas as *"key frames"* com a correspondente expressão facial.

Parke[18], apresentou um sistema informático de animação facial com sincronização labial. Neste trabalho, utilizou uma técnica denominada *"retoscoping"* de forma a obter uma sequência de parâmetros do movimento facial no momento em que um "actor"lia o texto.

Nos seres humanos a voz é criada pelo aumento e diminuição da pressão de ar na laringe e tracto vocal. Finalmente, sai pela boca ou pelo nariz[17].

O som propriamente dito é criado pela interacção do ar com as várias cavidades em conjunto com a vibração das cordas vocais. Consoante a posição dos vários órgãos intervenientes é criado um som diferente.

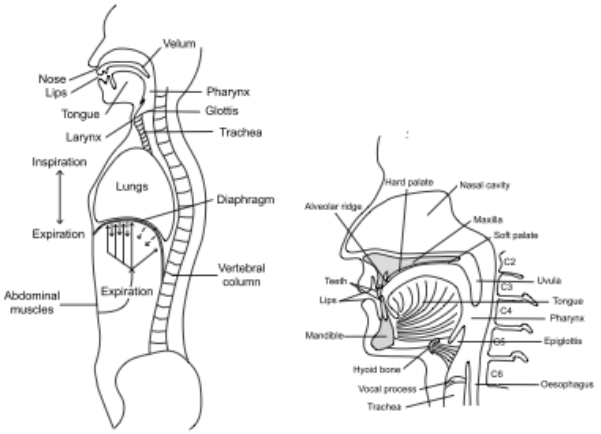

Figura 2.8: Anatomia do sistema de fala humano[1]

Em animação facial importa modelar e aproximar da realidade todos os aspectos visíveis da voz humana, assim, importa distinguir o que é visível do que não é. Ao analisar a figura 2.8, facilmente se conclui que grande parte do sistema de produção de voz está fora do alcance da visão humana, dentro da caixa torácica. Os únicos órgãos visíveis são a língua, os dentes e os lábios, pelo que é a estes que deve ser dada especial atenção.

Na figura 2.9 podemos verificar que, para cada tipo de som, existe uma posição específica dos três órgãos vistos anteriormente como importantes.

Dos três órgãos considerados mais importantes na fala, aquele ao qual deve ser dada mais relevância são os lábios, uma vez que é o órgão que está mais exposto e que, por sua vez, está mais sujeito à análise do utilizador. Movimentações erróneas ou falhas de sincronização são facilmente detectadas.

No que diz respeito aos dentes e à língua, já não existe uma preocupação tão grande, uma vez que estão posicionados dentro da cavidade bucal e são visíveis apenas em algumas ocasiões. No entanto, não se deve descurar a sua modelação, visto que são uma componente importante na aproximação do modelo à realidade e existem algumas expressões em que a sua falta é imediatamente notada.

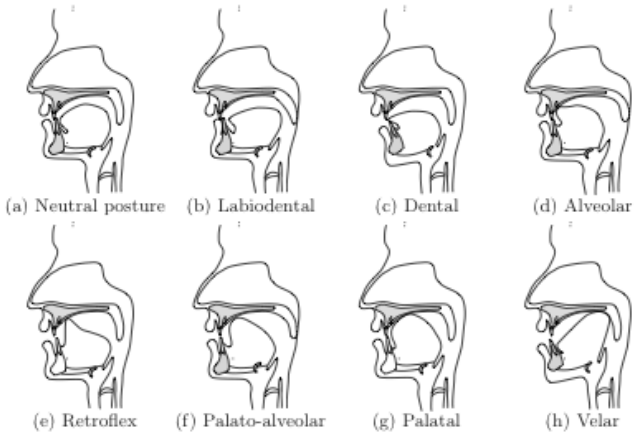

Figura 2.9: Posições base da fala humana[1]

2.3 Expressões faciais

A face humana é um forte meio de transmissão de mensagens, seja de forma explícita, através dos movimentos feitos enquanto falamos, por exemplo, ou de forma mais discreta, com um ligeiro levantar das sobrancelhas.

2.3.1 Emoções

A face humana é extremamente rica em informação acerca do estado emocional de uma pessoa, e despertou o interesse da ciência há já muito tempo. Desde o século XVII que se conhecem publicações que relacionam estados emocionais com reacções musculares(expressões faciais)[1].

Algumas teorias existentes actualmente foram inicialmente propostas por Charles Darwin, na sua obra *"The expression of the emotion in man and animals"* de 1904, e posteriormente revistas e actualizadas. A investigação de Darwin foi particularmente importante devido ao clima em que a ciência estava envolvida no decorrer do século XIX. Na época, era aceite que as expressões faciais, bem como outras características humanas, eram uma dádiva divina, e não uma reacção muscular que, segundo Darwin, tinha até algumas semelhanças com as de alguns animais[1].

Depois de observar cães, gatos, primatas e humanos, Darwin conseguiu provar as semelhanças entre as diversas espécies e assim fragilizar a teoria teológica[1].

Quase paralelamente à investigação de Darwin , outro cientista, de nome Guillaume Duchenne, realizou uma experiência que teve tanto de importante como de insólito. O impacto que esta teve não se deve apenas aos métodos ou aos resultados, mas também à forma de registo dos dados, que neste caso foram registados em fotografia.

Relativamente ao método experimental, este consistia em estimular, por meio de choques eléctricos, os diversos músculos faciais[3]. Com esta experiência, Duchenne conseguiu manipular e registar de forma controlada a expressividade de cada músculo ou grupo de músculos da face humana. Esta experiência teve uma importância tremenda no estudo das expressões faciais, porque, embora tivesse algumas inexactidões, foi com base nela que esta área de estudo foi realmente impulsionada.

Regressando ao conceito de emoção, este também foi alvo de bastante estudo no

Figura 2.10: Página da obra de Duchenne com algumas das fotos[3]

campo da psicologia[19]. Existem várias definições do conceito de emoção, segundo Cornelius estas podem ser organizadas em quatro grandes grupos: Darwiniana, Jamesiana, cognitiva e social construtivista[20].

A teoria Darwiniana defende que as emoções são fenómenos que evoluíram de instintos de sobrevivência. Isso significa que as emoções e as expressões deverão ser muito semelhantes em todos os humanos.

Para os Jamesianos, as emoções e expressões faciais estão intimamente ligadas, sendo que é impossível ter emoções sem expressões faciais e estas vêm sempre em primeiro lugar.

Na abordagem cognitiva a ideia é que o pensamento e as emoções são inseparáveis.

Finalmente, os construtivistas defendem que as emoções são produto da interacção social e que o seu sentido provém de regras sociais aprendidas.

Embora bastante distintas, todas as abordagens aceitam que as emoções podem ser vistas segundo um conjunto de dimensões[19].

O modelo OCC[4], apresentado na figura 2.11, define 22 emoções, divididas em três grupos: consequência de eventos, acções do agente e relação com objectos. Este modelo permite ainda a junção de grupos, sendo possível criar, por exemplo, um subgrupo de nome *consequência de eventos causados por acção de agentes*, tornando o modelo muito robusto. O modelo de Plutchik, mais simples, tem apenas 8 emoções, que podem ser combinadas de forma a criar estados emocionais mais complexos[5]. Como se pode verificar no círculo de emoções da figura 2.12 quanto mais perto do centro do círculo, mais intensa é a emoção.

Paul Ekman, defende um conjunto de emoções base ainda mais restrito, com apenas seis emoções: surpresa, medo, repugna, raiva, felicidade e tristeza[21]. Este modelo, devido à sua simplicidade, é amplamente utilizado[22][23][24][25][26] em sistemas de personagens virtuais. Existem até algumas ferramentas desenvolvidas com base neste conjunto de emoções, como é o caso do disco de emoções desenvolvido por Ruttkay *et al.* que tem como base as seis emoções de Ekman e um estudo de Scholsberg[27] que conclui que as expressões emocionais estão perceptualmente relacionadas, de tal forma, que é possível dispô-las num espaço bidimensional de forma circular, semelhante ao apresentado na figura 2.13.

Joy: (pleased about) a desirable event
Distress: (displeased about) an undesirable event
Happy-for: (pleased about) an event presumed to be desirable for someone else
Pity: (displeased about) an event presumed to be undesirable for someone else
Gloating: (pleased about) an event presumed to be undesirable for someone else
Resentment: (displeased about) an event presumed to be desirable for someone else
Hope: (pleased about) the prospect of a desirable event
Fear: (displeased about) the prospect of an undesirable event
Satisfaction: (pleased about) the confirmation of the prospect of a desirable event
Fears-confirmed: (displeased about) the confirmation of the prospect of an undesirable event
Relief: (pleased about) the disconfirmation of the prospect of an undesirable event
Disappointment: (displeased about) the disconfirmation of the prospect of a desirable event
Pride: (approving of) one's own praiseworthy action
Shame: (disapproving of) one's own blameworthy action
Admiration: (approving of) someone else's praiseworthy action
Reproach: (disapproving of) someone else's blameworthy action
Gratification: (approving of) one's own praiseworthy action and (being pleased about) the related desirable event
Remorse: (disapproving of) one's own blameworthy action and (being displeased about) the related undesirable event
Gratitude: (approving of) someone else's praiseworthy action and (being pleased about) the related desirable event
Anger: (disapproving of) someone else's blameworthy action and (being displeased about) the related undesirable event
Love: (liking) an appealing object
Hate: (disliking) an unappealing object

Figura 2.11: Lista de emoções definidas no modelo OCC[4]

Figura 2.12: Círculo de emoções do modelo de Putchik[5]

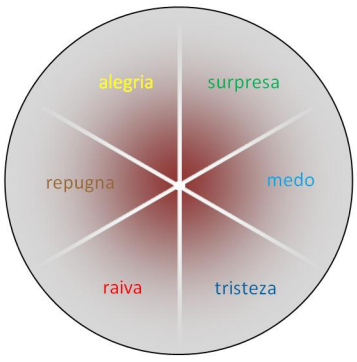

Figura 2.13: Círculo de emoções de Ruttkay [5]

Surpresa

A surpresa é a mais breve das emoções. É despoletada por um acontecimento inesperado e surge de forma súbita. Se houver tempo para pensar sobre o evento e

meditar sobre se está ou não surpreendido, então não está.

De uma forma muito simplista, esta emoção pode ser identificada pelas sobrancelhas levantadas, olhos muito abertos e mandíbula ligeiramente descaída, deixando a boca um pouco aberta, tal como se vê na figura 2.14[21].

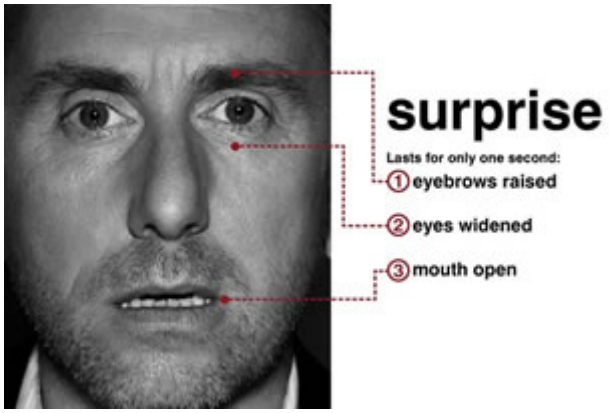

Figura 2.14: Surpresa. ©Lie to Me - Fox Networks

Medo

O medo pode ser físico, psicológico, ou ambos. Medo físico pode ser originado por uma simples vacina ou medo de situações onde exista risco de vida. Medo psicológico pode variar dos pequenos insultos à rejeição. Este último tipo de medo envolve prejuízo, por exemplo, da auto-estima ou confiança.

A reacção facial ao medo pode ser identificada pelas sobrancelhas levantadas e convergentes sobre o nariz, os olhos abertos e as pálpebras inferiores tensas e boca fechada e esticada no sentido das orelhas[21].

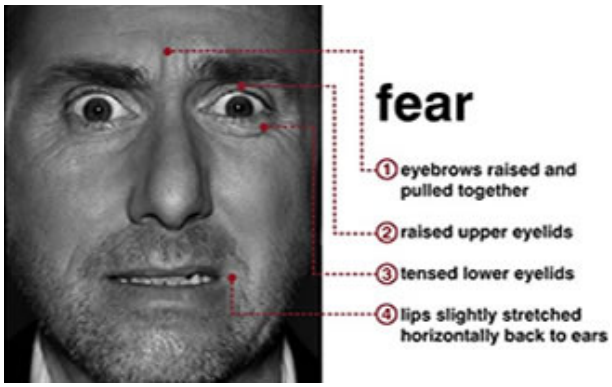

Figura 2.15: Medo©Lie to Me - Fox Networks

Repugna

A repugna é um sentimento de aversão. Podemos sentir repugna quando comemos algo que não gostamos. Algumas vezes, basta pensar em comer algo de que não se gosta para sentir repugna. Não é apenas pelo paladar que se sente repugna, o olfacto e o tacto são outros dos sentidos sensíveis à repugna.

Os sinais faciais mais importantes de repugna são o nariz enrugado e o lábio superior levantado, como se pode ver na figura 2.16. As sobrancelhas baixadas e pálpebras inferiores subidas, em conjunto com os anteriores, são também sinais de repugna[21].

Figura 2.16: Repugna©Lie to Me - Fox Networks

Raiva

A raiva é provavelmente a mais perigosa das emoções. Quando se experiencia a raiva, é provável que alguém seja magoado, seja o próprio ou terceiros. Parte da experiência de sentir raiva passa pelo risco de perder o controlo. Quando alguém está com raiva é normal fazer e dizer coisas que posteriormente de arrepende.

A raiva é um sentimento algo difícil de definir em termos de expressão facial, devido à grande variedade de expressões existente. Existem, no entanto, sinais que são claros e comuns: as sobrancelhas baixadas e juntas, em conjunto com as pálpebras tensas e os lábios ou fechados e um pouco tensos ou meio abertos numa forma algo quadrangular[21].

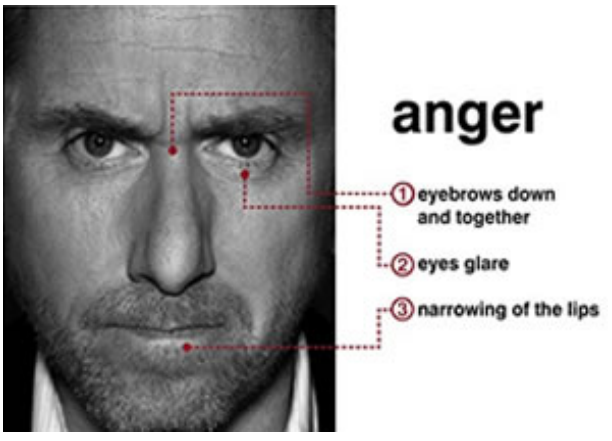

Figura 2.17: Raiva©Lie to Me - Fox Networks

Felicidade

Felicidade é aquela emoção que todos querem experimentar, é um sentimento positivo. Quando comparando com as restantes emoções apresentadas por Ekman

(1975), a felicidade é a única emoção positiva. O medo, a raiva, a repugna e a tristeza são todas elas emoções tristes. A surpresa é uma emoção neutra, dado que só pende para um dos lados devido ao contexto em que está inserida.

Detectamos felicidade numa cara quando os cantos da boca estão mais para trás e ligeiramente levantados. Uma característica importante é a ruga que vai desde o nariz até ao canto da boca. As bochechas estão ligeiramente levantadas e as pálpebras inferiores estão descontraídas e com algumas rugas[21].

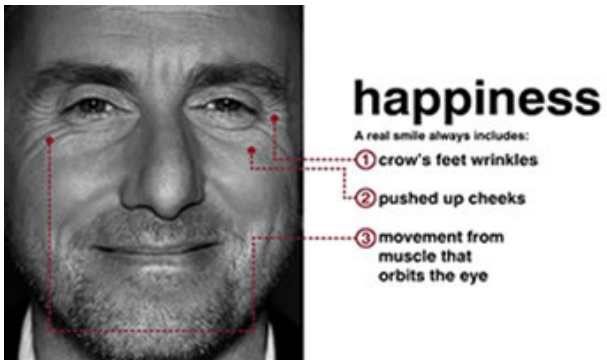

Figura 2.18: Felicidade©Lie to Me - Fox Networks

Tristeza

Tristeza é o sentimento do silêncio. Quando se está triste, chora-se discretamente e normalmente sozinho. Existem várias razões para experienciar este sentimento, como por exemplo o aparecimento de uma doença. Esta emoção geralmente é lenta, ou seja, quando se entra nela demora-se algum tempo até ultrapassar a situação.

Existem alguns pontos que se podem analisar para tentar saber se a pessoa está ou não triste. Um desses pontos são os cantos interiores das sobrancelhas levantados, o canto interior das pálpebras levantado e a pálpebra inferior levantada também. Os cantos da boca estão voltados para baixo ou os lábios podem tremer[21].

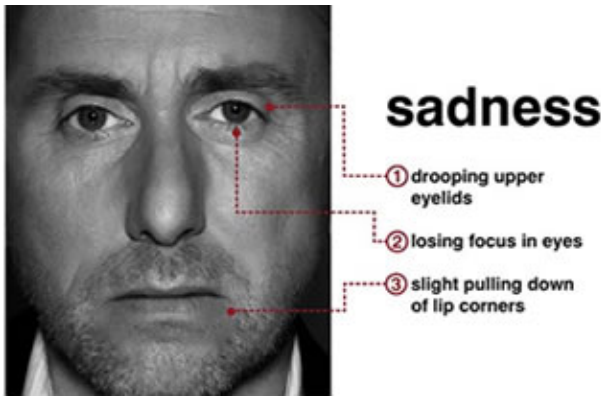

Figura 2.19: Tristeza©Lie to Me - Fox Networks

2.3.2 Visemas

O conceito de visema foi introduzido por Fisher em 1968[28], e refere-se à posição dos lábios, dentes e língua quando determinado som é produzido, ou seja, um visema

é a representação visual de um fonema[1]

Cada idioma tem o seu próprio subconjunto de fonemas, e consequentemente o seu próprio conjunto de visemas. Embora existam algumas semelhanças entre subconjuntos de cada idioma, existem também diferenças significativas. É relativamente fácil apercebermo-nos dessas diferenças, por exemplo, os falantes de línguas de base germânica apresentam grande dificuldade em reproduzir alguns fonemas de idiomas de base românica.

Por subconjunto de fonemas entende-se a quantidade de fonemas necessários para vocalização da totalidade das palavras do léxico de determinado idioma. Para português de Portugal esse subconjunto tem quarenta e três elementos, no entanto, no que diz respeito a visemas o conjunto é mais reduzido, bastando quinze para que seja possível animar com qualidade praticamente todas as palavras[9].

Cada fonema é representado no alfabeto fonético internacional[2] por um símbolo ou conjunto de símbolos. Com a crescente globalização dos computadores e como o conjunto de caracteres utilizado por eles não continha os símbolos presentes no IPA[3], John C. Wells, em 1995, decidiu definir um novo sistema de representação de fonemas constituído apenas por caracteres ASCII, que contivesse todos os símbolos do IPA. Assim apareceu o *X-SAMPA*[4]

Actualmente, esta representação está presente em muitas aplicações de TTS e é muito utilizada para representação de fonemas quer a nível académico quer a nível empresarial, tal como se pode verificar em [6] e [9].

Neto[9] apresenta a tabela 2.1, em que estabelece a relação entre fonemas e visemas. Esta tabela é de grande importância no processo de animação facial, uma vez que algumas aplicações de TTS geram, em conjunto com o ficheiro áudio, um ficheiro com os fonemas *X-SAMPA* correspondentes. Como não existe uma correspondência directa entre fonemas e visemas, esta tabela é necessária para efeitos de tradução.

Tabela 2.1: Relação entre fonemas e visemas para Português de Portugal[9]

Visemas	**Fonemas**	**Visemas**	**Fonemas**
sil	sil	*t*	t, d, n
@	@	*S*	S, Z
f	f, v	*a*	a, 6, 6 , 6 j , 6 w
g	k, g, L, J	*e*	e, E, e
l	l, I , R, r	*i*	i, i , j, j
O	O	*o*	o, o , o j
P	p, b, m	*u*	u, u , w, w , u j
s	s, z		

Na figura 2.20 são apresentados os visemas para Português de Portugal. Ao analisar essa figura verifica-se que o mesmo visema pode ser utilizado para representar

[1] Segundo o *Dicionário Universal da Língua Portuguesa* fonema é qualquer som elementar da linguagem articulada ou a unidade mínima do sistema fonológico de uma língua.

[2] `http://www.langsci.ucl.ac.uk/ipa/IPA_chart_(C)2005.pdf`

[3] *International phonetic alphabet* - Alfabeto fonético internacional

[4] Ver tabela que relaciona IPA com X-SAMPA no apêndice A.

vários sons. Em contexto de animação, os visemas são utilizados como *"key-frames"*, sendo por isso necessário definir as frames intermédias de forma a que exista realmente animação.

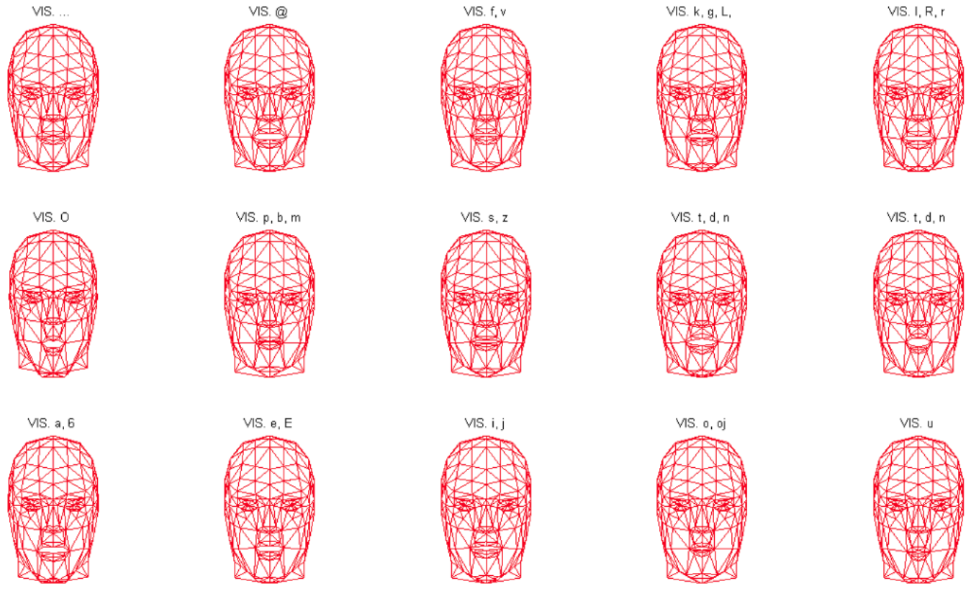

Figura 2.20: Visemas para Português de Portugal[6]

Para Português de Portugal[9] define os seguintes visemas:

- *@*

  O visema "@" é um mapeamento directo do fonema "@". Este fonema está presente em palavras como 'doc**e**' ou 'vint**e**'.

- *f*

  Este visema aparece como resultado do mapeamento dos fonemas 'f' e 'v'. Embora tenham sonoridade diferente, do ponto de vista visual estes fonemas são muito semelhantes. Por exemplo nas palavras '**f**alar' e'**v**erde' a posição labial inicial é práticamente igual.

- *g*

  O visema "g"será apresentado numa gama vasta de casos. Isto porque este visema traduz os fonemas 'k', 'g', 'L' e 'J'. Algumas palavras onde se pode verificar a ocorrência desse visema é respectivamente em '**c**om', '**g**rande', 'traba**lh**o' e 'vi**nh**o'.

- *l*

  À semelhança do visema anterior, também este representa um conjunto de fonemas que entre si têm bastantes semelhanças ao nível visual. Os fonemas representados pelo visema "l"são: 'l', 'l ', 'R' e 'r'. O primeiro corresponde por exemplo à palavra '**l**anche'. O segundo, embora seja muito semelhante a 'i ' apresenta uma pronúncia mais aberta. Os restantes correspondem respectivamente a '**r**ua' e 'ca**r**o'.

- *O*

  O visema "O"corresponde apenas ao fonema 'O' e está presente por exemplo em '**O**ntem'. Embora visualmente seja semelhante a outros visemas, por exemplo 'o', não é possível descartar um deles, porque as diferenças, embora discretas são perceptíveis pelo utilizador.

- *P*

  Este é talvez o visema mais simples de associar a fonema, visto que em algumas pronúncias existentes em Portugal já se torna difícil distinguir entre esses fonemas. São eles o 'p', 'b' e 'm' que estão presentes em palavras como '**p**ar, '**b**ar e '**m**ar.

- *s*

  O visema "s" representa os fonemas 's' e ´z'. Esta associação é também , à semelhança da anterior, uma associação lógica, visto que existem algumas pronúncias que utilizam tanto um som como o outro para a mesma palavra. As palavras '**c**éu' e 'ca**s**a' representam respectivamente os sons 's' e 'z'.

- *t*

  Da mesma forma que o visema 'p' representa os fonema 'p', 'b' e 'm', também o visema 't' representa os fonemas 't', 'd' e 'n'. Nas palavras '**t**enho', '**d**oce' e '**n**ada' é perceptível a semelhança visual entre esses sons.

- *S*

  As palavras '**ch**apéu' e '**j**óia', embora sejam completamente diferentes do ponto de vista fonético, apresentam muitas semelhanças na sua componente visual. Assim, o visema "S" representará os fonemas ´S' e 'Z'.

- *a*

  O visema "a" é um dos que traduz mais fonemas. Estes fonemas incluem também alguns ditongos com sonoridade específica, como aquele presente na palavra 't**ê**m'. O conjunto completo de fonemas traduzidos por "a" é: 'a', '6', '6 ', '6 j ' e '6 w '. Existem diversas palavras que exemplificam os diversos fonemas e nas quais é possível identificar as semelhanças visuais, por exemplo 'f**a**lo', 'c**a**ma', 'irm**ã**' ou 't**ê**m'.

- *e*

  A segunda vogal do alfabeto tem várias leituras possíveis, consoante o contexto em que é utilizada. O visema"e" representa alguns desses contextos, representando os fonemas 'e', 'E' e 'e '. O fonema 'e' pode ser encontrado na palavra 'faz**e**r', por sua vez, o fonema 'E' está presente em 'b**e**lo'. Na palavra 'empr**e**go' pode ser encontrado o fonema 'e '.

- *i*

  Os fonemas 'i', 'i ', 'j' e 'j ' são representados pelo visema "i". Este visema está presente em todas as palavras que contenham sons semelhantes aos existentes em 's**i**m' ou 'f**i**o'.

- *o*

  No que diz respeito ao visema "o", este também engloba vários fonemas. Uma das particularidades deste visema é que apenas representa fonemas mais fechados, como 'o' ou 'o ' ou o ditongo 'o j ', presentes em palavras como 'l**o**bo' ou 'cart**õe**s'.

- *u*

  Por último, o visema "u", que está presente em palavras como 'j**u**s', '**um**', ou 'e**u**'. Este visema representa os fonemas 'u', 'u ', 'w', 'w ' e 'u j '.

2.4 Animação facial

"The human face is a challenge for computer animation for at least two reasons. First the face is not a rigid structure but is a complex flexible surface. (...) Secondly faces are very familiar to us, we have a well developed sense of what expressions and motions are natural for a face". Parke(1972).

2.4.1 Um pouco de História

A animação facial computorizada é alvo de interesse dos cientistas há cerca de meio século. As primeiras imagens tridimensionais de faces foram geradas por Parke em 1971[1]. Os modelos e a animação eram muito simples, mas deram origem a um novo ramo de investigação, dedicado à representação e animação facial. Logo no ano seguinte Parke apresentou um novo modelo, desta feita já mais suave e realista. Para o conseguir recorreu a sombras, que em determinados casos ainda disfarçavam mais as arestas dos polígonos. Os dados das expressões faciais que iriam ser utilizados na animação foram recolhidos a partir da cara, pintada com linhas de orientação, do seu assistente[13].

Em 1974, na sua tese de Doutoramento, Parke volta a apostar na animação facial, e apresentou o primeiro modelo facial tridimensional parametrizado. No modelo apresentado, era possível definir uma grande quantidade de variáveis, adaptando o modelo a diversos tipos de face[18]. Nos anos que se seguiram, a animação facial não sofreu desenvolvimentos significativos, tendo vindo a ressurgir novamente no início dos anos 80 com uma abordagem baseada em músculos proposta por Platt[29].

Depois deste pequeno arranque, a animação facial voltou a despertar interesse tanto de académicos como de alguma indústria de animação, e em 1985 surge a primeira curta metragem, *Tony de Peltrie*, produzida por Bergeron e Lachapelle, onde a animação facial tridimensional foi uma parte fundamental.

Já no final da década de 80, Keith Waters apresentou uma nova abordagem para animação facial. O autor juntou o modelo parametrizado de Parke com o modelo muscular de Pratt. Desta junção surgiu um modelo muscular muito mais versátil e que se apresentava mais realista que os seus antecessores[22].

No início dos anos 90, outra curta metragem ficou famosa devido à utilização de técnicas de animação facial tridimensional. O filme *Tin Toy* da Pixar, que foi galardoado com um Óscar da Academia de Artes e Ciências Cinematográficas, recorreu a um modelo muscular de forma a animar o bebé[1].

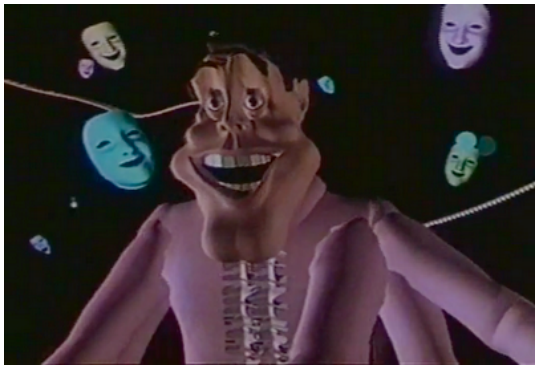

Figura 2.21: Tony de Peltrie©Centre de calcul de l'Université de Montréal

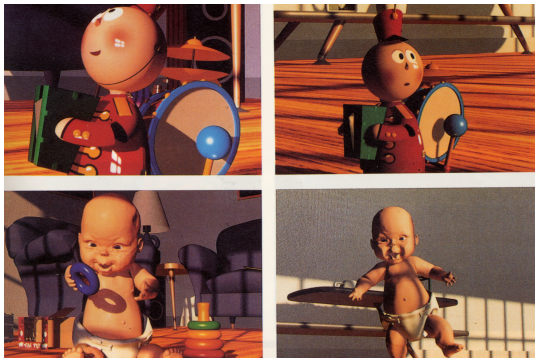

Figura 2.22: Tin Toy©Pixar

No decorer dos anos 90, devido ao aparecimento de diversos scanners 3D, a aquisição de dados de animação sofreu um avanço extraordinário, possibilitando dessa forma o evoluir das técnicas e da qualidade das animações faciais.

Já final do século foi também apresentada a norma ISO/IEC 14496-1(MPEG-4 para animação facial), cujo objectivo era, além de definir um conjunto de ferramentas, aliar as duas principais comunidades envolvidas na animação facial, os cientistas e os artistas[10].

Em 2001, Jörgen Ahlberg apresentou a terceira versão de um modelo facial, que nas versões anteriores tinha sido muito utilizado por académicos. Esse modelo era o escolhido devido à sua simplicidade. A terceira versão deste modelo mantém o baixo número de vértices e com a actualização, passa a ser compatível com a norma MPEG-4[8].

Nos primeiros anos do século XXI, muitas foram as aplicações de técnicas de animação facial 3D, das quais se podem destacar os filmes de animação ou os jogos, cada vez mais realistas, quer em termos de imagem quer em termos de movimento.

2.4.2 Olhar

Olhos dinâmicos e com algum detalhe são muito importantes para o sucesso na animação de expressões faciais. Eventos como seguir algo com o olhar ou até dilatação pupilar são dois exemplos de animação que contribuem para aumentar o realismo da animação[1].

Albrecht *et al.*(2002) apresentam uma proposta de utilização de piscar de olhos como sinais de pontuação. Ou seja, os olhos piscam quando há pausas no discurso.

Nessa mesma proposta, defendem a adição de mais acções de piscar de olhos, de forma a simular o que acontece com os humanos, que piscam os olhos com um período de aproximadamente 4,8 segundos, de forma a manter a humidade da córnea.

A direcção do olhar, tal como já foi referido, acrescenta valor emocional à animação. Os humanos quando confrontados com uma pergunta, apenas assumem que esta lhes é dirigida se estiverem sozinhos com o interrogador ou caso este lhe dirija directamente o olhar[30].

Outro exemplo muito simples está relacionado com os tempos mortos, ou seja, alturas em que não se está a olhar para nada em concreto. Nessa situação os olhos não permanecem numa posição estacionária, têm movimentos rápidos para diversos pontos do ambiente circundante[31]. Estes exemplos são prova do valor acrescentado que os movimentos oculares involuntários adicionam à animação facial.

2.4.3 Personificação: quais os seus efeitos?

À medida que a tecnologia e a técnica evoluem, o realismo das animações acompanha essa evolução, seja em qualidade de imagem seja na precisão e suavidade dos movimentos. Essa evolução induz diferentes reacções, algumas de apoio outras de oposição. Se por um lado há quem veja o desenvolvimento deste tipo de sistemas como uma evolução, há também os mais cépticos, para os quais qualquer tipo de evolução é sempre alvo de desconfiança.

Koda & Maes[32] realizaram um teste de forma a avaliar a reacção dos utilizadores a um sistema onde estaria presente uma personagem virtual. Este teste permitiu identificar de forma clara os dois grandes grupos referidos anteriormente, e dentro desse várias reacções distintas ao sistema testado. No entanto uma das grandes conclusões deste teste foi que a existência de uma personagem virtual é benéfica, no sentido em que ajuda o utilizador a focar-se numa determinada tarefa.

A origem do foco do utilizador na personagem, segundo Koda & Maes[32] está na tentativa do utilizador em interpretar a personagem, e desta forma manter a atenção naquilo que a personagem possa estar a transmitir.

Parke & Waters[1] introduzem o conceito de *"Believability Flip"*, que se refere ao ponto em que a percepção de realismo do utilizador em relação à personagem virtual decresce de forma abrupta, ou seja, passa de um valor extremamente alto para um extremamente baixo, tal como se pode verificar na figura 2.23. Este fenómeno

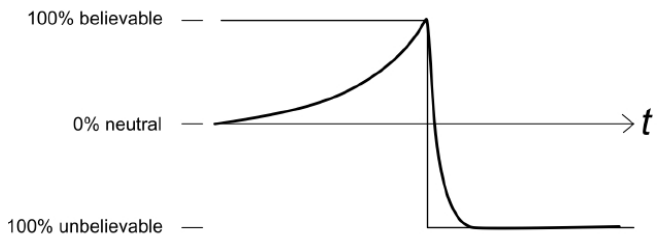

Figura 2.23: Believability Flip[1]

ocorre quando o utilizador, tendo uma imagem bastante positiva da personagem virtual, se apercebe de alguma falha ou incoerência, seja ela ao nível da aparência,

da falta de fluidez dos movimentos ou de falhas de sincronismo entre a animação e o discurso.

Quando este *"flip"* acontece é irreversível. Tudo aquilo que anteriormente se conseguia obter através da utilização da personagem, depois do *"flip"* deixa de ser possível. O utilizador, a partir desse momento deixa de responder aos estímulos da personagem.

2.5 Áreas de aplicação

O quantidade de aplicações possíveis para a animação facial é tão grande e distinta quanto as actividades humanas.

A indústria da animação é sem dúvida o maior produtor e também consumidor de animação facial tridimensional, seguida de perto pela indústria dos jogos. Nesta última, tanto os objectivos como as técnicas utilizadas são bastante diferentes. Hoje em dia, devido ao aumento crescente da capacidade dos CPU e dos GPU, é possível criar, em tempo real, animação de alta definição[1].

No entanto existem outras áreas, como a medicina, onde esta tecnologia tem uma aplicação crescente. Segundo Parke & Waters[1], a animação facial no contexto da medicina tem duas áreas de especial interesse: o planeamento cirúrgico e a simulação cirúrgica de tecido facial. Em ambos os casos, o objectivo é simular antes de submeter o paciente à intervenção cirúrgica.

Um dos exemplos em que seria útil a simulação é o planeamento de uma intervenção cirúrgica craniofacial. Este procedimento envolve o reposicionamento de ossos do crânio, o que pode ser comparado a uma complexa operação de cortar e colar tridimensional[33]. No que à simulação de tecido facial diz respeito, o objectivo é diferente. Nesta pretende-se simular a resposta da pele e do músculo depois de cortados, retirados ou refeitos[34].

Ainda no contexto da medicina, mas numa vertente completamente diferente, Orvalho *et.al* [35] apresentam um sistema que ajuda doentes de autismo no reconhecimento de emoções. Este sistema tira partido da animação facial para simular emoções de forma dinâmica e muito realista. O objectivo é estimular a aprendizagem por imitação. A figura 2.24 apresenta algumas personagens utilizadas nessa aplicação.

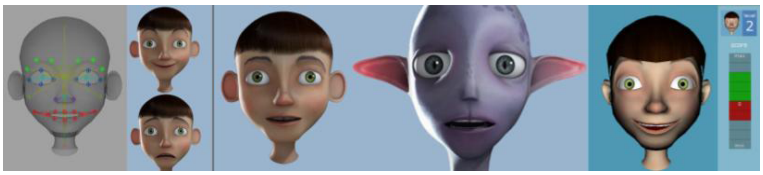

Figura 2.24: Exemplo aplicação Face in Motion©Face in Motion

Parke & Waters[1] identificam ainda outro grupo de aplicações, denominado *"Social agents and Avatars"*, que faz uso da animação facial. Neste grupo entram todas as aplicações em que o agente animado está presente, não sendo no entanto o foco principal da aplicação.

Existe uma vasta gama de aplicações onde esta tecnologia pode ser aplicada, como se pode verificar em [1] [12] [15] [36] [37]. De entre as várias propostas apresentadas nas obras citadas, podem ser salientadas as aplicações de ajuda à

navegação, professores virtuais e interfaces humano-computador avançadas que permitem a compreensão e produção de discurso. Parke & Waters referem-se a este tipo de agentes como *"social user interfaces"*, na medida em que são uma interface com o utilizador que permitem um alto nível de interacção[1].

É neste grupo de *"social user interfaces"* que se enquadra o agente a ser desenvolvido no presente estágio.

2.6 Análise de soluções relacionadas

A animação facial, em especial, a animação tridimensional de faces é uma das áreas de mais relevo na actual indústria dos jogos. Em aplicações web, esta tecnologia não tem ainda especial relevo, no entanto existem algumas soluções que se apresentam com muito qualidade.

Em relação a aplicações potencialmente concorrentes do SUIWA podemos referir o *3D Photo Face*[5] e o *Sitepal*[6]. Ambas as soluções apresentam as funcionalidades consideradas essenciais para um avatar tridimensional para a web, no entanto, e sendo essas duas soluções já comercializadas, ambas disponibilizam uma API, coisa que a aplicação SUIWA neste fase não tem presente. No entanto, ao contrário do SUIWA, as duas soluções apresentadas anteriormente requerem ao utilizador a instalação de um *plugin* de reprodução de aplicações *flash*.

Existem ainda outras aplicações que podem ser consideradas semelhantes, mas que no entanto não têm todos os requisitos necessários para se apresentarem como concorrentes a qualquer uma das soluções referidas anteriormente. Nesta categoria podem ser referidas a *Chloe*[7], que além de não ter uma aplicação de TTS a vocalizar a resposta não tem também qualquer tipo de animação, neste tipo de aplicações pode ser incluída também a *TIA*[8], que além de ter algum movimento ocasional, também não apresenta qualquer tipo de vocalização ou animação associada à resposta.

A tabela 2.2 apresenta a síntese de alguns parâmetros analisados nas diversas soluções. A tabela apresenta duas escalas de valores, consoante o parâmetro a que se refere. No caso da necessidade de instalação de *plugins* e na presença capacidades de TTS é utilizada uma escala binária de resposta "sim" ou "não", nos restantes casos a escala é mais complexa, podendo conter um dos seguintes valores: "muito fraco", "fraco", "razoável", "bom", "muito bom".

A nível visual, e como se pode verificar pelas figuras 2.25 e 2.26, a solução SUIWA e Oddcast apresentam resultados muito semelhantes, por sua vez a solução Sitepal apresenta um resultado menos satisfatório, visto que não utiliza como textura a imagem carregada, cria ela própria uma textura baseada na imagem fornecida pelo utilizador.

[5]Mais informação disponível em `http://www.oddcast.com`
[6]Mais informação disponível em `http://www.sitepal.com`
[7]Mais informação disponível em `http://www.virtuoz.com`
[8]Mais informação disponível em `http://www.telekom.si/tia`

Tabela 2.2: Tabela comparativa das várias soluções de animação facial para a web

	Qualidade Visual[a]	**Plugins**	**TTS**	**Emoções**[b]	**Discurso**[c]	**Movimentos da cabeça**
Oddcast	Boa	Sim	Não	Bom	Razoável	Bom
Chloe	Muito fraco	Não	Não	Muito fraco	Muito fraco	Muito fraco
TIA	Razoável	Não	Não	Muito fraco	Muito fraco	Bom
Sitepal	Razoável	Sim	Sim	Bom	Razoável	Bom
SUIWA	Boa	Não	Sim	Razoável	Bom	Bom

[a] A avaliação da qualidade visual foi realizada pelo autor e baseada na semelhança visual do modelo final em relação à foto de origem

. [b] A avaliação das emoções foi realizada pelo autor e teve como base a análise e comparação das emoções com imagens de referência presentes em [21]

. [c] A avaliação do realismo no discurso foi realizado pelo autor em conjunto com outros elementos da equipa de desenvolvimento e foi feito comparando as soluções entre si e com um conjunto de visemas de referência definidos por [6]

.

Figura 2.25: Imagem original

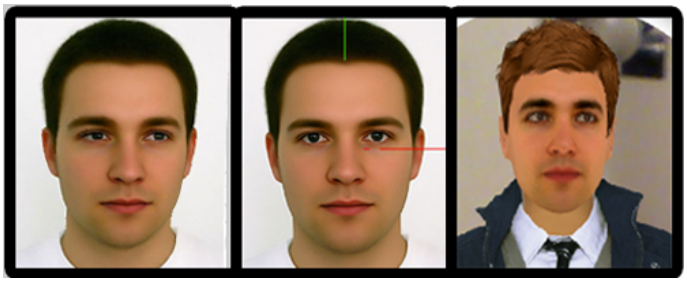

Figura 2.26: a)Oddcast©Oddcast b)SUIWA c)Sitepal©Sitepal

Capítulo 3

Opções tecnologicas

Neste capítulo serão apresentadas e explicadas as principais opções tomadas neste projecto. Desde a utilização do modelo genérico *Candide-3*, passando pelo standard MPEG-4 para animação facial até à utilização de Node.js como plataforma de desenvolvimento.

3.1 Standard MPEG-4, ISO/IEC 14496-2

Ao propor, em 1978, um sistema de parametrização, Paul Ekman iniciou a definição de um standard para a animação facial[11].

Em 1999 foi então definido o standard MPEG-4, que propunha a deformação de um modelo facial de forma directa, actuando sobre pontos de referência. Esta especificação foi o primeiro sistema de controlo facial parametrizado a ser definido como standard[10]. Com esta especificação surgiu uma nova técnica de animação facial que, por ser mais económica em termos de comunicação, se mostra mais adequada para aplicações em rede[19].

Esta norma tem em conta também a sincronização da animação com outros tipos de *media*, tais como stream de áudio ou integração com TTS. Em ambos os casos esta sincronização é feita com base em *time stamps*. Esta característica tem especial relevância no contexto deste projecto, uma vez que a sincronização da animação com o TTS é um dos objectivos da aplicação SUIWA.

A utilização de *Face Animation Parametrs*(FAPs) permite a animação de qualquer modelo compatível, no mesmo cliente MPEG-4[10]. FAPs[1] são então um conjunto de 68 parâmetros, divididos em 10 grupos, cada um deles relacionado com uma parte da face[38], tal como na tabela 3.1, que representam um conjunto de acções faciais básicas tais como o movimento da cabeça ou o controlo da boca, olhos ou língua.

O primeiro passo na definição do standard MPEG-4 é a definição de um modelo facial no seu estado neutro, como se mostra na figura 3.1. A definição de um estado neutro implica que a face esteja posicionada de tal forma que a partir daí seja possível efectuar transformações que resultem em qualquer das expressão faciais pretendidas.

Desta forma, este standard assume que na face neutra o olhar está direccionado

[1] Face animation parameter.

Tabela 3.1: Grupos de FAPs[10]

	Grupo	Número de FAPs
1.	Visemas e expressões	2
2.	Maxilar, queixo, interior dos lábios, cantos dos lábios	16
3.	Olhos, pupilas e pálpebras	12
4.	Sobrancelhas	8
5.	Bochecha	4
6.	Língua	5
7.	Rotação da cabeça	3
8.	Exterior dos lábios	10
9.	Nariz	4
10.	Orelhas	4

no eixo Z, todos os músculos faciais estão relaxados, as pálpebras estão tangentes à íris, as pupilas têm um terço do diâmetro da íris e a boca está fechada e sem qualquer expressão, os dentes superiores e inferiores estão em contacto e a língua está na horizontal, com a ponta em contacto com a linha de contacto dos dentes.[10].

A figura 3.1, além de representar a face no seu estado neutro, tem também a representação gráfica das distâncias a serem utilizadas para calcular as FAPUs[2].

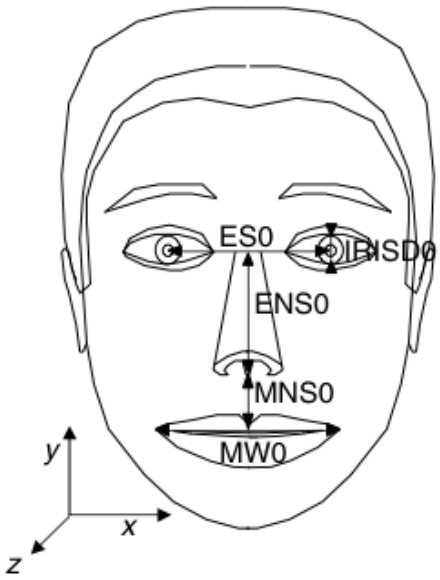

Figura 3.1: Modelo facial no estado neutro[7]

As FAPUs são definidas como fracções das distâncias definidas entre pontos faciais chave presentes na figura 3.1, e têm como principal objectivo a interpretação e a adaptabilidade das FAPs a qualquer modelo compatível.

Cada FAP é definida em ordem à FAPU correspondente, que, por sua vez, é calculada no modelo em utilização através da distância entre os pontos chave correspondentes. Dessa forma é possível garantir que a mesma FAP tem exactamente

[2]Face animation parameter units.

o mesmo efeito em qualquer modelo, desde que esse modelo seja compatível com a standard.

A distância entre os pontos chave, tal como já foi referido, é calculada de forma distinta em cada modelo, no entanto, a transformação dessas medidas em FAPUs é comum a todos os modelos, e é feita de acordo com a tabela 3.2. Na primeira coluna é referida a medida a utilizar no cálculo, a segunda coluna tem uma pequena descrição da medida em causa e na última coluna está o cálculo em si.

Tabela 3.2: FAPUs[10]

IRISD0	Diâmetro da iris (por definição é igual à distância entre a pálpebra superior e inferior)	IRISD=IRISD0/1024
ES0	Distância entre olhos	ES=ES0/1024
ENS0	Distância entre olhos e nariz	ENS=ENS0/1024
MNS0	Distância entre boca e nariz	MNS=MNS0/1024
MW0	Largura da boca	MW=MW/1024
AU	Unidade angular	$10e^{-5}$ radianos

Sendo um standard, o MPEG-4 não aplica as transformações a pontos com localizações específicos, aplica sim essas transformações a pontos virtuais, que cada modelo deve ter mapeados para que seja considerado *"de acordo com"* o standard. No total, o standard define 84 pontos virtuais, denominados FFPs[3], na face neutra. O seu objectivo é providenciar uma orientação espacial para a definição das FAPs. Alguns FFPs, como as que definem o cabelo, não são afectadas pelas FAPs, no entanto são necessárias de forma a ser possível a utilização com um conjunto mais vasto de modelos. As figuras 3.2 e 3.3 mostram a localização de cada um dos FFPs definido no standard.

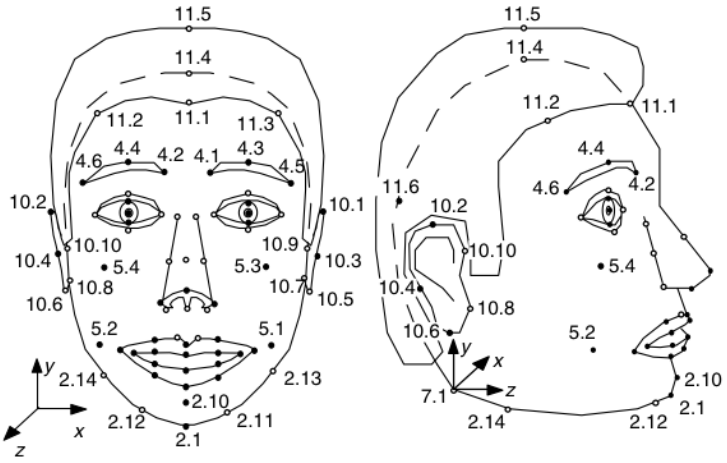

Figura 3.2: FFPs da face[7]

Como já foi referido anteriormente, as FAPs são um conjunto de parâmetros representativos de acções faciais básicas e permitem a simulação de expressões faciais naturais[10]. Na definição do standard, cada FAP tem associado a si a FAPU

[3]Facial feature point.

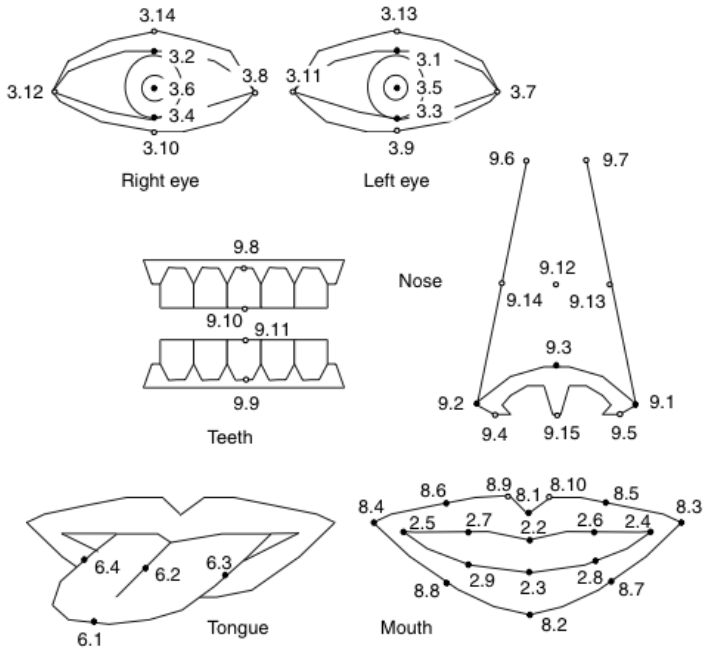

Figura 3.3: FFPs dos olhos, lábios, dentes, língua e nariz[7].

correspondente, o grupo ao qual pertence e a direcção do movimento positivo (caso não seja uni-direccional)[4].

3.2 Modelo Candide-3

Candide é uma máscara facial parametrizada, desenvolvida especificamente com o objectivo de codificar a face humana com base num modelo pré-definido. O seu baixo número de polígonos permite uma reconstrução rápida e pouco exigente em termos de poder computacional[8]. A primeira versão deste modelo foi desenvolvida por Mikael Rydfalk da Universidade de Linköping em 1987 e era constituída por 75 vértices e 100 faces[39]. Este modelo foi utilizado em muito poucas ocasiões. Por sua vez, uma versão ligeiramente modificada por Mârten Strömberg, com 79 vértices e 108 faces, já obteve maior aceitação[8]. Esta versão ficou conhecida por *Candide-1.*

A versão seguinte do modelo *Candide*, a *Candide-2*, foi desenvolvida por Bill Welsh na sua tese de doutoramento. Este modelo, já mais complexo, conta com 160 vértices e 238 faces. Esta diferença significativa em relação às versões anteriores acontece porque nesta versão já são considerados no modelo os dentes, o cabelo e os ombros[40]. Na imagem 3.4 (cima e centro) é possível ver a diferença entre a versão 1 e 2 deste modelo.

A versão 3 deste modelo, desenvolvida por Jörgen Ahlberg em 2001, além de acrescentar alguns vértices, eliminou outros[5]. Estas alterações foram motivadas quer pela aparência irrealista da versão 2, que apresenta olhos e boca rudemente definidos, quer também pelo facto de existir a necessidade de tornar o modelo *Candide* compatível com o standard MPEG-4 para animação facial[10]. Este standard,

[4]No apêndice D está disponível a lista completa de FAPs definidas no standard.

[5]No apêndice B está disponível a lista completa de vértices e de faces que constituem o modelo Candide-3.

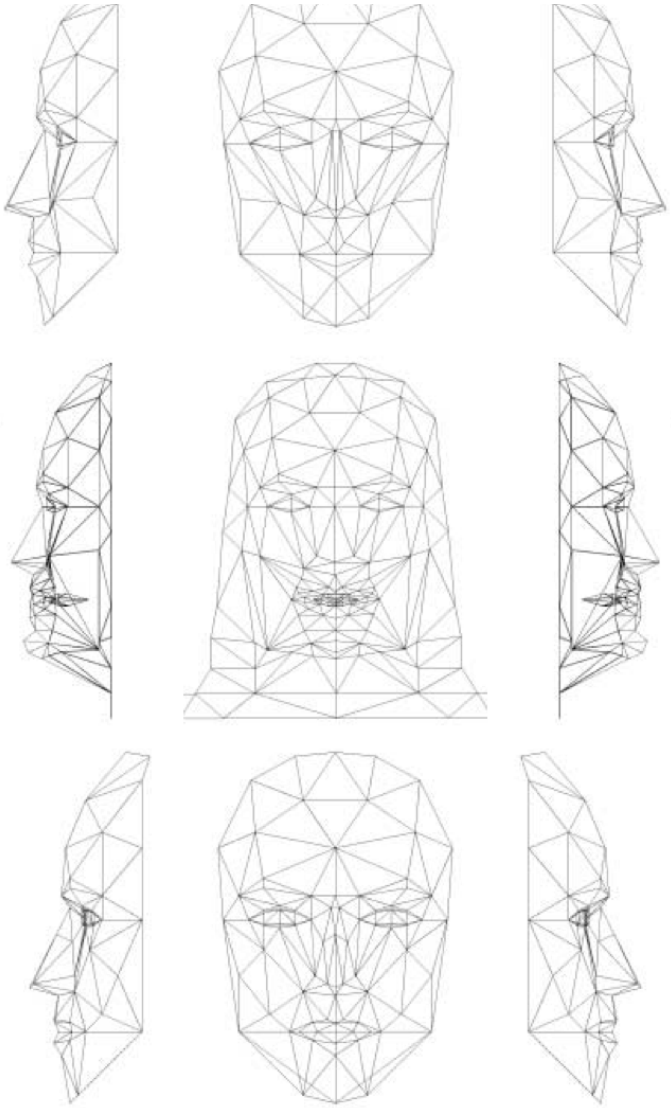

Figura 3.4: Representação dos modelos Candide-1 (em cima), Candide-2 (ao centro) e Candide-3(em baixo) [8]

como foi descrito anteriormente, define alguns FFPs que não tinham correspondente entre os vértices definidos na versão 2 do modelo *Candide*. De forma a resolver esse problema, foram adicionados alguns vértices, ficando assim a versão 3 do modelo com 113 vértices e 168 faces.

A imagem 3.4(baixo) apresenta uma representação gráfica do modelo *Candide-3*. Como se pode verificar pela comparação desta com a imagem 3.4(centro e cima), a versão 3 apresenta diferenças bastante significativas em relação às anteriores. No caso da versão 2 foram retirados os vértices e as faces que representavam os ombros e o cabelo. Além disso, foram refinadas a boca e os olhos.

Ajustar e animar o modelo

A versão 3 do modelo *Candide*, além de definir uma forma facial genérica, define também um modelo de parametrização e dois modelos de animação.

A parametrização, tal como está definida no modelo original, permite o ajuste

do tamanho da cabeça, da posição vertical dos olhos, sobrancelhas, boca e nariz, assim como da largura e altura dos olhos ou do tamanho do nariz.

Os modelos de animação disponibilizados na versão 3, estão de acordo, respectivamente, com as FAPs[6] da norma MPEG-4 e com as *Action Units*[7] definidas por Ekman e Friesen em *Facial action conding system*[11].

Segundo Ahlberg[8], para um modelo facial ser compatível com a norma MPEG-4, há duas questões essenciais: que FFPs correspondem a cada vértice e como se calculam as FAPUs. Em resposta à primeira questão, Ahlberg, como já foi referido anteriormente, adicionou alguns vértices ao modelo, no entanto, não implementou a totalidade das FFPs, mais concretamente, deixou de fora as pertencentes ao grupo 6, uma vez que o modelo não tem língua. Da mesma forma, deixou de fora as FFPs representativas dos dentes, das orelhas e do cabelo[8][8]. Tal como já foi referido anteriormente, as FAPUs são unidades de medida cujo principal objectivo é tornar genéricas as transformações realizadas no modelo, ou seja, se a uma determinada configuração do modelo for aplicada uma transformação que pisca os olhos, essa transformação terá que obter um resultado semelhante quando aplicada a uma configuração distinta. Esse resultado é obtido pela definição de medidas genéricas que se auto calculam consoante o modelo em causa. A tabela 3.3 mostra a forma de cálculo das várias unidades de medida na versão 3 do modelo.

Tabela 3.3: Tabela de relação entre FAPUs e o modelo Candide[8]

	FAPU	**CANDIDE**
AU	Angular unit	$1e^{-5}$ radians
MW	Mouth width	31.x - 64.x
MNS	Mouth-nose separation	6.y - 87.y
ENS	Eye-nose separation	*pupil.y* - 6.y[a]
ES	Eye separation	*rpupil.x*[b] - *lpupil.x*[c]
IRISD	Iris diameter	73.y - 74.y

[a] Ponto médio de *rpupil* e *lpupil*
. [b] Ponto médio dos vértices 69, 70, 73, 74
. [c] Ponto médio dos vértices 67, 68, 71, 72
.

Tendo as FAPUs e as FFPs já é possível implementar as FAPs definidas no standard MPEG-4, no entanto o modelo*Candide-3* não permite a implementação de todas as FAPs, isto porque, como referido anteriormente, também não tem implementadas a totalidade das FFPs. Assim, não são possíveis de implementar as FPAs 43-47(correspondentes à língua) e 65-68(correspondendo às orelhas). De igual forma as FAPs 23-30 não estão presentes nesta implementação do modelo, uma vez que dizem respeito aos olhos e as faces necessárias não estão implementadas[8].

[6]Lista completa de FAPs no apêndice D

[7]Lista completa de *Action Units* disponível no apêndice F.

[8]A lista completa de vértices e a respectiva tradução para FFPs da norma MPEG-4 está disponível no apêndice C.

Candide-3 & SUIWA

A escolha da versão 3 deste modelo como base de trabalho neste projecto esteve relacionada com o facto de este ser bastante leve em termos de vértices e faces e também por estar de acordo com os mais recentes standards de animação facial, como é o caso da norma MPEG-4.

O facto do modelo apresentar poucos vértices tem um valor acrescentado nesta aplicação específica, uma vez que o objectivo é que esta seja utilizada em ambiente web, e como tal, qualquer optimização é muito útil. O facto da aplicação ser compatível com o standard em vigor torna essa aplicação muito mais versátil, daí a importância em utilizar um modelo compatível com a norma MPEG-4 para animação facial.

Embora o modelo *Candide-3* tenha muitas vantagens, no decorrer do desenvolvimento do SUIWA ficou patente que também apresenta algumas fraquezas, daí terem sido feitas alterações ao modelo por forma a ir ao encontro aos objectivos da aplicação. As alterações não se fizeram ao nível do modelo propriamente dito, mas sim da zona envolvente (fundo). De referir que as alterações efectuadas não interferiram com a compatibilidade com o standard, no entanto introduziram alguns vértices extra.

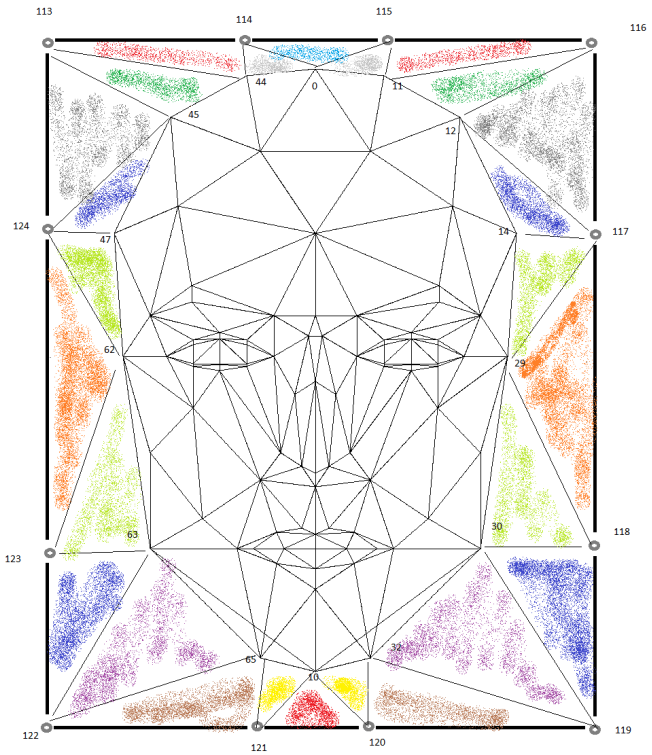

Figura 3.5: Primeira alteração ao modelo Candide-3. Adaptado de [8]

A primeira alteração efectuada ao modelo teve como objectivo adicionar contexto à face, ou seja, considerar uma parte do fundo da imagem. Esta alteração veio aumentar em muito a sensação de realismo da animação. A imagem 3.5 mostra os vértices adicionados, bem como as novas faces que representam o fundo. Os novos vértices foram adicionados ao final da lista de vértices, de forma a manter a

estrutura original do ficheiro sem alterações.

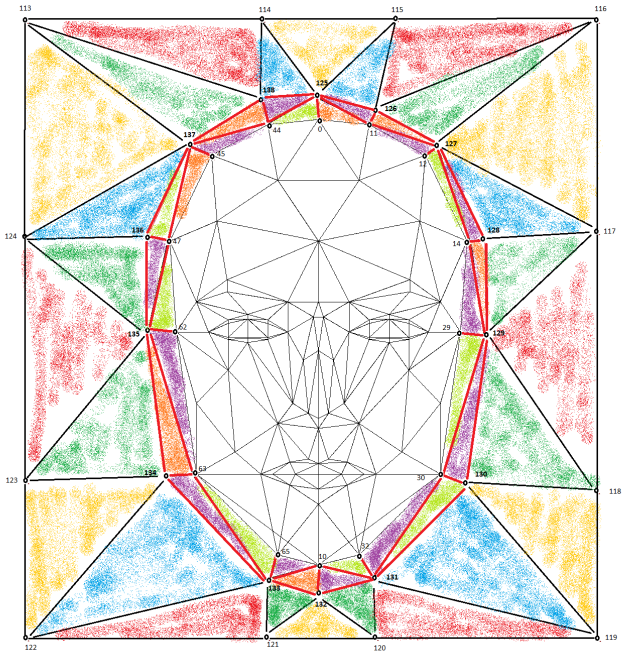

Figura 3.6: Segunda alteração ao modelo Candide-3. Adaptado de [8]

Nos testes à versão modificada do modelo genérico verificou-se que as faces adicionadas esticavam quando o modelo seguia o rato. Para resolver esse problema foi adicionada uma segunda linha de vértices e faces que iriam "absorver" esse efeito, ficando o utilizador com uma animação mais próxima da realidade.

Na imagem 3.6 está representada a linha extra de vértices adicionada entre o modelo e o contorno adicionado na alteração anterior. Embora na imagem esta linha seja visível, isso não acontece no modelo em si, uma vez que a linha se encontra "escondida" por detrás da face do modelo, escondendo assim o efeito de esticar que é criado nas novas faces.

Um modelo com poucos vértices e faces, tal como referido anteriormente, apresenta vantagens por exemplo em questões de desempenho. Dessa forma, antes de se optar pela adição de vértices ao modelo, foi testada outra abordagem, na qual o fundo não tinha qualquer relação com este, evitando dessa forma a adição de vértices ao modelo facial. No entanto, essa solução revelou-se pouco eficaz, especialmente devido ao facto do modelo seguir o rato, tendo assim movimento rotativo.

Quando existia movimento, partes do fundo que não deveriam ser visíveis ficavam descobertas. Então, foram adicionados vértices ao modelo e foi comparado o desempenho entre as versões com mais vértices e com menos, que se verificaram praticamente iguais em termo de desempenho, tendo obtido cada uma delas valores próximos dos 60fps[9].

Sendo apenas uma malha tridimensional, qualquer transformação aplicada ao modelo, mesmo que não fosse à totalidade do modelo, não surtiria o mesmo efeito que tendo o fundo independente do modelo.

[9]Fotogramas por segundo.

3.3 Node.js

Node.js[10] é uma plataforma de desenvolvimento especialmente vocacionada para aplicações web, sejam elas cliente ou servidor.

A plataforma Node.js apresenta uma grande escalabilidade em aplicações em rede devido à junção de chamadas assíncronas de Input/Output(I/O), JavaScript do lado do servidor, utilização de funções anónimas de JavaScript e uma arquitectura baseada em eventos com apenas uma *thread*[41].

O *core* desta plataforma é uma máquina virtual de JavaScript implementada sobre o motor de JavaScript V8 da Google™, auxiliado por um *event loop* não bloqueante e por um conjunto de bibliotecas de I/O de ficheiros e de rede.

Embora a linguagem utilizada para programar em Node.js seja JavaScript, esta não é exactamente igual à utilizada para programar para um browser, uma vez que em Node.js não existe qualquer das capacidades de interacção com o browser. Além de código em JavaScript, esta plataforma apresenta também a possibilidade de incorporar módulos escritos em C/C++, como forma de potenciar as suas capacidades.

O JavaScript em si, tem uma grande desvantagem, que são os objectos globais, que quando definidos como globais, o são para a totalidade da aplicação. No entanto, a plataforma Node.js, com recurso ao módulo *CommonJS*, resolve esse problema introduzindo o conceito de contexto, ou seja, cada objecto só é global dentro do contexto em que é definido[41].

Como refere Herron[41], a utilização da mesma linguagem para programação da aplicação cliente e da aplicação servidor oferece várias vantagens pontenciais, das quais importa destacar o facto da mesma equipa de desenvolvimento poder trabalhar nas duas aplicações sem haver necessidade de trocar de paradigma de desenvolvimento, o código desenvolvido para uma das aplicações poder facilmente ser migrado para a outra e também o facto das ferramentas de desenvolvimento, debug e testes serem comuns, o que implica menores custos para a empresa.

Vantagens e desvantagens na utilização de Node.js

A principal vantagem na utilização de Node.js é a sua performance. Segundo Herron[41] a causa dessa performance é a junção entre o motor de JavaScript V8 da Google™com uma arquitectura assíncrona baseada em eventos.

Enquanto nas aplicações síncronas que utilizam *threads* estas esperam que os pedidos de I/O sejam terminados para seguirem para o próximo pedido, em Node.js isso não acontece, o pedido é feito e a aplicação passa imediatamente para o próximo pedido, voltando apenas ao pedido inicial quando este despoleta um evento. Como o Node.js tem apenas uma *thread*, não existe troca de contexto, logo o custo associado a essa troca também não existe.

Por sua vez, em aplicações em que seja necessário muito processamento, a performance de Node.js já não é tão satisfatória. A explicação para esta degradação de performance está relacionada novamente com o facto de apenas existir uma *thread*. Caso essa *thread* seja ocupada com muito processamento os restantes pedidos têm que esperar que termine. Um exemplo desta situação é por exemplo o cálculo

[10]Mais informação disponível em `http://nodejs.org/`

da sequência de *fibonacci*, em que ao fim de poucos segundos o Node.js deixa de responder.

As duas premissas levam a uma conclusão, a plataforma Node.js é realmente vantajosa em aplicações onde seja necessário responder a muitos pedidos de I/O, por exemplo sistemas onde existam muito pedidos de dados a bases de dados e pouco processamento do lado do servidor. Já não é tão vantajosa a sua utilização em ambientes de muito cálculo do lado do servidor.

Node.js & SUIWA

A utilização da plataforma Node.js neste projecto foi uma decisão tomada pela empresa, não tendo o estagiário qualquer influência nessa decisão. No entanto, uma vez que a necessidade de processamento do lado do servidor não é muito intensa[11], a adopção da plataforma Node.js para o projecto SUIWA revelou-se positiva.

Ao nível de IDE(Integrated Development Environment) a opção foi o Sublime Text 2[12]. Esta aplicação revelou-se interessante uma vez que está dotada de ferramentas de reconhecimento de várias linguagens de programação e possibilita a instalação de diversos *plugins* que permitem a personalização do ambiente de desenvolvimento às necessidades de cada projecto.

3.4 WebGL e Three.js

Embora a especificação da versão 5 do HTML tenha previsto um ambiente de renderização 3D, actualmente esse ambiente ainda não se encontra definido. Dessa forma, e de forma a conseguir utilizar as capacidades inerentes ao ambiente 3D num *browser*, o Kronos group^{TM}publicou em 2011 a especificação do WebGL[13].

WebGL

O WebGL apresenta-se como um contexto de renderização 3D para o elemento Canvas de HTML5. Este contexto permite a renderização utilizando uma API semelhante ao OpenGL.

Actualmente a especificação de HTML5 conta apenas com a definição do contexto 2D(*"CanvasRenderingContext2D"*). O que o WebGL apresenta é um outro contexto denominado *"WebGLRenderingContext"*.

O aparecimento deste novo contexto permite explorar novos caminhos na programação Web, tirando mais partindo das máquinas e dando aos utilizadores uma experiência diferente daquela que estavam habituados.

Devido à grande diversidade de utilizações que os gráficos 3D podem ter, o WebGL decidiu disponibilizar um conjunto de primitivas que possam ser utilizadas para diferentes propósitos[42].

Segundo a especificação, existem diversas bibliotecas que permitem uma interacção mais simples com a API do WebGL, umas mais genéricas e outras mais específicas.

Neste projecto a escolha da biblioteca a utilizar recaiu sobre o Three.js.

[11]Mais detalhes no capítulo V.

[12]Mais informação em `http://www.sublimetext.com/`.

[13]Mais informação em `https://www.khronos.org/registry/webgl/specs/1.0/`.

Three.js

Esta biblioteca, escrita em JavaScript, permite uma interacção mais amigável com a API do WebGL. É uma biblioteca cujo um dos objectivos é ser leve e fácil de utilizar[43].

Além da utilização do contexto 3D permitido pelo WebGL, esta biblioteca tem também capacidade de desenho vectorial, recorrendo ao elemento SVG do HTML5 e também de utilização do contexto 2D do elemento Canvas.

Existe uma comunidade muito activa que suporta e desenvolve esta biblioteca, evoluindo-a muito rápidamente. Desde o início de 2012 já foram disponibilizadas 3 versões.

A escolha desta biblioteca para o projecto SUIWA foi muito influenciada pelas capacidades demonstradas em alguns exemplos presentes em[43], e também pelo facto da comunidade de suporte ser muito dinâmica, dando suporte, em tempo útil, a todas as questões colocadas.

Capítulo 4

Planeamento e metodologia

Neste capítulo será apresentado o planeamento do estágio, sendo as tarefas divididas por semestres, e a metodologia de desenvolvimento utilizada. Nas figuras 4.1 e 4.2 estão representados os mapas de Gantt relativos a cada um dos semestre e servem, de alguma forma, para auxiliar na compreensão do planeamento. Associado a cada uma das figuras está uma explicação um pouco mais detalhada de cada uma das tarefas apresentadas graficamente.

O número de horas de trabalho durante o primeiro semestre foi de 28 horas semanais. Esse número, embora superior ao exigido pelo Departamento de Engenharia Informática, foi acordado entre o estagiário e a entidade de acolhimento. Sendo que no decorrer do segundo semestre foi cumprido o estipulado pelo regulamento de estágios que define como tempo de trabalho 40 horas semanais.

4.1 Planeamento do primeiro semestre

O primeiro semestre caracterizou-se pela utilização de ferramentas Microsoft™, nomeadamente Visual Studio 2010™como ferramenta de desenvolvimento e IIS™(Internet Information Services) como servidor Web. Teve como principais tarefas o estudo do estado da arte e a realização de algumas provas de conceito[1] de forma a validar algumas questões, como por exemplo a sincronização de áudio com a animação.

Estado da arte

Embora na figura 4.1 não apareça de forma explícita o estudo do estado da arte, aparece um conjunto de tarefas que o constituem.

O mercado de aplicações semelhantes foi a primeira tarefa a ser realizada por motivos de contextualização do estagiário nos resultados esperados no final do estágio.

Finda a análise das soluções semelhantes, foi iniciado um período de estudo do HTML5, e das suas capacidades para representação 3D. Durante esse período foram desenvolvidas pequenas aplicações para testar a viabilidade ou não da utilização de HTML5. Ainda no estudo do estado da arte foram analisados alguns modelos facias 3D genéricos e avaliada a sua adequação ao projecto em causa.

[1] Explicadas em mais detalhe no capítulo 6.

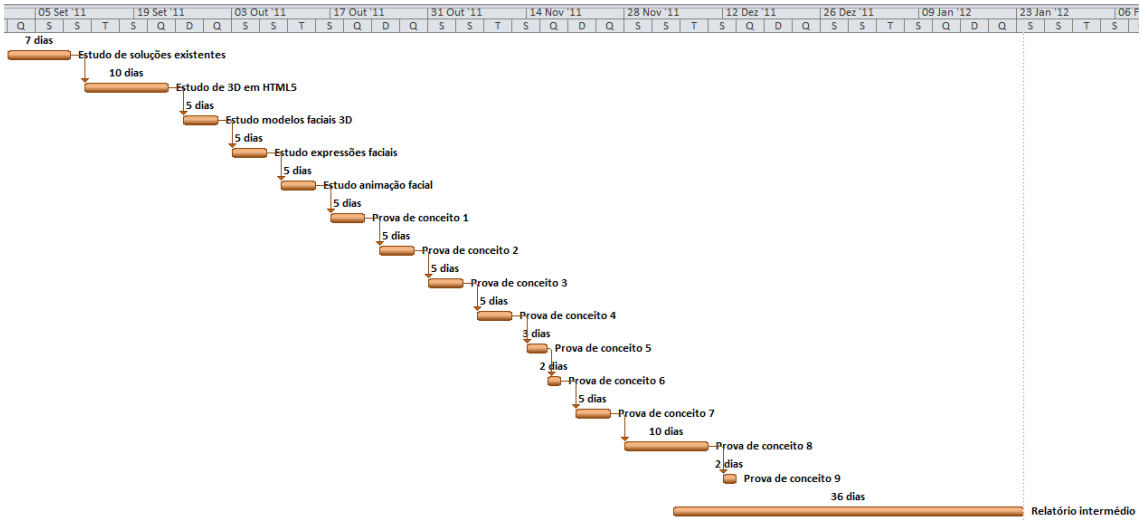

Figura 4.1: Planeamento do primeiro semestre

Para finalizar o estudo do estado da arte, foi investido algum tempo na análise da face humana, seja ao nível das expressões seja ao nível das possibilidades de animação.

Provas de conceito

A segunda parte dos trabalhos do primeiro semestre esteve relacionada com o estudo e desenvolvimento de um conjunto de pequenas aplicações para teste de alguns conceitos e funcionalidades, conforme se explica em maior detalhe no capítulo 6.

- Animação 3D (Prova de conceito 1)

  A primeira aplicação desenvolvida teve como objectivo testar a animação 3D num *web browser* e consistiu apenas na criação de alguns sólidos geométricos e adicionar movimento a um deles.

- Animação 3D com áudio síncrono (Prova de conceito 2)

  Tendo como base a primeira aplicação desenvolvida, esta prova de conceito consistiu na adição de áudio síncrono com o movimento já implementado na aplicação anterior.

- Animação 3D com pedido de dados ao servidor (Prova de conceito 3)

  Nesta iteração, o ambiente é novamente o mesmo, mas desta feita existe pedido de informação ao servidor acerca de características específicas do ambiente, tais como dimensões de objectos ou sentido inicial do movimento.

- Animação 3D com pedido de áudio ao servidor (Prova de conceito 4)

  Aqui, além de serem pedidos ao servidor dados específicos para a construção do ambiente é também pedido o áudio a executar sincronamente, escolhido de forma aleatória no servidor.

- Animação 3D com atraso na resposta do servidor (Prova de conceito 5)

Este teste utiliza novamente o ambiente definido anteriormente, mas desta feita foi introduzido algum atraso na resposta do servidor, de forma a avaliar o comportamento da aplicação.

- Teste da animação em ambiente de pré-produção (Prova de conceito 6)

  A última prova de conceito com este ambiente consiste na disponibilização da aplicação num ambiente semelhante ao ambiente de produção.

- Upload de imagem, marcação de pontos chave e serialização dos pontos (Prova de conceito 7)

  O objectivo desta aplicação passou pelo desenvolvimento de um ambiente diferente e já mais semelhante ou que foi utilizado inicialmente na aplicação SUIWA. O objectivo era implementar o upload de uma imagem, a marcação de pontos chave nessa imagem e a serialização dessa informação para um ficheiro.

- Carregamento da imagem e e aplicação de textura com base nos pontos previamente definidos (Prova de conceito 8)

  Tendo como base a aplicação e os dados da iteração anterior, o objectivo aqui passou pela aplicação de uma imagem como textura de um objecto simples 3D. Esta aplicação deveria ser feita tendo em consideração os pontos chave definidos anteriormente.

- Desenvolvimento de ambiente com múltiplas camadas de objectos 3D (Prova de conceito 9)

  A última iteração realizada nesta aplicação teve como objectivo testar a sobreposição e a transparência de objectos 3D no browser.

Ao longo do semestre foi dedicado também esforço para a redacção do relatório intermédio, tendo o final do semestre sido integralmente ocupado com essa tarefa.

4.2 Planeamento do segundo semestre

A segunda etapa deste estágio caracterizou-se por uma mudança no paradigma de desenvolvimento. Desde o início do ano 2012 que o grupo Inogate adoptou, para desenvolvimento, algumas ferramentas *open source*, nomeadamente ao nível de servidor web, que desde essa data passou a ser Node.js[2].

Numa primeira fase do desenvolvimento foi dada especial atenção ao ajuste de um modelo genérico para que se adaptasse a uma face específica, escolhida pelo utilizador mediante o upload de uma imagem.

O motor de animação é uma das funcionalidades chave do sistema. Numa segunda iteração foi dada especial atenção a este módulo, que é responsável pela modificação do modelo, de forma a que este simule emoções ou visemas.

[2]Mais informação em `www.http://nodejs.org/`

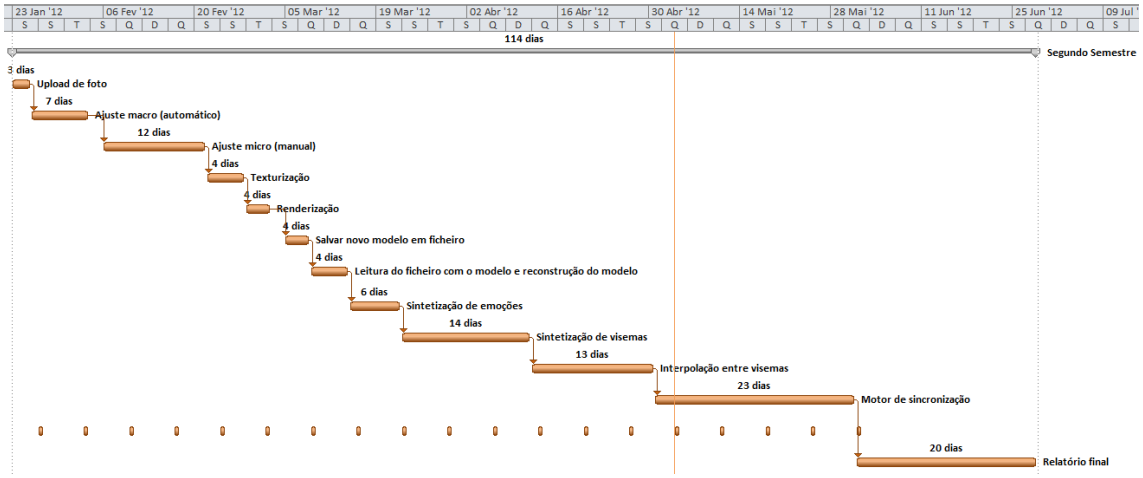

Figura 4.2: Planeamento do segundo semestre

Upload de foto

O upload de uma foto é o primeiro passo para a criação de um novo modelo. Nesta fase do desenvolvimento pretendia-se que fosse desenvolvido um sistema de upload de imagens que posteriormente as redimensionasse para um tamanho pré-definido, mantendo a relação entre a altura e a largura.

Ajuste do modelo genérico

Tendo como guia a imagem carregada anteriormente ou alguma das existentes no sistema, pretendia-se ajustar um modelo genérico para que este se tornesse o mais semelhante possível com a foto de referência.

A primeira fase do ajuste é automática, ficando o utilizador responsável por refinar esse ajuste posteriormente.

- *Ajuste macro (automático)*

  #### Reconhecimento automático da área da face

  De forma a retirar ao utilizador algum trabalho no ajuste do modelo à foto, esta primeira fase do ajuste é automática. É utilizada uma framework externa que implementa openCV[3] para reconhecimento facial.

  A área definida como sendo a face é demarcada e posteriormente, nessa área é renderizado o modelo genérico. Esse modelo é ajustado com operações de escala no eixo X e Y e rotação sobre o eixo Z, para que se torne mais semelhante à foto.

  #### Edição dos ajustes automáticos

  Partindo do ajuste feito de forma automática, o utilizador tem a possibilidade de corrigir alguns ajustes automáticos que não estejam de acordo com a sua preferência.

- *Ajuste micro (manual)*

[3]Biblioteca de visão computacional em tempo real. Mais informação disponível em `http://opencv.willowgarage.com/wiki/`

Esta tarefa consistia no desenvolvimento de um sistema que permita o ajuste de algumas características específicas da face humana, de forma a minimizar as possíveis inconsistências com a realidade.

As características a ajustar são aquelas que mais se destacam na face humana, e que, no caso de existirem erros na sua identificação, permitem ao utilizador um rápido reconhecimento dessas falhas.

Ajustes da face

Este pequeno módulo é o responsável pelo sistema de ajustes de algumas características da face, que não são consideradas globais, como é o caso da posição das sobrancelhas, do comprimento da cabeça ou da altura do osso zigomático[4]

Ajustes dos olhos

Ao nível dos ajustes dos olhos é possível editar o seu posicionamento, a largura e a altura.

Ajustes do nariz

A principal característica do nariz que este módulo permite editar é a sua posição vertical. No entanto, é também possível ajustar a sua altura e o posicionamento vertical da ponta.

Ajustes da boca

No que à boca diz respeito é possível editar o posicionamento e a dimensão. Além disso, é também possível alterar o posicionamento da linha de junção do lábio superior e inferior.

Texturização

De forma a ser possível aplicar uma textura a uma forma tridimensional é necessário definir um mapa de textura (mapa UV)[5].

Nesta fase foi desenvolvido um método automático de criação desse mapa, tendo como base o modelo já ajustado à foto de referência.

Renderização

Depois de ter o modelo completamente definido interessa desenhá-lo. Assim, foi desenvolvido um sistema de renderização da cena especificada, ou seja, um sistema cíclico de desenho das várias componentes da cena, que está constantemente a desenhar a cena, permitindo assim a posterior animação.

Salvar novo modelo para ficheiro

Depois de personalizado o modelo, importa guardá-lo para que seja possível a sua utilização na aplicação.

[4]Osso da bochecha.

[5]Mapeamento de coordenadas necessário para aplicação da textura num modelo 3D. Explicado em pormenor no capítulo VII.

O ficheiro em causa tem informação acerca da localização dos vértices, e da sua correspondência na criação das faces. Tem também informação relacionada com o mapa UV, para que a sua reconstrução seja possível em qualquer altura.

Por uma questão de simplicidade na criação, manipulação e legibilidade do ficheiro, o formato adoptado para o ficheiro em causa foi XML (Extensible Markup Language).

Leitura do ficheiro com a informação de reconstrução do modelo

De forma a atingir um dos objectivos do projecto(animar um modelo facial tridimensional personalizado) é necessário carregar esse modelo para o sistema de animação. Assim, foi necessário desenvolver um módulo de leitura do ficheiro XML que define o modelo e, posteriormente, reconstruir e renderizar esse mesmo modelo no sistema de animação.

Sintetização de emoções

A primeira fase do desenvolvimento do motor de animação consiste num pequeno motor de emoções, que dá ao modelo a capacidade de expressar as seis emoções definidas por Ekman (alegria, tristeza, raiva, surpresa, repugna e medo)[21].

Sintetização de visemas

Esta fase é em tudo semelhante à anterior, sendo que aqui o que se pretendia era a sintetização de 14 dos 15 visemas essenciais para reproduzir todo o léxico da língua Portuguesa de Portugal. O visema em falta é a face neutra, pelo que não foi implementado nesta fase.

O único objectivo aqui presente foi a reprodução isolada dos visemas, deixando a sua interpolação e todo o processo de animação para as fases posteriores.

Interpolação entre visemas

Nesta fase do desenvolvimento, o que se pretendia era a definição de um modelo de interpolação entre visemas. Deve ter-se em conta que o modelo expressa emoções ao mesmo tempo que fala, ou seja, foi também nesta fase definida uma forma de manter sinais faciais consistentes com a emoção em simultâneo com a animação do discurso.

Motor de sincronização

A última etapa da fase de implementação foi uma nova iteração no desenvolvimento do motor de animação. Esta nova iteração teve por objectivo adicionar métodos de sincronização entre a animação e o áudio.

Relatório final

O relatório final do estágio, para além dos 20 dias para elaboração previstos para o final da fase de desenvolvimento, teve previsto também uma componente semanal. Todas as sextas-feiras, além da escrita de um relatório interno, foi também dedicado algum tempo ao relatório final.

4.3 Metodologia de desenvolvimento

Cada tarefa tem uma ou várias formas de ser realizada de forma mais rápida e eficiente, e o desenvolvimento de software não é excepção.

No caso específico do desenvolvimento de software existem várias abordagens possíveis, não sendo nenhuma delas completamente correcta ou incorrecta, mas sim mais ou menos adequada à realidade de cada projecto ou empresa.

4.3.1 Metodologia AGILE

A metodologia de desenvolvimento AGILE define um conjunto de métodos iterativos e incrementais de criar software.

Esta metodologia promove o planeamento adaptativo e um desenvolvimento e libertação de versões evolucionário, ou seja, no final de cada ciclo de desenvolvimento é lançada uma versão de testes , dando dessa forma início a um novo ciclo. De referir que cada ciclo de desenvolvimento pode ser constituído por mais do que uma iteração, podendo estas ser executadas em série ou em paralelo.

Embora esteja muito presente o conceito de planeamento *"on the fly"*, existem prazos que devem ser cumpridos, nomeadamente a data limite para a entrega da solução final.

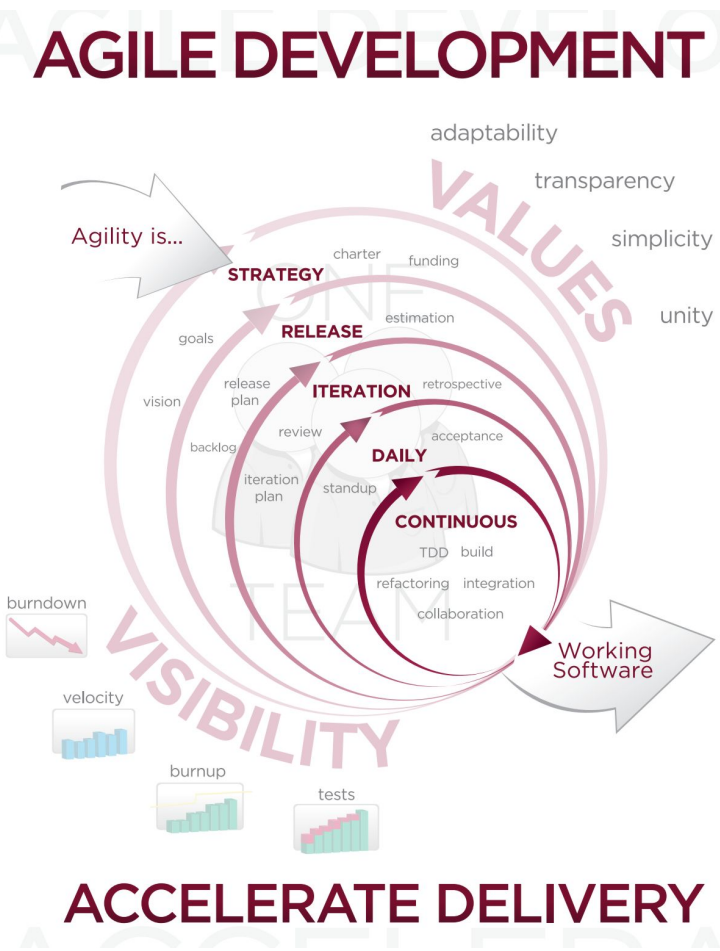

Figura 4.3: Metodologia Agile

Esta metodologia tem como guia um conjunto de doze regras definidas no manifesto Agile[6], das quais se destacam:

- Entrega rápida e contínua de sofware com valor;
- Aceitação de alterações de requisitos mesmo numa fase tardia do desenvolvimento;
- Cliente e equipa de desenvolvimento devem trabalhar juntos, diariamente, durante o decorrer do projecto;
- O método mais eficiente e eficaz de passar informação para e dentro de uma equipa de desenvolvimento é através da conversa pessoal e directa;
- A atenção permanente à excelência técnica e um bom desenho da solução aumentam a agilidade;
- As melhores arquitecturas, requisitos e desenhos surgem de equipas auto-organizadas.

Estas regras, em conjunto com as restantes presentes no manifesto, traduzem-se, na prática, em tarefas divididas em pequenos incrementos que exigem pouco planeamento, em equipas de desenvolvimento localizadas num espaço de trabalho amplo, onde é privilegiado o contacto directo entre as várias pessoas e também numa formação de equipas pouco rígida, com pouco interferência na forma como os seus elementos abordam cada tarefa, exigindo apenas a responsabilização de cada um pelas opções tomadas.

A figura 4.3 mostra uma representação visual das várias fases desta metodologia, bem como dos objectivos delineados para cada uma dessas fases. Seguindo do que tem periodicidade maior para o que tem periodicidade menor, temos a definição da estratégia, que engloba, entre outras coisas, um conjunto de objectivos e a definição do financiamento. Para atingir os objectivos propostos na zona de estratégia são realizadas um conjunto de *releases* de sub-soluções, que posteriormente farão parte da aplicação final. Para cada uma das *releases* é efectuado um pequeno planeamento e estimado o tempo de desenvolvimento. Cada *release* é dividida em pequenas iterações podendo estas ser realizadas em série ou em paralelo.

Ao longo de todo o projecto com uma periodicidade geralmente diária, são realizadas pequenas reuniões de orientação e de validação de trabalho.

4.3.2 Metodologia interna de desenvolvimento

A metodologia de desenvolvimento utilizada no grupo Inogate apresenta-se como sendo baseada em Agile, no entanto com algumas alterações de forma a que se adapte à realidade do mercado onde está inserido.

Assim, no início de cada projecto, além de um primeiro levantamento de requisitos feito com o cliente, são também analisados os standards de arquitectura, design e desenvolvimento passíveis de serem utilizados em cada projecto especificamente.

A actividade *A1*, tal como se pode verificar na figura 4.4, acompanha toda a extensão do projecto. Esta é uma das semelhanças com a metodologia Agile, o

[6]Mais informação disponível em `http://agilemanifesto.org/`

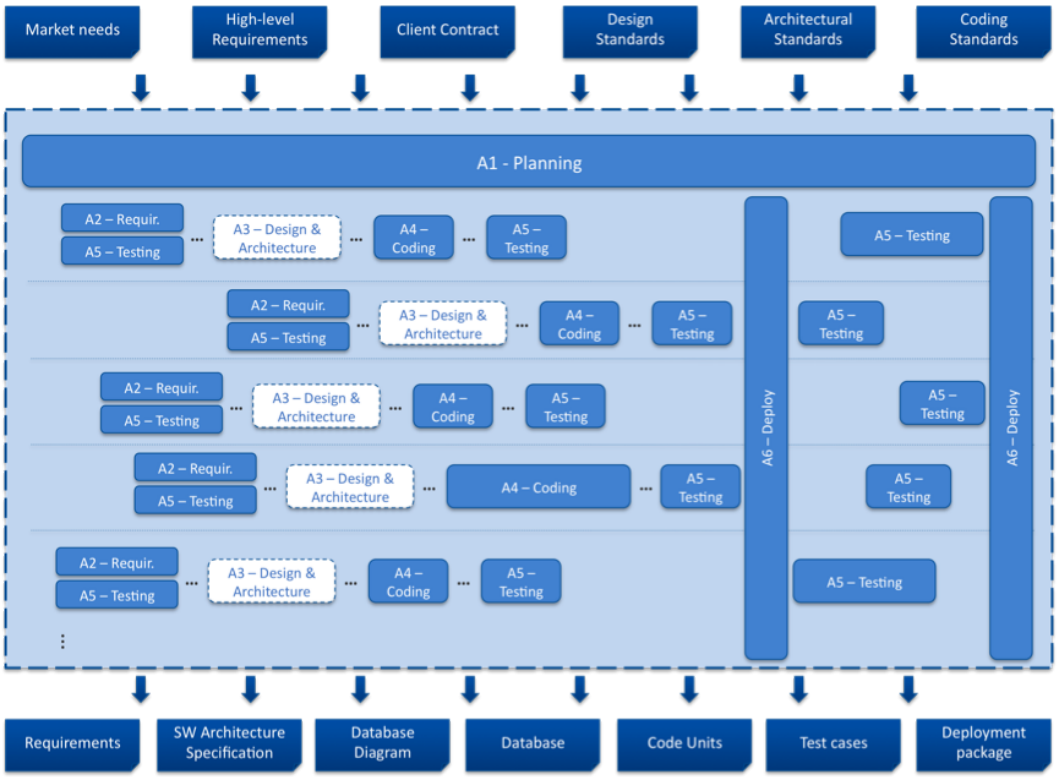

Figura 4.4: Metodologia de desenvolvimento do grupo Inogate

planeamento *"on the fly"*. Nesta actividade é feito o levantamento dos requisitos de alto nível e a definição de deadlines para lançamento de versões, sejam elas intermédias ou a versão final. Esta actividade pressupõe ainda a existência de reuniões periódicas, no entanto não define a periodicidade destas.

A actividade *A2* está definida em cada tarefa definida num determinado *"sprint"* de desenvolvimento, ou seja, na tarefa *A1* são definidas tarefas a desempenhar num ciclo de desenvolvimento e são definidos os requisitos de alto nível associados a cada tarefa. Posteriormente, em cada tarefa, são detalhados os requisitos de alto nível correspondentes, para que a equipa de desenvolvimento saiba com clareza o que deve e como deve implementar.

Na actividade identificada como *A3* é definida a arquitectura a utilizar bem como os elementos de design. Só depois de ter todos os elementos bem definidos é que a equipa de desenvolvimento dá início ao desenvolvimento propriamente dito.

Terminado o desenvolvimento é definida uma bateria de testes a aplicar ao módulo em causa de forma a validar se este efectivamente corresponde às especificações definidas no início da tarefa. Esta sequência de testes é representada na figura 4.4 como *A5*.

A actividade *A6* representa o lançamento de uma nova versão da aplicação, seja ela de uma nova funcionalidade ou de uma versão completa. No diagrama seguinte, esta actividade aparece duas vezes, sendo que a que aparece mais à direita representa o lançamento para o cliente, enquanto a outra representa um lançamento num servidor de testes interno.

Capítulo 5

Arquitectura da solução

O objectivo desta aplicação é a criação da interface visual de uma assistente pessoal virtual. O sistema deverá permitir a criação de um avatar personalizado baseado numa foto escolhida pelo utilizador e também a animação desse avatar simulando discurso e emoções.

O sistema será auxiliado por um motor de semântica que terá a capacidade de responder a perguntas tanto de âmbito específico da aplicação como perguntas mais genéricas.

De forma a transformar o texto devolvido pelo motor de semântica em informação útil para criar uma sequência de animação, a aplicação será auxiliada por um motor de TTS. Essa motor será o responsável por criar o ficheiro de áudio tanto em Português[1] como em Inglês[2], ambos com vozes masculina ou feminina.

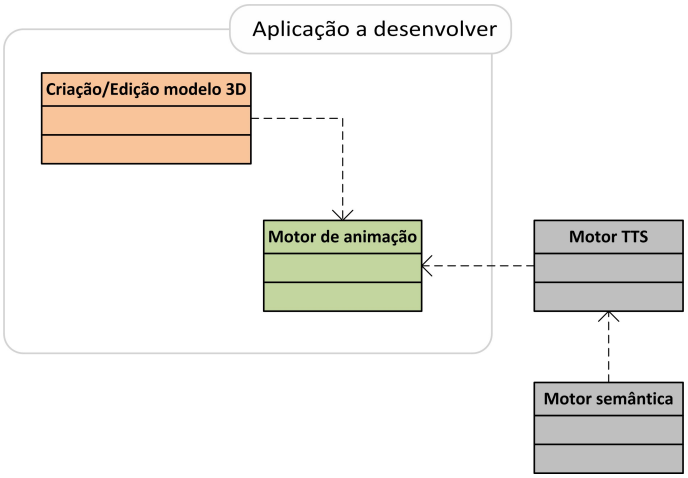

Figura 5.1: Contexto de desenvolvimento da aplicação SUIWA

Desta forma, e tal como se pode ver na figura 5.1, é possível limitar o âmbito da aplicação ao desenvolvimento do módulo de criação e edição do modelo 3D personalizado e do sistema de animação baseado na informação recebida dos restantes módulos.

O módulo de semântica apenas será integrado no projecto numa fase posterior, não tendo qualquer impacto nesta fase do desenvolvimento.

[1] Português do Brasil.
[2] Inglês do Reino Unido.

5.1 Análise de requisitos

Uma das fases mais importantes do planeamento de um projecto de software é o levantamento de requisitos. Dependendo daquilo a que o requisito diz respeito ele pode ser considerado funcional ou não funcional.

5.1.1 Requisitos funcionais

Requisitos funcionais são aqueles que dizem respeito a funções do sistema ou de um componente.

Na tabela 5.1 está uma síntese dos requisitos funcionais identificados para a aplicação SUIWA.

Tabela 5.1: Tabela de requisitos funcionais

ID	Requisito	Descrição	Caso de uso
RF001	*Upload* de foto	O sistema deve permitir o upload de fotos para criação de modelo 3D.	UC007
RF002	Escolha de foto já carregada	O sistema deve permitir a escolha de uma das imagens já carregadas na aplicação para criação do modelo 3D.	UC003
RF003	Ajuste modelo genérico	Sistema de ajuste do modelo genérico à foto escolhida pelo utilizador.	UC004
RF004	Criar modelo 3D	O sistema deve ter a capacidade de criar uma forma tridimensional consistente com a imagem carregada e os ajustes feitos.	UC001
RF005	Guardar modelo 3D	Deve existir no sistema uma forma de guardar um modelo criado num ficheiro, para utilização posterior.	UC001
RF006	Carregar modelo 3D	Deve existir no sistema uma forma de ler um modelo previamente guardado em ficheiro.	UC002
RF007	Seguir o rato	O modelo 3D deve ter a capacidade de se "movimentar" para seguir o movimento do rato.	UC002
RF008	Discurso	O modelo 3D deve ter a capacidade de simular discurso, com base em informação recebida de aplicação externa.	UC006
RF009	Emoções	O modelo 3D deve ter a capacidade de simular emoções.	UC006
RF010	Sincronismo	O modelo deve ter a capacidade de sincronizar a animação com a reprodução do áudio.	UC006

Na tabela 5.1 é possível verificar que cada requisito está associado a um caso de uso, assim, para o UC001 temos todos os requisitos relacionados com a criação de

um novo modelo, desde o carregamento de novas imagens até ao guardar o modelo já criado. Os requisitos RF001 e RF002, em tempo de utilização, são mutuamente exclusivos, ou seja, o utilizador ou escolhe uma foto já carregada ou carrega outra à sua escolha. O requisito RF003 compreende a implementação da totalidade do mecanismo de ajuste do modelo genérico, desde ajuste globais tais como escalas ou rotações, até ajustes mais precisos ao nível dos olhos ou da boca. O RF004 diz respeito à geração do modelo 3D em si, partindo dos ajustes e da imagem previamente definidos. Este passo deve conter o mapeamento de coordenadas para criar um *UV map*[3].

Enquanto no UC001 é importante guardar o modelo em ficheiro, no UC002 importa ler esse ficheiro e carregar essa informação no sistema, é isso que representa o RF006. OS RF007, RF008 e RF009 dizem respeito ao motor de animação, no entanto estão divididos em requisitos mais pequenos de forma a simplificar a sua compreensão e também a induzir a uma implementação modular, na qual a falta de uma das componentes não invalide o correcto funcionamento das restantes.

5.1.2 Requisitos não funcionais

Requisitos não funcionais são requisitos que não estando relacionados com funcionalidades do sistema definem objectivos em termos de desempenho, usabilidade, segurança, disponibilidade ou tecnologias envolvidas. A tabela 5.2 apresenta a síntese de requisitos não funcionais definidos para a aplicação.

Este conjunto de requisitos, embora não estejam directamente relacionados com funcionalidades a implementar, têm grande influência no desenvolvimento da solução. O NF001 e NF003 têm grande impacto na escolha das tecnologias a utilizar, visto que a solução final deve ser disponibilizada na web, no entanto, sem recurso a qualquer tipo de *plugins*.

A aparência e movimento realista (NF005 e NF006) têm também grandes implicações ao nível do desenvolvimento. Enquanto o NF005 implica a criação de uma textura com base numa foto, de forma a ser o mais próximo possível da aparência desejada, o NF006 implica o estudo dos movimentos faciais humanos, de forma a evitar movimentos erróneos e não consistentes com a mobilidade humana.

5.2 Casos de uso

Esta aplicação proporciona ao utilizador duas formas principais de interacção, como se pode ver na figura 5.2, a criação de um modelo tridimensional personalizado e a animação da leitura de um texto.

Para criar um novo modelo, o utilizador tem a possibilidade de carregar uma foto à sua escolha ou escolher uma das que já se encontram no sistema. Depois de concluído esse passo ser-lhe-á pedido que ajuste um modelo genérico à face (identificada automaticamente) da foto que escolheu, ficando assim concluída a criação de um novo modelo.

A interacção com o modelo é mais simples, tendo o utilizador apenas que escrever algum texto e esperar que o sistema comece a sua leitura. A leitura do texto é

[3]Conjunto de coordenadas que indicam que pontos da textura estão mapeados para determinado vértice do modelo.

Tabela 5.2: Tabela de requisitos não funcionais

ID	Requisito	Descrição
NF001	Interface web	O sistema será desenvolvido para posterior disponibilização em plataforma web, concretamente no browser *Google Chrome*^{TM}.
NF002	Facilidade de utilização	A utilização do sistema deverá ser muito simples, pelo que o interface com o utilizador deverá ser auto-explicativo.
NF003	Sem plugins	A aplicação deverá ser desenvolvida sem recurso a qualquer tipo de plugins, ou seja, o código da aplicação cliente deverá ser apenas HTML, CSS e Javascript.
NF004	Utilização de tecnologia actual	Utilização de HTML5 e CSS 3.
NF005	Aparência realista	O modelo 3D deve ser o mais realista possível no que ao aspecto visual diz respeito.
NF006	Movimento realista	O modelo 3D deve ser muito realista em termos de movimento facial.
NF007	Tempo de acesso	A aplicação deverá demorar menos de 10seg a carregar[44].
NF008	Disponibilidade	Sendo que a aplicação será integrada numa solução de maior dimensão, deverá estar disponível em paralelo com esta.

composta pela reprodução do áudio correspondente ao texto introduzido e por uma animação do modelo, na qual este apresenta expressões faciais consistentes com as palavras reproduzidas, dando uma sensação de maior realismo ao utilizador.

Tabela 5.3: Tabela de casos de uso

ID	Caso de uso	Descrição
UC001	Criar novo modelo	Criar um novo modelo com base numa imagem.
UC002	Interagir com o modelo	Interagir com o avatar, introduzindo texto e visualizando a animação.
UC003	Escolher uma foto	Escolher foto a utilizar no modelo.
UC004	Ajustar o modelo	Processo de ajustar o modelo de forma a corresponder às características da foto escolhida.
UC005	Escrever texto a animar	Escrever texto que será posteriormente animado.
UC006	Visualizar a animação	Visualizar o resultado da animação com áudio sincrono.
UC007	Carregar uma nova foto	Escolher uma foto que posteriormente será carregara para a aplicação.

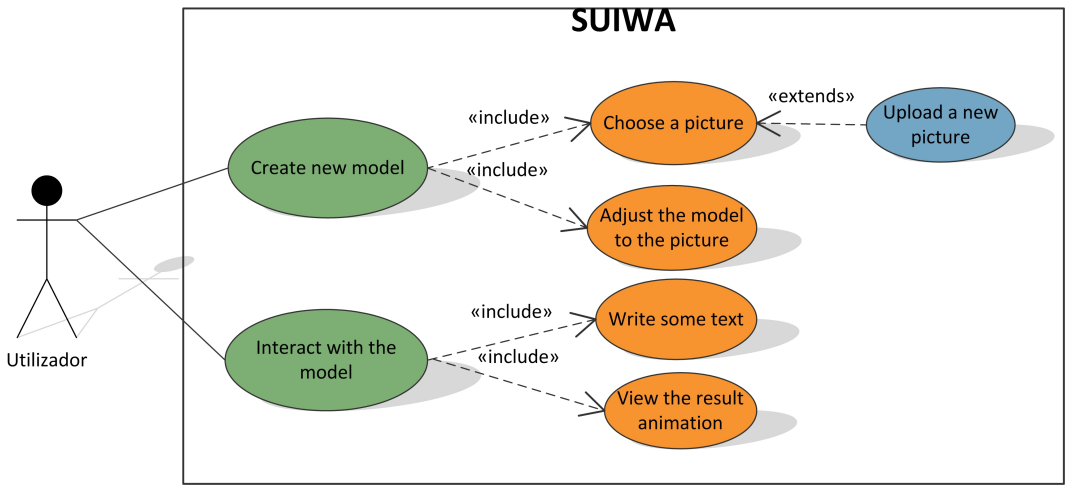

Figura 5.2: Diagrama de casos de uso da aplicação SUIWA

5.3 Arquitectura da solução

Sendo uma aplicação desenvolvida para a web, existe necessidade de desenvolvimento de alguns módulos que fazem parte do servidor. A figura 5.3 mostra a interacção entre o utilizador, o browser (aplicação cliente) e o servidor. Nessa figura é possível verificar que a carga maior de processamento está na aplicação cliente, sendo o servidor utilizado apenas para devolver as páginas, guardar e ler ficheiros e fazer pedidos à aplicação de TTS, enviando posteriormente o resultado para o browser.

De forma a perceber claramente o que se pretende da aplicação SUIWA, quais os módulos core do seu desenvolvimento e quais as suas interacções com aplicações externas, foi desenvolvida uma arquitectura do sistema onde essa informação está patente. Na figura 5.4 é apresentada uma representação gráfica dessa arquitectura. Numa primeira análise da figura 5.4 é possível verificar que por exemplo o sistema de TTS não faz parte das funcionalidades a serem implementadas, e como tal, será utilizada uma aplicação externa.

Carregar imagem

O primeiro módulo da aplicação, tal como o nome indica, tem como principal função permitir ao utilizador o envio de uma imagem para o servidor. Esta imagem será redimensionada para um valor de largura pré-definido e posteriormente utilizada como guia na modificação do modelo genérico e na criação da textura.

Criar modelo

Neste módulo, o objectivo é criar um novo modelo tridimensional, baseado numa imagem guardada no sistema (seja uma já existente ou acabada de carregar pelo módulo anterior). A face a modelar será, numa primeira fase, detectada pelo sistema, sendo os ajustes mais finos realizados pelo utilizador num sistema que permite ajustar as principais variáveis na face humana.

Depois de todos os ajustes efectuados é criado um ficheiro com a informação do novo modelo e da textura, a criação do ficheiro é justificada pela necessidade

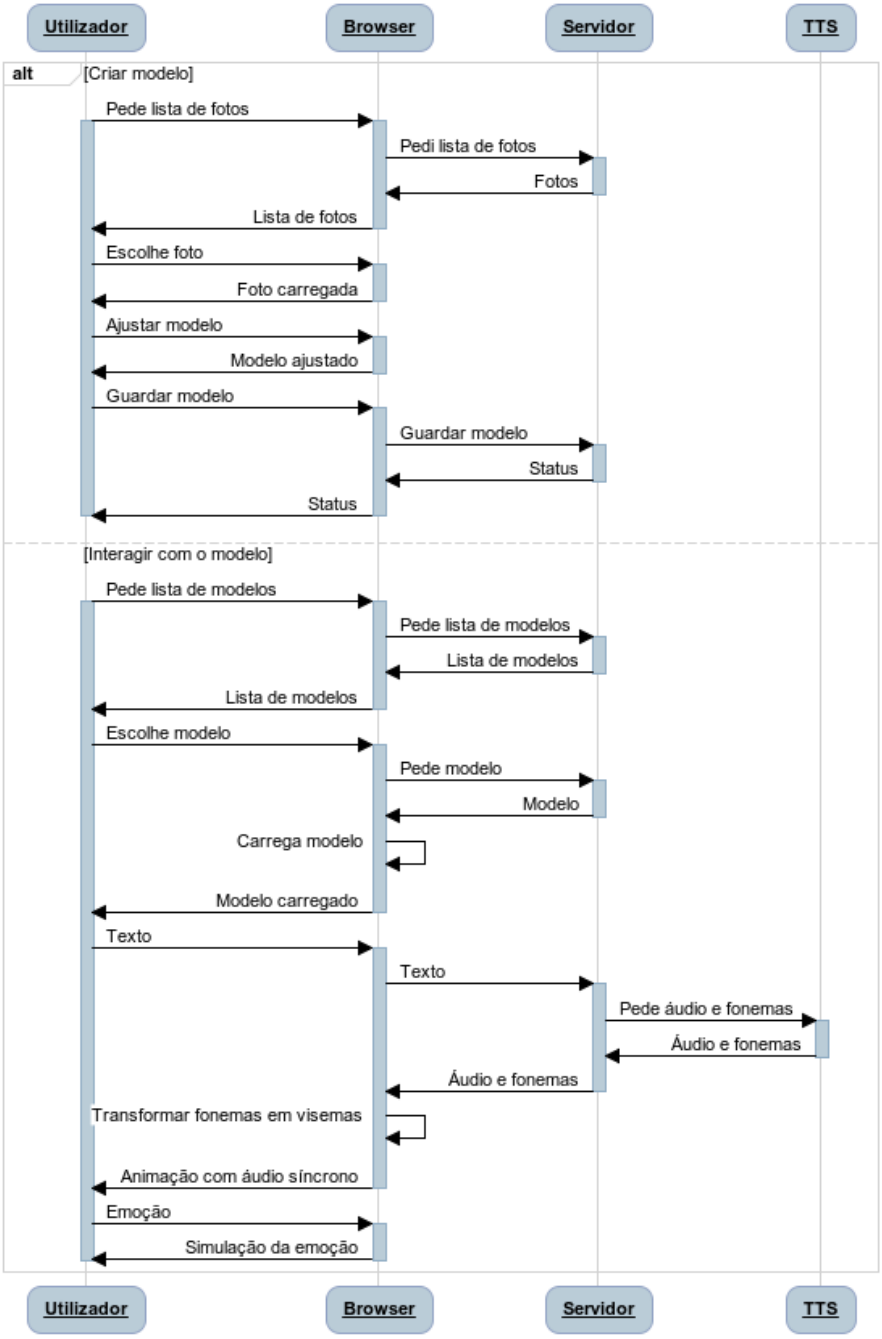

Figura 5.3: Diagrama de sequência da aplicação SUIWA

de utilização desse mesmo modelo em qualquer altura, sem existir a necessidade de efectuar sempre os ajustes.

Interagir com o SUIWA

Este módulo representa o objectivo final desta aplicação, ou seja, a interacção com o assistente pessoal virtual. No contexto deste estágio, a interacção está limitada à escrita de algum texto e à visualização da animação do modelo enquanto este "lê" o texto fornecido. Além dessa interacção directa, existe uma forma de interacção

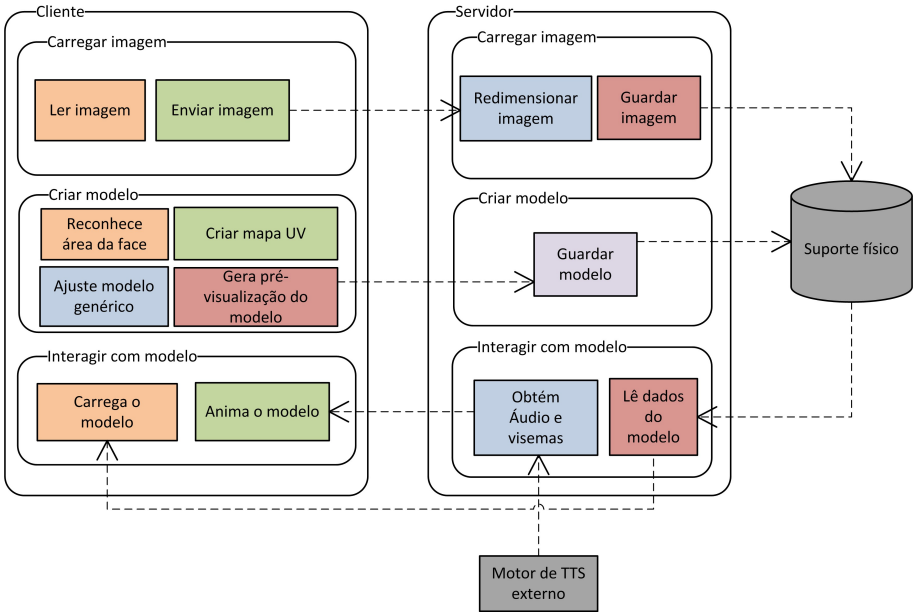

Figura 5.4: Diagrama da arquitectura da aplicação SUIWA

indirecta, em que o modelo simula algumas emoções e tem alguns "comportamentos"que aumentam o seu realismo, como por exemplo, piscar os olhos.

5.4 Análise de risco

Este projecto foi desde o início, um projecto arriscado, quer pelos objectivos que se pretendiam atingir, quer pela dependência de terceiros para a sua conclusão com sucesso.

O primeiro factor que poderia pôr em risco o projecto era a falta de suporte por parte de alguns browsers às tecnologias que se pretendiam utilizar. Nomeadamente, a falta de suporte à tag *Canvas* de HTML5 e ao *WebGL* por parte da versão actual do browser da Microsoft™. Outra preocupação, inerente à anterior estava relacionada com a falta de informação acerca desse suporte em versões posteriores do browser, informação essa que até à data deste relatório também não está disponível.

Depois de analisar as tendências de utilização dos actuais browsers, a empresa optou por arriscar e promover o desenvolvimento da aplicação, descartando todas as versões existentes do browser da Microsoft™. Direccionou-se assim para versões futuras tanto do Internet Explorer™, como dos restantes browsers que à data não ofereciam o suporte necessário.

Actualmente, e como se pode verificar pela tabela 5.4, apenas o Internet Explorer não oferecem o suporte necessário para a execução da aplicação.

Outro factor de risco considerado está relacionado com a aplicação de TTS. Visto tratar-se de uma aplicação externa existia o risco de não estar disponível na data em que fosse necessária.

A solução encontrada para o caso dessa aplicação não estar disponível tinha apenas a finalidade de testar e validar a aplicação desenvolvida, não sendo por isso uma solução definitiva. Essa solução passava por simular, de forma manual a informação que deveria ser devolvida pela aplicação de TTS, ou seja, seria gravado

Tabela 5.4: Tabela de suporte dos diferentes browsers às funcionalidades necessárias à execução da aplicação

	Suporte HTML5	**Suporte WebGL**
	Parcial	Não
	Parcial	Activação explicita
	Parcial	Activação explicita
	Parcial	Sim
	Parcial	Sim

um ficheiro de áudio com a leitura de uma pequena frase e posteriormente seria analisado de forma a criar a informação relativa a visemas e aos tempos em que estes seriam mostrados.

Essa solução no entanto não se revelou necessária, uma vez que a aplicação TTS foi disponibilizada em tempo útil.

Capítulo 6

Trabalho desenvolvido

Neste capítulo será apresentada a aplicação desenvolvida, assim como as principais opções tomadas, as dificuldades encontradas e as respectivas soluções. Serão também apresentados alguns testes e os respectivos resultados e conclusões.

Além da aplicação propriamente dita, será também apresentado um conjunto de pequenas aplicações desenvolvidas com o intuito de validar se o projecto era exequível, e quais as limitações ao seu desenvolvimento.

Além das grandes tarefas já referidas e que serão apresentadas em pormenor nas secções seguintes, foram também realizadas algumas tarefas de apoio a projectos já existentes na empresa, como a escrita de documentação (exemplo de um manual para o projecto Avatar pode ser consultado na apêndice I) ou a migração do projecto Avatar para o novo sistema de TTS(IVONA TTS).

No decorrer do estágio foi ainda elaborada documentação interna relacionada com o projecto. Foram elaborados relatórios com periodicidade semanal (onde era descrito o trabalho realizado naquele período), e também um documento que apresenta um levantamento de algumas soluções semelhantes, centros de investigação e conferências de interesse para o projecto. Esse documento pode ser consultado no apêndice H. Foi ainda elaborado um manual de instalação da aplicação SUIWA[1], um manual de utilizador[2] e um manual de instalação da aplicação IVONA TTS[3].

6.1 Provas de conceito

As provas de conceito desenvolvidas foram pequenas aplicações cujo objectivo primário foi validar a exequibilidade do projecto. Como objectivo secundário, estas aplicações serviram para antever algumas dificuldades que no futuro pudessem pôr em causa o desenvolvimento da solução principal. O desenvolvimento destas pequenas aplicações foi importante na medida em que a resolução de dificuldades em aplicações de pequena dimensão, regra geral, será de mais fácil do que a mesma questão num projecto maior.

Estas provas de conceito consistiram em duas aplicações, divididas em pequenas tarefas com objectivos muito concretos. Uma das aplicações foi uma pequena

[1]Documento disponível no apêndice K.
[2]Documento disponível no apêndice L.
[3]Documento disponível no apêndice J.

animação 3D semelhante ao jogo Pong, que consistia numa caixa aberta à frente e atrás de forma a ser possível ver uma bola no interior a mover-se em direcção a cada uma das paredes, como se pode ver na figura 6.1. A outra aplicação consistia num pequeno sistema de marcação de pontos de referência numa imagem carregada pelo utilizador. Consequentemente, essa imagem era aplicada sobre um objecto na forma de textura, utilizando os pontos de referência definidos pelo utilizador no passo anterior.

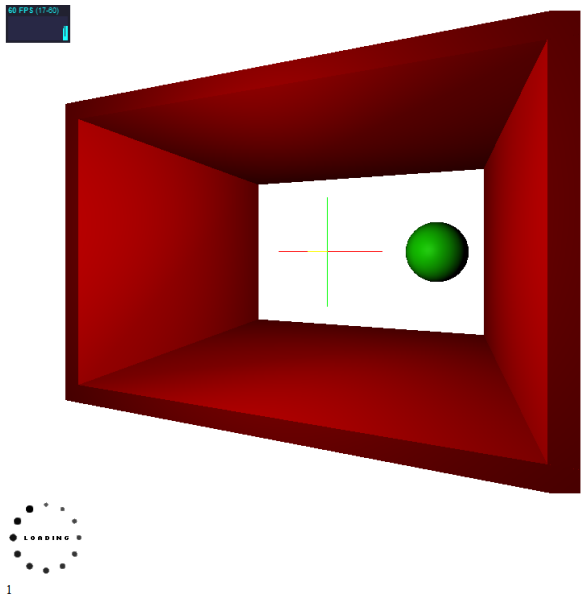

Figura 6.1: Cenário da primeira aplicação desenvolvida

De seguida são apresentadas as pequenas iterações desenvolvidas até à conclusão de cada uma das aplicações.

Animação 3D

O objectivo desta prova de conceito foi apenas o desenvolvimento da animação, ou seja, criar a caixa, a bola e fazer a respectiva animação. Sempre que era detectado contacto com uma das paredes, a bola invertia o sentido.

O desenvolvimento desta pequena aplicação teve como objectivo principal a introdução ao desenvolvimento com a framework *Three.js*[4] como motor de renderização 3D. Foi possível verificar que existem muitas semelhanças entre esta framework e o OpenGL, especialmente na constituição da cena, na qual é necessário, à semelhança do OpenGL, definir um ponto de luz para que toda a cena fique visível. Esta característica, só por si, já introduz alguma ideia de realismo, uma vez que, na realidade, sem qualquer tipo de iluminação a capacidade visual humana é quase nula.

Animação 3D com áudio síncrono

Depois de concluída a animação, foi necessário verificar a sincronização do áudio com a animação, visto esse ser um ponto fulcral do trabalho final.

[4]Biblioteca 3D para JavaScript. Forma de interacção com WebGL - `https://github.com/mrdoob/three.js`

De forma a testar e verificar o delay entre o áudio e a animação, foi adicionado um evento que reproduzia um som cada vez que a bola entrasse em contacto com uma das paredes. Numa primeira abordagem, a sincronização não estava a ser bem conseguida em todos os contactos e após alguns segundos a aplicação falhava. Depois de alguns testes foi possível verificar que isso se devia ao facto de o som estar a ser carregado para a memória do *browser* a cada contacto, ou seja, cada vez que a bola batia numa das paredes o som era novamente carregado para a memória do *browser* e depois reproduzido. Facilmente se pode concluir que essa era a razão pela qual a aplicação falhava e, por conseguinte, pode também concluir-se que ao carregar o som a cada choque entre a bola e a parede, estava a ser adicionado tempo extra ao evento, o que originaria o atraso verificado. Ao colocar o carregamento do áudio para fora do ciclo de animação, foi possível resolver os dois problemas, tendo o resultado final sido satisfatório.

Animação 3D com pedido de dados ao servidor

Visto que o trabalho final não será completamente *client-side*, tornou-se necessário testar a aplicação fazendo um pedido ao servidor de alguns dados necessários para configurar o cenário. Assim, foi implementado um sistema de pedidos AJAX que pedia ao servidor o sentido inicial da bola e as dimensões da bola e das paredes. Assim que a resposta é recebida, a animação começa, da forma já referida no passo anterior.

Outra das funcionalidades desta pequena aplicação é a contagem de ciclos (a bola bater nas duas paredes). Essa contagem é feita do lado do servidor, sendo o número do ciclo actual mostrado na página da animação e actualizado a cada ciclo.

Animação 3D com pedido de áudio ao servidor

A aplicação final será integrada com um software TTS que deverá correr num servidor. Assim, foi de grande importância testar qual o atraso que poderia haver no caso do áudio não ser gerado do lado do cliente, mas pedido ao servidor.

A aplicação era em tudo semelhante à descrita anteriormente, sendo que a diferença estava nos pedidos ao servidor. Na versão anterior era feito apenas um pedido no arranque da aplicação. Neste caso existia um pedido a cada ciclo de choques da bola. De cada vez que se completava um ciclo era feito um pedido ao servidor de forma a obter o áudio que seria utilizado no ciclo seguinte.

Nesta simulação foi possível verificar que a animação continuava antes de ser obtida a resposta do servidor, o que originava dessincronização do áudio. A resolução deste problema foi obtida forçando uma pausa na animação até obter a resposta do servidor. Esta solução não compromete o objectivo da aplicação final, visto que a animação da fala com sincronização labial só irá começar depois de ter do lado do cliente a resposta do servidor, não existindo assim risco de assincronismo.

Animação 3D com atraso na resposta do servidor

De forma a testar a aplicação para os casos em que o servidor demorasse a responder ou as condições da rede atrasassem a resposta, foi introduzido um tempo de espera aleatório antes do servidor responder, de forma a que a aplicação cliente tenha necessariamente que esperar pela resposta.

Esta prova de conceito revelou-se de extrema importância, visto que foi necessário adaptar o código de forma a que a animação não continuasse sem ter recebido a resposta do servidor. Foi introduzida uma pausa forçada na animação dentro do pedido AJAX, pausa essa que só era libertada quando a resposta do servidor chegava.

Teste da animação 3D em ambiente de pré-produção

Neste passo do desenvolvimento, foram retirados todos os tempos de espera introduzidos para o teste anterior e foi criada uma *pool* de ficheiros áudio, de onde o servidor escolhia aleatoriamente um, a cada ciclo de animação. Esse áudio era enviado como resposta ao pedido AJAX. De forma a aproximar o teste o mais possível do ambiente real, a aplicação foi testada a correr num servidor externo. Como era esperado, devido aos resultados e aos ajustes realizados no teste anterior, a aplicação teve o comportamento correcto, não mostrando atraso na execução do som.

Upload de imagem, marcação de pontos chave e serialização dos pontos

Esta prova de conceito foi a primeira iteração no desenvolvimento da segunda aplicação. O objectivo era criar um sistema de *upload* de imagens que posteriormente seriam marcadas com os pontos de referência pretendidos. A informação relativa aos pontos deveria ser posteriormente guardada de forma permanente num ficheiro ou numa base de dados.

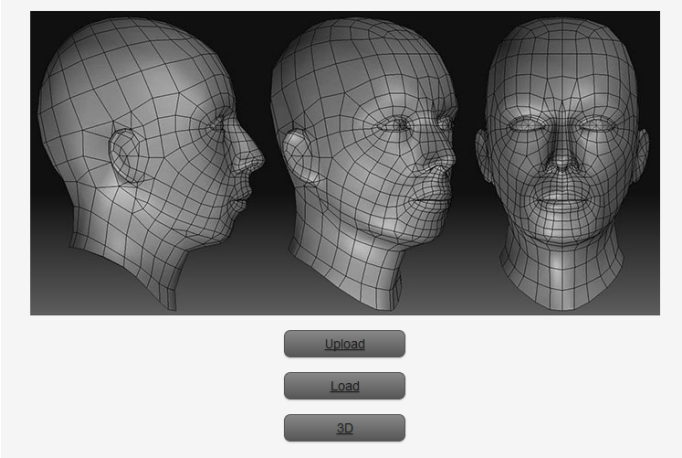

Figura 6.2: Página inicial da segunda aplicação desenvolvida

O sistema de *upload* foi desenvolvido utilizando as capacidades do MVC[5], visto que torna o sistema muito menos complexo. A marcação de pontos foi feita recorrendo ao script javascript. Os dados correspondentes a cada ponto foram posteriormente gravados num ficheiro XML com o mesmo nome da imagem à qual se referem.

As principais dificuldades, nesta pequena aplicação, foram sentidas aquando do desenvolvimento da funcionalidade de gravação do ficheiro XML. Visto ser um formato muito específico, as primeiras abordagens não obtiveram o resultado desejado.

[5]Model, View, Controler

No entanto, após consulta de alguma documentação, foi relativamente simples resolver a questão, uma vez que o MVC já integra uma API para esse efeito.

O ficheiro XML criado é uma representação do objecto existente na aplicação, em que cada propriedade é representada, no ficheiro, por uma *tag* com o respectivo valor.

Carregamento da imagem, criação e aplicação de textura com base nos pontos previamente definidos

No seguimento da prova de conceito anterior, era esperado nesta fase que fossem lidos os dados guardados no ficheiro XML e a imagem respectiva, e que fosse criada uma figura na qual fosse aplicada a imagem como textura, tendo em consideração os pontos previamente marcados (que seriam as margens da textura).

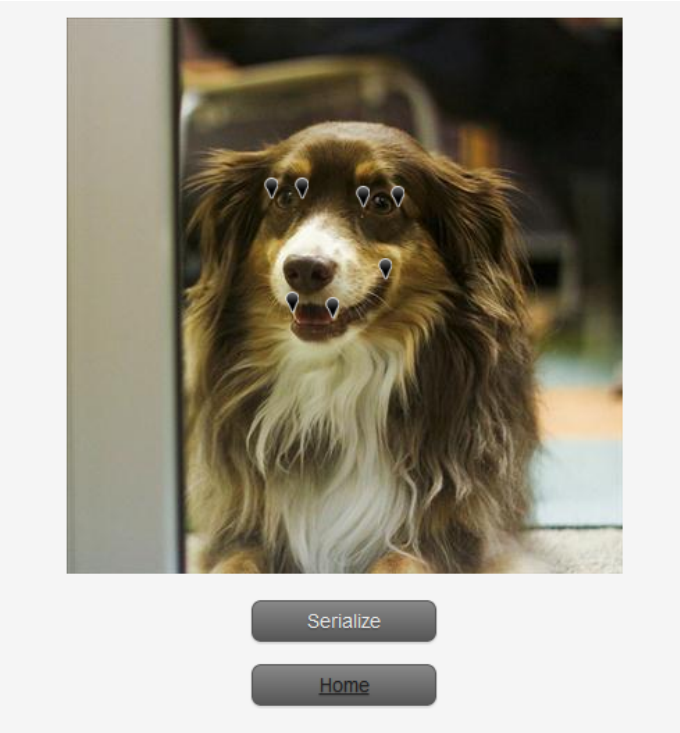

Figura 6.3: Sistema de marcação de pontos de referência

De forma a fazer uma aprendizagem incremental da framework *Three.js*, seguiu-se uma abordagem crescente no que respeita à complexidade das tarefas, ou seja, numa primeira abordagem foi criada uma forma 3D e nela foi aplicada uma cor simples, tendo sido realizados alguns testes com diferentes tipos de materials (lambertmaterial, normalmaterial e phongmaterial). Depois dos testes realizados foi substituída a cor simples por uma textura sem nenhum mapeamento específico, sendo apenas uma imagem aplicada numa face. Concluído o passo anterior com sucesso, a última etapa era a aplicação de uma textura com um mapeamento definido pelos pontos guardados no ficheiro XML. Este passo foi o que mais dificuldades apresentou, sendo também o que mais tempo levo a ser concluído.

Para aplicar a textura segundo as especificações do utilizador, foi necessário definir explicitamente que coordenadas da imagem iriam ser mapeadas para os vértices

da face. Este mapeamento foi feito recorrendo a coordenadas UV, ou seja, coordenadas de textura. Nesse mapeamento foi definido que coordenadas da textura correspondiam a cada coordenada do modelo, sendo assim possível aplicar parte de uma imagem como textura.

Figura 6.4: Aspecto final da prova da aplicação de textura a um objecto

Durante o desenvolvimento desta prova de conceito verificaram-se algumas diferenças entre o sistema de coordenadas do modelo e da textura. O sistema do modelo, por defeito, tem a sua origem situada no centro do primeiro objecto a ser desenhado, enquanto que o sistema de eixos da textura tem origem no canto superior esquerdo da imagem. Outro aspecto que importa referir é que as coordenadas da textura estão definidas entre 0 e 1.

Desenvolvimento de ambiente com múltiplas camadas de modelos

O objectivo principal desta aplicação foi o desenvolvimento de objectos desenhados a várias profundidades, ou seja, no final, estes objectos estarão uns em frente dos outros. Este teste tem uma utilidade muito prática para o trabalho final, uma vez que o modelo facial vai ter três camadas de objectos (a face, os dentes e a língua), de forma a que a simulação seja mais realista. Assim, foram desenhadas várias faces com diferentes profundidades, de forma a umas esconderem as outras. O desenvolvimento ocorreu sem problemas.

6.2 SUIWA

O desenvolvimento do SUIWA envolveu a integração de várias especialidades, desde a anatomia e a psicologia até à computação gráfica.

De seguida são apresentados alguns pontos de particular interesse no desenvolvimento da aplicação.

6.2.1 Marcação de características faciais

Um dos passos mais importantes na criação de um modelo 3D personalizado é a marcação de pontos chave, tais como olhos, boca e nariz, bem como a delimitação da área da cara.

Existem várias opções técnicas para alcançar o objectivo que é ter a marcação de todos os pontos necessários para a construção e animação do modelo. Maurel *et al.* apresenta no seu trabalho uma técnica de sobreposição e ajuste do modelo à imagem de referência[45], enquanto Zhang *et al.* adopta uma técnica de marcação dos pontos chave de forma individual e sequencial[46].

Enquanto a técnica utilizada por Maurel se pode tornar visualmente mais confusa, a adoptada por Zhang dá ao utilizador menos liberdade e menos noção do resultado final do ajuste.

Não comprometendo a simplicidade o sistema, optou-se por dar privilégio à liberdade e facilidade do utilizador em ajustar o modelo. A técnica de ajuste desenvolvida utiliza a sobreposição de um modelo pré-definido sobre uma foto de referência, como na figura 6.5.

A utilização desta técnica liberta o utilizador de uma sequência fixa de ajuste do modelo, podendo este retocar qualquer sector a qualquer momento, não necessitando de refazer ajustes previamente feitos, como aconteceria no caso da marcação sequencial.

Numa primeira fase é utilizada uma *framework* Javascript que implementa *OpenCV* para reconhecimento facial, que faz uma aproximação automática da localização da face e a sobreposição do modelo predefinido.

A localização da moldura onde é desenhado o modelo genérico pode ser alterada, bastando para isso arrastá-la. Esta funcionalidade é possível graças ao atributo *draggable* de HTML5. Embora faça parte da nova especificação de HTML, esta funcionalidade já está disponível na última versão de todos os *browsers*.

O ajuste do modelo é conseguido através de um painel de ajuste à base de *sliders*, tal como mostra a figura 6.6. A opção por esta ferramenta deve-se ao facto de ser mais simples para o utilizador arrastar uma barra e ver, em tempo real, o resultado do que movimentar cada ponto do modelo de forma independente.

O incremento a aplicar pelo movimento de cada *slider* tem em conta não só o objectivo específico desse *slider*, como também o sentido em que se faz esse movimento. Assim, o sentido positivo é aquele que se faz da esquerda para a direita, no caso de *sliders* horizontais, e de baixo para cima, no caso de *sliders* verticais, sendo o negativo, em ambos os casos, o inverso.

Cada *slider* tem associado um factor de movimento. Caso se pretenda um ajuste mais fino é aplicado um factor multiplicativo κ de 0.003, caso contrário esse factor terá o valor de 0.005. Esse factor, multiplicado pela diferença entre a posição inicial e final do *slider*, define o valor do factor de incremento δ, a somar ás componentes

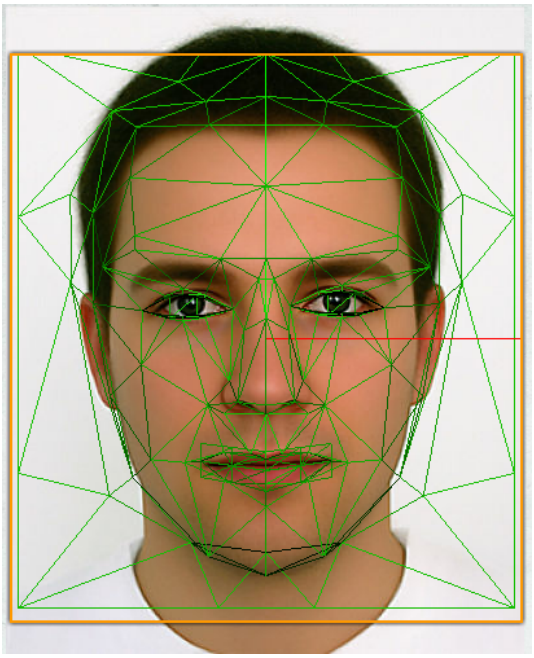

Figura 6.5: Marcação de pontos chave por sobreposição de modelo genérico

Figura 6.6: Sistema de ajuste do modelo genérico

dos pontos afectados por esse *slider*.

$$\delta = (posF - posI) * \kappa$$

Os valores 0.003 e 0.005 foram obtidos por experimentação e são aqueles que permitem um ajuste mais exacto do modelo. Estes valores apresentam-se também como uma boa opção uma vez que não limitam o espectro de ajuste, ou seja, o utilizador tem grande liberdade de ajuste quer seja para faces mais pequenas quer seja para faces maiores.

Em termos de código existem modificadores gerais e locais, sendo que a diferença entre eles é a abrangência das suas modificações. Enquanto que os gerais modificam a totalidade do modelo, os modificadores locais apenas afectam áreas específicas.

A face humana, além de grandes semelhanças tem também grandes diferenças, o que implica que a aplicação tenha que ter a capacidade de ajustar o modelo genérico a essa diversidade. Assim, foram implementados dois níveis de modificadores gerais, como se pode ver na figura 6.7.

Do primeiro nível fazem parte a rotação em torno do eixo Z, a escala no eixo X, a escala no eixo Y e a escala em X e Y da área circundante à face. Estas ferramentas dão ao utilizador a capacidade de efectuar os grandes ajustes, como a definição da área da face, ou da área que deseja considerar como fundo, para, em ambiente de animação contextualizar a face da personagem.

Do segundo nível, embora sejam também considerados gerais, fazem parte aqueles modificadores que ajustam de forma um pouco mais fina algumas características já afectadas pelos modificadores de primeiro nível, como é o caso da largura do queixo, a altura da testa ou o tamanho das bochechas. Estas características, embora sejam consideradas gerais, visto definirem a forma da face, são utilizadas para tornar cada face única.

Figura 6.7: Sliders de ajuste geral divididos em dois níveis

Ao nível de modificadores específicos, estes, por questões de organização e facilidade de utilização, estão divididos em quatro grupos: sobrancelhas, olhos, nariz e boca.

Ao nível das sobrancelhas e dos olhos, como se pode verificar pelas figuras 6.8 e 6.9, os modificadores são semelhantes. Para cada uma dessas características é

possível ajustar a posição vertical, a posição horizontal, a largura e a altura. Uma vez que pode não existir simetria entre o lado esquerdo e o lado direito da face (seja por questões de fisionomia ou devido à foto), cada modificador é constituído por um conjunto de dois sistemas de ajuste, esquerda e direita, respectivamente.

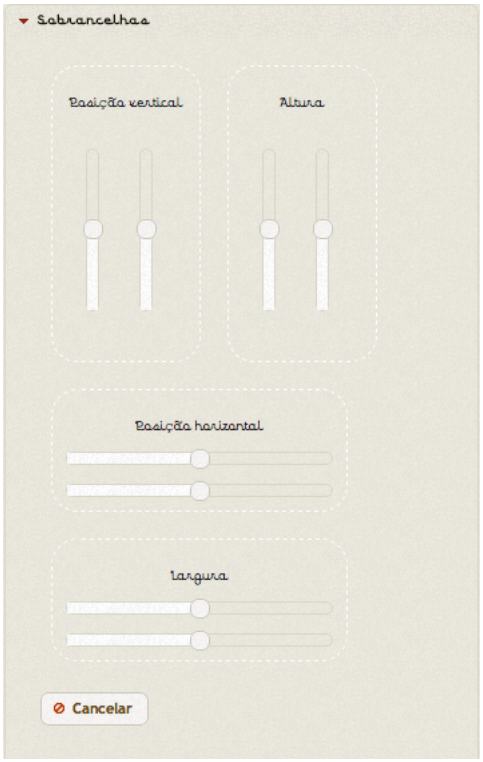

Figura 6.8: Sliders de ajuste das sobrancelhas

Nos *sliders* verticais a associação entre esquerda e direita é imediata, naqueles que se apresentam na horizontal, o superior corresponde ao lado esquerdo, enquanto que o inferior ao lado direito.

O nariz, sendo uma característica não tão proeminente neste projecto, tem um conjunto de ajustes mais simples, sendo ainda assim possível alterar o seu comprimento, a sua posição vertical e a posição vertical da ponta.

Ao nível da boca, e sendo esta utilizada numa parte fundamental do projecto, existe um vasto conjunto de opções de ajuste. Ao nível de posicionamento é possível alterar a sua localização em relação ao eixo X e ao eixo Y. O seu tamanho também pode ser alterado, sendo possível modificar tanto a altura como a largura. Existe ainda a opção de modificar o seu ângulo, ou seja, alterar a posição do canto da boca, tanto à esquerda como à direita.

Como se pôde verificar em todas as figuras anteriores, é possível anular as alterações efectuadas em determinada componente, utilizando para isso o botão "Cancelar".

6.2.2 Mapa UV

Uma textura 2D, para ser aplicada a um objecto 3D, necessita de um mapa que defina a forma como essa aplicação será feita (mapa UV).

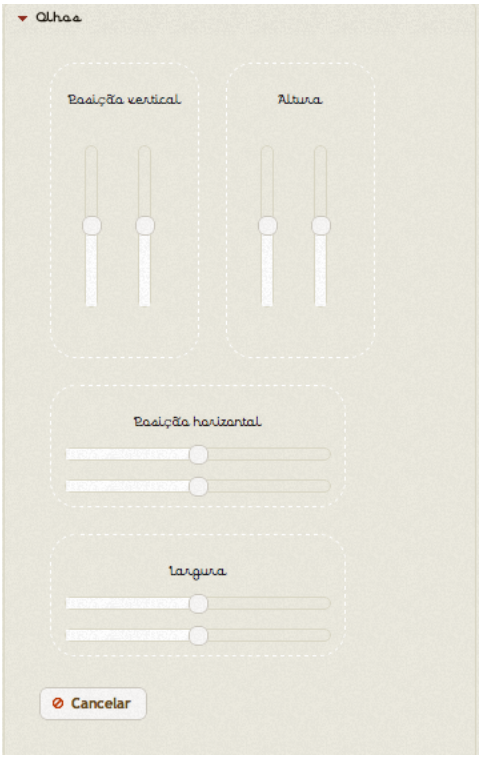

Figura 6.9: Sliders de ajuste dos olhos

Figura 6.10: Sliders de ajuste do nariz

O sistema de coordenadas utilizado para descrever objectos 3D (X, Y, Z) é diferente do sistema de coordenadas utilizado para descrever a colocação e transformação dos mapas (U, V). U é o equivalente a X (direcção horizontal do mapa); V é o equivalente a Y (direcção vertical do mapa)[47].

Em objectos simples como cubos ou esferas este mapa já se encontra definido, no entanto em objectos complexos, como é o caso de um modelo facial, é necessário criar esse mapa para que as texturas fiquem devidamente aplicadas[47].

Neste projecto, a forma utilizada para criar esse mapa tem como ponto de partida as coordenadas dos vértices do modelo de referência.

Este mapeamento, além de ser uma transformação de um sistema 3D para um sistema 2D é também uma alteração no sentido do eixo vertical e no ponto inicial do sistema de coordenadas, tal como se pode verificar pelas imagens 6.12 e 6.13.

O mapeamento de coordenadas do modelo para o mapa UV é constituído por

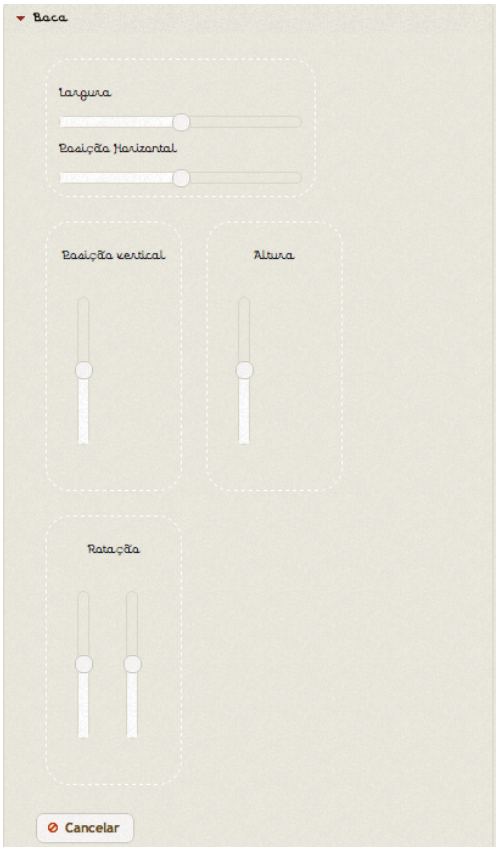

Figura 6.11: Sliders de ajuste da boca

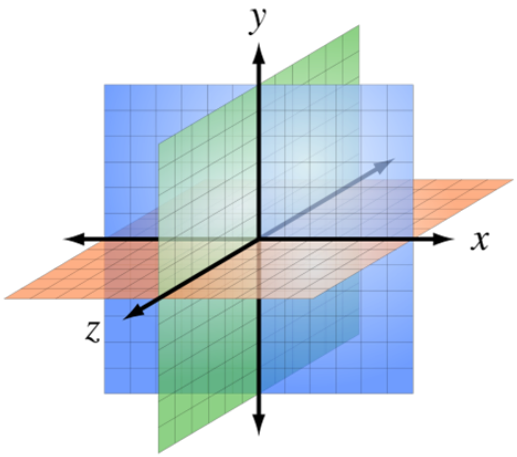

Figura 6.12: Sistema de eixos do modelo 3D

vários passos. O primeiro deles é a projecção das coordenadas 3D do modelo em coordenadas do *veiwport*. Essa projecção foi calculada com a função *ProjectVector()* do *Three.js*. Para efeitos de compreensão da explicação seguinte assuma-se que α é a projecção da coordenada x e β é a projecção da coordenada y.

Tal como referido anteriormente existem diferenças significativas entre os sistemas de coordenadas de origem e de destino, como tal, são necessários alguns ajustes, de forma a eliminar essas diferenças. Tendo em conta a inversão do sentido do eixo

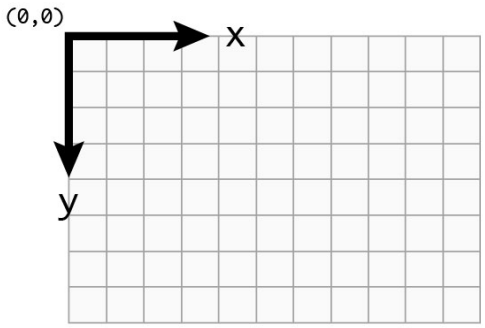

Figura 6.13: Sistema de eixos do browser

vertical, deve ser então utilizado nos cálculos relativos ao eixo vertical o simétrico do valor da projecção correspondente ($\beta negativo$). Desta forma, uma das diferenças está já resolvida.

Outro problema é a localização da origem, que no sistema de origem se apresenta no centro da área de "desenho", enquanto no sistema de destino se encontra no canto superior esquerdo. Tendo em conta que os valores de α e β estão compreendidos no intervalo [-1, 1], ao ser adicionado 1 ao valor da projecção o intervalo anterior passa a ser [0, 2].

O próximo passo é mapear as coordenadas, agora definidas no intervalo [0, 2], para coordenadas do *browser*, mais especificamente para coordenadas dentro do elemento *canvas*. Para isso o valor já calculado é multiplicado pelo tamanho do elemento *canvas*, seja em termos de largura ou de altura. Neste caso concreto é necessário dar especial atenção à largura e à altura do elemento, visto que terá que ser dividido por 2, de forma a compensar o facto das coordenadas estarem definidas entre [0, 2].

Uma vez que o elemento *canvas* está sobreposto à imagem de referência, e aquilo que pretendemos são coordenadas dessa imagem, o próximo passo é adicionar às coordenadas já calculadas um *offset* tanto horizontal, representado por σ, como vertical, representado por τ, de forma a que as coordenadas passem a utilizar como referencial o referencial da imagem.

Todos estes passos resultam em duas fórmulas, uma para a coordenada horizontal, representada por μ, e outra para a coordenada vertical, representada por ν.

$$\mu = (\alpha + 1) * (canvasWidth/2) + \sigma$$
$$\nu = (-\beta + 1) * (canvasHeight/2) + \tau$$

O cálculo dos *offsets* é apresentado na formula seguinte. Como o objectivo é calcular o *offset* do elemento *canvas* em relação à imagem de referência, o primeiro passo é calcular o *offset* de cada elemento em relação à origem e posteriormente subtrair ao *offset* do elemento *canvas* o da imagem de referência. Dessa forma, temos que σ é o *offset* horizontal e τ o vertical.

$$\sigma = divOffsetTop - picOffsetTop$$
$$\tau = divOffsetLeft - picOffsetLeft$$

Por fim, e como os mapas UV apenas utilizam coordenadas no intervalo [0, 1], é necessário efectuar uma nova conversão. Deste modo, basta dividir as coordenadas

já calculadas pela componente respectiva da imagem, ou seja, a coordenada x é dividida pela largura da imagem de referência e a coordenada y é dividida pela altura da mesma imagem.

$$\mu = \mu / picWidth$$
$$\nu = \nu / picHeight$$

6.2.3 Simulação de discurso e emoções

Os módulos de simulação de discurso e emoções são dois módulos chave da aplicação. No entanto, nem tudo o que é necessário para esse módulos cumprirem com os seus objectivos faz parte do âmbito do projecto. Enquanto a simulação de discurso depende de uma aplicação externa para criar o áudio e identificar os fonemas e o respectivo tempo, a simulação das emoções depende da identificação externa da emoção em causa.

Uma vez que se pretendem simular dois conjuntos distintos de expressões faciais, e a bibliografia consultada[10][7][11] apresenta dois modelos distintos de animação facial, foi tomada a opção de implementar cada um dos conjuntos com um método diferente e de posteriormente os comparar.

Discurso

A simulação de discurso, tal como referido anteriormente, depende da transformação de texto em áudio e da identificação dos fonemas e do respectivo tempo de ocorrência. Para esse fim foi utilizada a aplicação IVONA TTS[6]. Esta aplicação é executada a partir da linha de comandos. Dessa forma foi implementado no servidor um método que executa a aplicação e que faz o *parse* da resposta respectiva, enviando para o cliente a informação acerca do ficheiro áudio e dos visemas.

No cliente, a animação é criada em tempo real, ou seja, à medida que o áudio é reproduzido a personagem é transformada.

Como existem expressões faciais comuns a vários fonemas, o cliente faz também um pequeno *parse* dos fonemas que recebe, transformando essa informação, num dos visemas implementados.

Uma parte fundamental na simulação de discurso é o sincronismo entre a componente visual e a componente auditiva. A forma escolhida para manter esse sincronismo foi a confrontação entre o tempo actual de reprodução do áudio e o *time stamp* de cada visema. Caso o tempo actual seja igual ou superior ao *time stamp* do visema, esse visema é apresentado e passa para o visema seguinte, que só será mostrado quando o tempo de reprodução do áudio for novamente igual ou superior ao *time stamp* do visema actual. Esta função está dentro do ciclo de animação, pelo que, só pára a sua execução quando não existirem mais visemas para apresentar.

Em relação aos visemas, estes foram definidos tendo como base o trabalho de Neto[9], que identifica os quinze visemas essenciais para Português de Portugal, tal como referido no capítulo 2.

Estes visemas, além de definidos para a simulação de discurso em Português de Portugal, foram também a base utilizada para a simulação de discurso em Português do Brasil e Inglês Britânico. A utilização destes visemas para as duas novas línguas

[6]Mais informação em `http://www.ivona.com/en/`

foi possível uma vez que, segundo John C. Wells, a resposta visual a determinado som é semelhante em qualquer idioma.

A adição destas línguas à capacidade de simulação de discurso deveu-se ao facto de a aplicação de TTS utilizada não ter ainda disponível a língua Portuguesa de Portugal.

A implementação utilizou as AU definidas por Ekman, Friesen e Hager[11][7]. Embora Ekman e Friesen tenham definido um conjunto de 46 AU, apenas 17 foram implementadas[8]. A opção de implementar apenas uma parte das AU em detrimento da totalidade foi tomada, uma vez que para as necessidades da aplicação algumas delas não se revelaram necessárias.

Além das AU, a implementação dos visemas foi complementada com movimento dos dentes superiores e inferiores, criando desta forma uma simulação mais complexa e realista.

Em relação às AU, estas foram definidas segundo conjuntos de vértices que ao serem movimentados em conjunto produziam o efeito documentado em[11]. A relação entre cada uma das AU implementadas com os respectivos vértices do modelo pode ser consultada no apêndice G.

Como as AU não são uni-direccionais, existe um sistema de pesos, que permitem intensificar ou suavizar determinada AU numa expressão facial. Este sistema de pesos permite também inverter a acção de uma AU, basta para isso dar-lhe um peso negativo.

A simulação de discurso, no interface com o utilizador, caracteriza-se por ter um local para inserção de texto e um pequeno painel de controlo, onde é possível escolher qual o idioma que será utilizado na transformação e qual o género da voz utilizada, como apresentado na figura 6.14.

Figura 6.14: Interface com o utilizador do módulo de simulação de discurso

[7]A lista completa de AU pode ser consultada no apêndice F.
[8]A lista de AU implementadas pode ser consultada no apêndice G.

Emoções

No que à simulação de emoções diz respeito, o objectivo é apenas a simulação da sua componente visual, estando fora do âmbito do projecto a identificação da emoção a simular. Como tal, a forma de visualizar a expressão facial associada a cada emoção é a escolha manual da emoção pretendida.

A simulação de emoções foi desenvolvida com base em FAPs[9], ao contrário da simulação de discurso, que foi desenvolvida com base em AU.

Embora na sua génese sejam muito semelhantes, ambas representando movimentos da face humana, as FAPs representam movimentos elementares, geralmente ao nível de músculos isolados, enquanto as AU representam movimentos mais complexos, tipicamente de grupos de músculos.

Para simular uma emoção fazendo uso das FAPs é necessário identificar as áreas da face afectadas por essa emoção. A obra de Ekman e Friesen[21] foi essencial nesse processo, na medida em que identifica muito claramente as transformações faciais existentes em cada área da face para cada uma das seis emoções base de Ekman. Tendo os movimentos nas diversas áreas da face identificados, basta identificar quais as FAPs que os simulam.

As FAPs, como as AU, têm um sistema de pesos que definem, para determinada emoção, qual a intensidade dessa FAP na expressão final.

Além dos pesos, as FAPs têm ainda um sistema de medidas, tal como já referido no capítulo 2, que permite que, uma vez implementadas, possam ser utilizadas em qualquer modelo, desde que este seja compatível com a norma MPEG-4.

Cada FAP é representada por um conjunto de vértices e pela indicação da unidade de medida a utilizar[10]. Para simular uma emoção, é necessário, além da identificação das FAPs, o cálculo das medidas[11] para controlo do movimentos e a definição do peso que cada FAP terá na face final.

A simulação de emoções pela personagem está dependente de um módulo de reconhecimento de emoções, que, embora se encontre em fase de desenvolvimento, não fez parte dos objectivos deste estágio. Como tal, e como forma de validar as expressões faciais correspondentes a cada uma das emoções pretendidas, foi inserido na aplicação um conjunto de botões, correspondendo cada um deles a uma emoção específica, tal como na figura 6.15.

Figura 6.15: Interface com o utilizador do módulo de simulação de emoções

AU vs. FAPs

Como se pode verificar nos pontos anteriores, existem muitas semelhanças entre AU e FAPs, entre as quais o facto dos dois métodos utilizarem um sistema de pesos para controlo da intensidade da sua acção.

[9] A lista completa de FAPs pode ser consultada no apêndice D.
[10] A lista de FAPs implementadas pode ser consultada no apêndice E.
[11] Apresentado no capítulo 2.

No entanto, existem também diferenças significativas, sendo que a maior é ao nível da acção de cada um deles. Enquanto as AU são mais complexas nos movimentos que representam, as FAPs, por sua vez, são mais simples. Esta diferença, em termos de utilização traduz-se numa maior simplicidade na utilização das AU, mas numa maior flexibilidade na utilização das FAPs.

A liberdade obtida com a utilização das FAPs pode ser um pouco enganadora, uma vez que essa liberdade permite a definição de movimentos irreais. Embora as AU também o permitam, é uma situação mais controlada, uma vez que as restrições de movimentos também se aplicam aos movimentos irreais.

Em suma, embora sejam muito semelhantes, as FAPs têm a vantagem de fazer parte do *standard* MPEG-4 e darem muita liberdade na definição dos movimentos faciais. Por sua vez as AU ganham vantagem no que toca à facilidade de definição de expressões faciais, uma vez que cada AU contempla expressões mais complexas.

6.2.4 Aumento de realismo

Uma forma de aumentar, para o utilizador, o realismo de uma personagem humana animada é a atribuição de características e acções tipicamente humanas a essa personagem. Para alcançar esse objectivo foram desenvolvidos dois módulos, um que simula o piscar de olhos e outro que altera o foco de atenção da personagem. Com estes módulos pretende-se que o utilizador tenha uma maior sensação de interacção com uma personagem real.

Piscar de olhos

Enquanto humanos, estamos muito familiarizados com o contacto visual, logo, o acto de olhar para os olhos daqueles com quem estamos é automático, daí a importância deste módulo.

A forma encontrada para simular o piscar de olhos no SUIWA é uma simulação daquilo que acontece na realidade com o ser humano, ou seja, as pálpebras superiores deslocam-se até tocarem nas pálpebras inferiores.

No ser humano existe um excesso de pele que permite esse movimento. Na simulação esse movimento implica que a textura que simula a pálpebra seja esticada.

Dessa forma foi definida uma transformação ao modelo, que, utilizando as FAP19[12] e FAP20[13], simula o fechar e abrir dos olhos.

O piscar de olhos humano respeita dois intervalos de tempo, um para o movimento de fecho e outro para o movimento de abertura. Doughty define que em média se piscam os olhos 10,3 vezes por minuto, com um desvio padrão de 3,1. Isso quer dizer que o ser humano pisca os olhos entre 4,47 e 8,3 vezes por minuto[48]. Como o objectivo é aproximar o mais possível a personagem da realidade, foi definido um temporizador cuja função é controlar o fecho dos olhos. Este faz piscar os olhos em intervalos aleatórios compreendidos entre [4, 8] segundos.

Estando o fechar dos olhos já definido, falta ainda definir a sua abertura. Assim, o temporizador associado define quanto tempo vão estar os olhos fechados. Trutoiu *et al.* definem que, a cada piscar de olhos, se fica de olhos fechados entre 50 e

[12] close_t_l_eyelid - Ver apêndice D.
[13] close_t_r_eyelid - Ver apêndice D.

150 milisegundos[49]. Assim foi definido na aplicação um novo temporizador que respeita este intervalo.

Alteração do foco de atenção da personagem

Outra forma de adicionar realismo à personagem é adicionar-lhe movimento. Se o movimento parecer autónomo e espontâneo mais efeito produz. Assim, foi implementado um módulo de movimento que é activado pelo movimento do rato.

O objectivo deste módulo é que a personagem siga o rato,alterando o foco de atenção quando este se movimenta, rodando a cabeça.

O rato tem dois eixos em que se movimenta, sendo que em cada eixo esse movimento se faz quer no sentido positivo quer no sentido negativo.

A personagem, para seguir o rato, tem implementado o mesmo sistema, guiado por dois eixos e fazendo distinção entre o sentido do movimento.

Para a detecção da direcção e sentido do movimento é calculada a diferença entre a posição actual e a posição anterior do rato. Caso esse valor seja diferente de 0 no eixo X, há movimento nesse eixo. Caso essa diferença seja positiva, o movimento está a ser feito no sentido positivo do eixo, caso contrário o movimento é feito no sentido negativo. O mesmo acontece para o eixo Y.

Outro passo importante na implementação deste módulo é o cálculo do ângulo de rotação (α). Aqui a opção tomada foi a de, por cada pixel de deslocamento do rato, rodar a personagem α graus.

O valor de α é uma fracção do valor de π. Ao utilizar π como referência considera-se que o movimento da personagem apenas se faz em metade de uma circunferência, isto porque não existe a necessidade de a personagem olhar para trás, apenas para a frente e para os lados.

Para a animação ser suave e o mais precisa possível é necessário que a unidade mínima de movimento seja pequena. Assim dividiu-se π pelo produto de uma constante calculada por experimentação pela dimensão de metade da janela, como na formula seguinte.

$$\alpha = \pi/(const * windowCenter)$$

Ao introduzir o tamanho da janela nos cálculos está-se, de alguma forma, a forçar que a personagem tenha uma amplitude de movimentos semelhante, seja qual for o tamanho da janela. Está-se ainda a garantir que não rode por excesso nem por defeito.

A rotação em si é conseguida pela formula seguinte. No entanto, quando o movimento se faz no sentido negativo o valor α é substituído por $2\pi - \alpha$. Em termos de posição final é igual ter $-\alpha$ ou $2\pi - \alpha$, no entanto tanto o sin com o cos apresentam valores distintos em cada um dos casos, sendo o valor que se pretende dado por $\sin(2\pi - \alpha)$ e $\cos(2\pi - \alpha)$.

$$newX = oldX * \cos\alpha - oldY * \sin\alpha$$
$$newY = oldY * \cos\alpha + oldX * \sin\alpha$$

6.2.5 Avaliação e validação

Para verificar o cumprimento dos objectivos definidos, foram realizados alguns testes, uns de caracter mais objectivo, como desempenho da animação, outros de

caracter mais subjectivo, como um inquérito à expressividade das faces.

Expressividade facial

De forma a avaliar o realismo das expressões faciais definidas foi realizado um inquérito on-line, sendo que os resultados, depois de analisados, levaram a alterações de algumas expressões faciais representativas de emoções.

O inquérito foi realizado numa plataforma online[14] e foi "publicitado" quer em redes sociais quer em listas de email.

Este inquérito teve uma amostra aleatória de 246 indivíduos, na sua maioria residentes em Portugal, no entanto, cerca de 6% das respostas tiveram origem exterior.

Os resultados obtidos foram muito importantes, visto terem permitido o refinamento de algumas expressões que não obtiveram consenso.

Pela análise das figuras 6.16 e 6.17 é possível verificar que, nas faces 1 e 2, a emoção "tristeza" e "raiva", respectivamente, reuniram consenso entre os inquiridos, face às restantes opções. Embora com alguns ajustes adicionais os resultados pudessem ser mais favoráveis, as faces 1 e 2 apresentam já alguma capacidade de transmitir as emoções pretendidas.

Quanto às faces seguintes, é possível verificar algumas tendências de associação das faces 3 e 5, a "surpresa" e "raiva", respectivamente. No entanto, os valores estão muito abaixo do valor de aceitação, e como tal concluiu-se que havia a necessidade de uma intervenção mais profunda ao nível da expressividade destas faces.

Na face 4 a dificuldade de identificação da expressão é clara, basta para isso verificar a percentagem de respostas *"Não sabe"*.

Assim, pela análise dos resultados, pode concluir-se que algumas das expressões faciais submetidas a inquérito não são de fácil identificação pelo utilizador, pelo que se revela necessário redefinir áreas chave como é o caso dos olhos e da boca.

Uma razão possível para a dificuldade de identificação das expressões 3, 4 e 5 poderá residir no tipo de análise realizada, ou seja, este inquérito apresentou aos inquiridos apenas uma face, sem qualquer tipo de informação acerca da resposta motora associada. Essa situação, em conjunto com o facto de, no dia-a-dia, analisarmos uma pessoa pela sua resposta global, não só pela expressão facial, podem estar na origem das dificuldades de identificação das emoções em causa.

Outra razão a considerar, para a dificuldade de identificação das expressões 3, 4 e 5, é o facto destas serem, em certa medida, semelhantes, o que pode ter criado alguma dificuldade extra na sua identificação.

Após a análise dos resultados do inquérito foram redefinidas as expressões faciais que apresentaram menos consenso. O resultado dessa evolução das expressões faciais pode visto na figura 6.18.

Estas expressões, foram sujeitas a uma avaliação por parte de todos os membros da equipa de desenvolvimento da Innabler, S.A. Mesmo não reunindo consenso entre todos os colaboradores acerca da expressão presente em cada uma delas, a resposta foi unânime no que dizia respeito à evolução em relação à versão anterior.

[14] `http://www.freeonlinesurveys.com/`

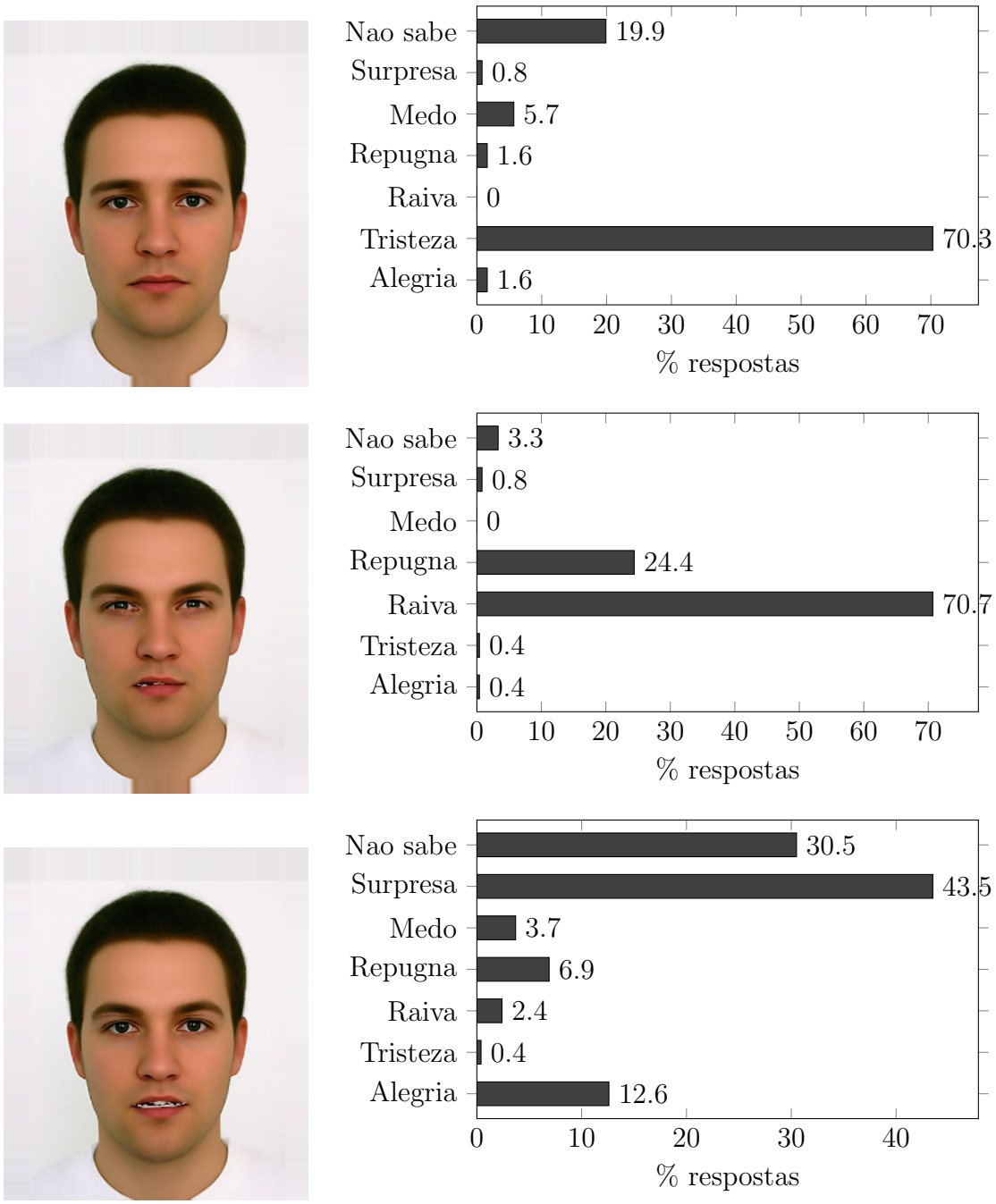

Figura 6.16: Expressões faciais e respostas das perguntas 1, 2 e 3 respectivamente

Desempenho

Uma das formas de avaliação da aplicação passa pela avaliação do desempenho da animação. Uma das métricas utilizadas para avaliar esse desempenho é o número de fotogramas por segundo que o *browser* é capaz de reproduzir.

Para obter o número de fotogramas por segundo foi utilizada uma classe Javascript chamada *stats*[15]. Esta classe permite visualizar, em tempo real, quer o número de fotagramas por segundo quer o tempo de renderização de cada fotograma. Permite ainda identificar o intervalo (valor máximo e mínimo) que determi-

[15]Mais informação disponível em `https://github.com/mrdoob/stats.js/`

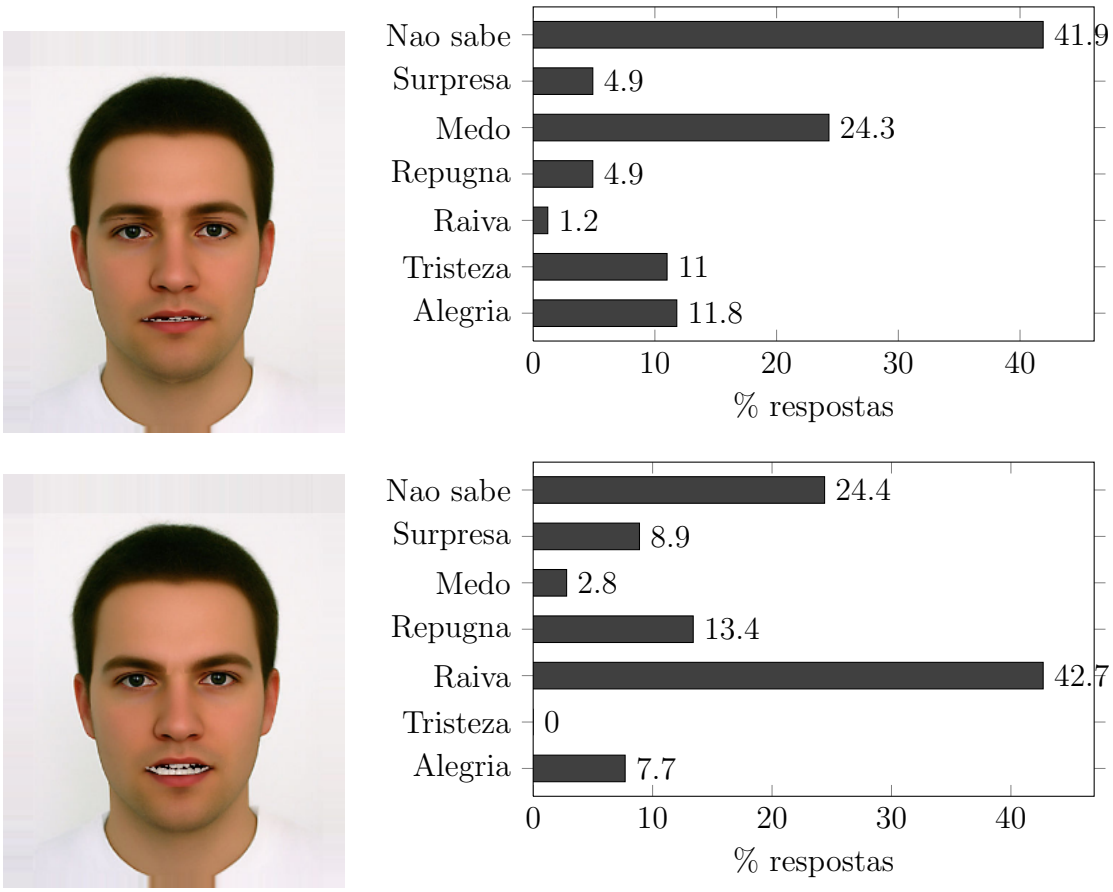

Figura 6.17: Expressões faciais e respostas das perguntas 4 e 5 respectivamente

Figura 6.18: Expressões faciais de alegria e surpresa depois de redefinidas

nada animação atinge.

Na tabela G.1 é visível que, do conjunto de *browsers* analisados, apenas o Internet Explorer™não suporta a aplicação. Dentro daqueles que a suportam, é possível verificar que não existem diferenças muito significativas quanto ao intervalo de fps (frames por segundo), exceptuando o caso do Safari™que apresenta valores um pouco inferiores.

Embora o olho humano não tenha capacidade de distinguir animações com mais de 30fps o cérebro tem. Embora não sejam diferenças muito significativas, quanto

Tabela 6.1: Suporte dos diferentes browsers à aplicação SUIWA

	Versão	Suporte SUIWA	Desempenho
	v9	Não[a]	-
	v12	Sim[b]	58-182 *fps*
	v5	Sim[c]	42-92 *fps*
	v13	Sim	60-176 *fps*
	v19	Sim	60-149 *fps*

[a] Sem informação acerca de suporte em versões futuras
[b] Necessita de activação explicita de WebGl e de aceleração por hardware
[c] Apenas em Mac OS X e com activação explicita de WebGL

maior o número de fotogramas por segundo mais detalhe os objectos animados apresentam, ou seja, a animação é fluida seja a 60fps ou a 120fps, no entanto a 120fps o denominado *motion blur* é mais reduzido.

Dessa forma, a diferença existente entre o Safari™e os restantes *browsers* acaba por se revelar quase imperceptível em termos visuais quando se utiliza a aplicação.

Uma questão que importa referir é que tanto o Opera™como o Safari™, embora suportem WebGl e aceleração gráfica por hardware, necessitam que o utilizador active explicitamente essas funcionalidades, ao contrário do Chrome™e do Firefox™, em que está activo por omissão.

Os valores apresentados na tabela G.1 foram obtidos numa máquina com processador Intel Core™i7-2600k, 8Gb de memória RAM e placa gráfica com 1Gb de memória dedicada, a correr o sistema operativo Windows7 Profissional™, excepto o *browser* Safari, que foi testado num MacBook Pro com processador Intel Core 2 Duo de 2.4Ghz, 4 Gb de memória RAM e uma placa gráfica com 256Mb de memória dedicada.

Devido às características da máquina de testes, não foram registadas alterações significativas quando estavam mais aplicações em execução.

Numa outra máquina, um portátil, com processador Core 2 Duo de 1.6Ghz, 3Gb de memória RAM e uma placa gráfica com 512Mb de memória dedicada, com o sistema operativo Windows7 Home Premium™, os resultados, sem carga, não apresentam diferenças significativas, como se pode verificar na tabela 6.2.

Nos testes efectuados em carga, o peso das outras aplicações em execução rondou os 60% da capacidade de processamento. Essa carga de processamento adicional foi reflectida de forma ligeira no desempenho da aplicação, que manteve desempenhos sempre satisfatórios.

Outra métrica importante na avaliação da animação é o número de visemas descartados de forma a manter o sincronismo. Nessa medida foi implementado um contador que contava o número de visemas mostrados. Esse valor foi depois

Tabela 6.2: Desempenho da aplicação SUIWA em diferentes sistemas

	Versão	Máquina de testes sem carga	Portátil sem carga	Máquina de testes em carga	Portátil em carga
	v12	58-182 *fps*	53-175 *fps*	55-182 *fps*	48-168 *fps*
	v13	60-176 *fps*	55-176 *fps*	54-175 *fps*	51-170*fps*
	v19	60-149 *fps*	60-149 *fps*	61-145 *fps*	53-141 *fps*

confrontado com o número total de visemas enviados pelo servidor, de forma a ter a razão entre o total de visemas e o número de visemas mostrados.

Em todos os testes realizados se concluiu que todos os visemas eram apresentados. No entanto, em alguns casos, o tempo que estavam visíveis era mais reduzido, de forma a minimizar a propagação de eventuais atrasos.

Capítulo 7

Considerações finais e trabalho futuro

Neste capítulo, além de algumas considerações acerca do trabalho desenvolvido, será também apresentado o que se espera desta aplicação para o futuro. Além disso, serão também apresentadas algumas propostas de aplicação do SUIWA.

7.1 Considerações finais

Este estágio teve dois momentos distintos de desenvolvimento. O primeiro, coincidente com o primeiro semestre, correspondeu ao desenvolvimento de algumas provas de conceito, que serviram principalmente para testar as capacidades 3D dos *browsers* e algumas funcionalidades consideradas fundamentais, como é o caso da sincronização entre o áudio e a animação.

No segundo momento foi iniciado o desenvolvimento de um assistente pessoal virtual 3D, cuja aparência fosse muito realista. No decorrer do segundo semestre, e tendo em conta o conhecimento já adquirido com as provas de conceito do primeiro semestre, foi desenvolvida a aplicação SUIWA.

Para conseguir uma aparência final realista e consistente com a imagem de referência, utilizou-se essa imagem como textura do modelo 3D, modelo esse personalizado de acordo com a imagem de referência fornecida pelo utilizador.

A personalização do modelo foi conseguida por meio de *sliders*, em que cada um deles é responsável por editar uma característica específica do modelo.

A fase de animação do modelo foi conseguida com recurso a duas técnicas distintas, uma para simulação de discurso(AUs) e outra para a simulação de emoções(FAPs).

Para a simulação de discurso foi necessário recorrer a uma aplicação externa (IVONA TTS) que fizesse a transformação de texto em áudio, e que, em simultâneo, extraísse informação necessária para a apresentação das expressões faciais correspondentes ao áudio.

O desenvolvimento foi marcado por alguns desafios, entre os quais a não utilização de *plugins* e o sincronismo entre a animação e a reprodução de áudio. Todos eles, em conjunto com o facto de existir um *feedback* visual do trabalho realizado, foram uma grande motivação para o desenvolvimento do projecto.

7.2 Trabalho futuro

Neste momento, o SUIWA já tem dois sistemas que conferem um pouco mais de realismo à animação, são eles: o piscar de olhos e o seguir o movimento do rato. No entanto, se o utilizador estiver muito tempo sem movimentar o rato, a personagem permanece imóvel, o que não acontece com um humano.

Uma funcionalidade que lhe iria conferir mais algum realismo seria o movimento espontâneo. Este módulo teria a capacidade de movimentar a personagem no caso de não ser detectado movimento durante algum tempo.

Outro ponto a ter em conta é a migração para o segmento móvel. Dado que actualmente já existe uma grande quantidade de dispositivos móveis que permitem acesso à Internet, seria interessante considerar para o futuro a migração para esse segmento. A migração poderia ser tanto da solução integrada, actualmente em desenvolvimento na empresa, como do SUIWA por si só. Esta migração acarretaria algumas questões, em relação à geração do áudio pela aplicação de TTS, por exemplo. No caso dessa geração ser feita do lado do servidor, implicaria algum tráfego adicional para transferir o áudio gerado. No caso da geração de áudio ser feita no próprio terminal deve ser avaliada a viabilidade de gerar tanto o áudio como a informação de visemas.

Como já foi referido anteriormente, esta aplicação será integrada numa de maior dimensão, vocacionada para a vertente empresarial. Nesta fase, a aplicação "hospedeira" está a cerca de 40% do seu desenvolvimento, estando já prevista a integração do SUIWA num futuro próximo. Além das funcionalidades previstas de resposta a perguntas, o desenvolvimento de um módulo de IA(Inteligência artificial) que permita ao SUIWA adquirir conhecimentos específicos baseando-se em dados inseridos no restante sistema seria uma mais valia. Um exemplo seria no caso da marcação de uma reunião. Ao pedir ao assistente para marcar uma reunião, este analisaria as marcações já registadas no calendário e avaliaria da disponibilidade ou não para a nova marcação.

Bibliografia

[1] F. Parke and K. Waters, *Computer Facial Animation*. A. K. Peters, Ltd., 2nd ed., 2008.

[2] J. Sobotta, R. Pabst, and R. Putz, *Sobotta: Atlas da anatomia humana*, vol. 1: Cabeça, pescoço e estremidade superior. Guanabara Koogan, 20 ed., 1995.

[3] G. B. Duchenne, *The mechanism of human facial expression*. Cambridge University Press, 1862.

[4] B. R. Steunebrink, M. Dastani, and J.-J. C. Meyer, "The occ model revisited," *4th Workshop on Emotion and Computing Paderborn Germany*, 2009.

[5] R. Plutchik, *Emotion: A psychoevolutionary synthesis*. Harper & Row, 1980.

[6] J. Nunes, L. Sá, and F. Perdigão, "Talking avatar for web-based interfaces," in *EUROCON - International Conference on Computer as a Tool*, pp. 1–4, 2011.

[7] *ISO/IEC 14496-1 Visual*, 2 ed., 2001.

[8] J. Ahlberg, "Candide 3 - an updated parameterised face," Tech. Rep. LiTH-ISY-R-2326, Dept. of Electrical Engineering, Linköping University, Sweden, 2001.

[9] J. P. Neto, R. Cassaca, M. Viveiros, and M. Mourão, "Design of a multimodal input interface for a dialogue system," in *Proc. PROPOR 2006*, vol. 3960, Springer, 2006.

[10] I. Pandzic and R. Forchheimer, *MPEG-4 Facial Animation. The Standard, Implementations and Aplications*. John Wiley and Sons, Ltd., 2002.

[11] P. Ekman, W. V. Friesen, and J. C. Hager, *Facial action coding system*. Research Nexus division of Network Information Research Corporation, 2002.

[12] I. Pandzic, "Talking virtual characters for the internet," in *ConTel*, (Croatia), 2001.

[13] F. I. Parke, "Computer generated animation of faces," Master's thesis, University of Utah, USA, 1972.

[14] I. Pandzic and G. Sannier, "From photographs to interactive visual characters on the web," in *Proceedings Scanning 2000*, 2000.

[15] I. Pandzic, "Facial animation framework for the web and mobile platforms," in *Proceedings of the International conference on 3D web technology*, 2002.

[16] I. Pandzic, D. Millen, and J. Ostermann, "Synthetic faces: what are they god for?," in *The Visual Computer*, 1999.

[17] H. Gray and P. L. Williams, *Gray's anatomy*. Churchill Livingstone, 37 ed., 1989.

[18] F. I. Parke, *A parametric model for human faces*. PhD thesis, University of Utah, 1974.

[19] S. Garchery, A. Egges, and N. Magnenat-Thalmann, "Fast facial animation design for emotional virtual humans," in *Proc. Measuring Behaviour, Wageningen, N.L.*, 2005.

[20] R. R. Cornelius, *The science of emotion. Research and tradition in the psychology of emotion*. Prentice-Hall, Inc., 1996.

[21] P. Ekman and W. V. Friesen, *Unmasking the face*. USA: Prentice-Hall, Inc., 1975.

[22] K. Waters, "A muscle model for animating three-dimensional facial expression," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, vol. 21 of *SIGGRAPH '87*, (USA), pp. 17–24, ACM, 1987.

[23] Z. Ruttkay, J. Hendrix, P. ten Hagen, A. Lelièvre, and H. N. abd Behr de Ruiter, "A facial repertoire for avatars," in *Workshop on Interacting Agents*, pp. 27–46, 2000.

[24] J. Faustino, A. P. Cláudio, and M. B. Carmo, "Faces - biblioteca de expressões faciais," in *Actas da 2ª Conferência Nacional em Interacção Pessoa-Máquina*, Interacção 2006, pp. 139–142, 2006.

[25] A. Arya and S. DiPaola, "Socially communicative characters for interactive applications," *International Conference on Computer Graphics, Visualization and Computer Vision*, 2006.

[26] A. Arya, S. DiPaola, and A. Parush, "Perceptually valid facial expressions for character-based applications," in *International Journal of Computer Games Technology*, vol. 2009, 2009.

[27] H. Schlosberg, "The description of facial expressions in terms of two dimensions," *Journal of exprimental psychology*, 1952.

[28] C. G. Fisher, "Confusions among visually perceived consonants," *Journal of Speech and Hearing Research*, vol. 11, pp. 796–804, 1968.

[29] S. M. Platt, "A system for computer simulation of the human face," Master's thesis, The Moore School, University of Pennsylvania, 1980.

[30] I. Albrecht, J. Haber, K. Kahler, M. Schroder, and H.-P. Seidel, "May i talk to you? :-) - facial animation from text," in *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, 2002.

[31] X. Mao and Y. X. Zheng Li, "Emotional gaze behaviour generation in human-agent intereaction," *ACM*, 2009.

[32] T. Koda and P. Maes, "Agents with faces: the effects of personification of agents," in *Procedings of HCI'96*, 1996.

[33] M. W. Vannier, J. F. Marsh, and J. O. Warren, "Three-dimensional computer graphics for craniofacial surgical planning and evaluation," in *Proceedings SIGGRAPH '83*, 1983.

[34] W. Larrabee, "A finite element model of skin deformation. i. biomechanics of skin and soft tissue: A review," *The Laryngoscope*, vol. 96, no. 4, pp. 399–405, 1986.

[35] V. Orvalho, J. Miranda, and A. A. Sousa, "Facial sinthesys of 3d avatars for therapeutic applications," in *The 14th Annual CyberTherapy and CyberPsychology Conference*, 2009.

[36] I. Pandzic, J. Ostermann, and D. Millen, "User evaluation: synthetic talking faces for interactive services," *The Visual Computer*, vol. 15, pp. 330–340, 1999.

[37] N. Magnenat-Thalmann and D. Thalmann, *Handbook of virtual humans*. John Wiley and Sons, Ltd., 2004.

[38] V. T. AB, "Mpeg-4 fba. an overview.," tech. rep., Visage Technologies AB, 2007.

[39] M.Rydfalk, "Candide, a parameterized face," Tech. Rep. LiTH-ISY-I-866, Dept. of Electrical Engineering, Linköping University, Sweden, 1987.

[40] B. Welsh, *Model-Based Coding of Images*. PhD thesis, British Telecom Research Lab, 1991.

[41] D. Herron, *Node Web Development*. Packt Publishing, 2011.

[42] K. Group, "Webgl specification." `https://www.khronos.org/registry/webgl/specs/1.0/`, 2011.

[43] M. Doob, "Three.js." `https://github.com/mrdoob/three.js/`, Abril 2012.

[44] J. Shaw, "Web application performance testing — a case study of an on-line learning application," *BT Technol J*, vol. 18, no. 2, 2000.

[45] P. Mourel, A. MacGonigal, R. Keriven, and P. Chauvel, "3d model fitting for facial expression analysis under uncontrolled imaging conditions," *2008 19th International Conference on Pattern Recognition*, pp. 1–4, 2008.

[46] Y. Zhang, Q. Ji, Z. Zhu, and B. Yi, "Dynamic facial expression analysis and synthesis with mpeg-4 facial animation parameters," *IEEE Transactions on cirtuits and systems for video technology*, vol. 18, no. 10, pp. 1383–1396, 2008.

[47] E. Azevedo and A. Conci, *Computação gráfica. Teoria e prática*. Editora Campus, 2003.

[48] M. J. Doughty, "Further assessment of gender- and blink pattern-related differences in the spontaneous eyeblink activity in primary gaze in young adult humans," *Optometry & Vision Science*, vol. 79, no. 7, pp. 439–447, 2002.

[49] L. C. Trutoiu, E. J. Carter, I. Matthews, and J. K. Hodgins, "Modeling and animating eye blinks," *ACM Transactions on Applied Perception*, vol. 2, no. 3, 2011.

[50] D. Crockford, *JavaScript the good parts*. O'Reilly, 2008.

[51] S. Stefanov, *JavaScript Patterns*. O'Reilly, 2010.

[52] M. Pilgrim, *HTML5: Up and Running*. O'Reilly, "010.

[53] Z. M. Gilenwater, *Stunning CSS3: A project-based guide to the latest in CSS*. New Riders, 2011.

[54] J. Osipa, *Stop staring*. Wiley Publishing, Inc., 3 ed., 2010.

[55] S. R. Marschner, B. Guenter, and S. Raghupathy, "Modeling and rendering for realistic facial animation," in *Eleventh Eurographics Rendering Workshop*, 2000.

[56] A. Arya and S. DiPaola, "Face modeling and animation language for mpeg-4 xmt framework," in *IEEE Transactions on Multimedia*, vol. 9, pp. 1137–1146, 2007.

[57] Z. Ruttkay, H. Noot, and P. ten Hagen, "Emotion disc and emotion squares: tools to explore the facial expression space," *Computer Graphic Forum*, vol. 22, no. 1, pp. 49–54, 2003.

[58] R. W. H. Lau, F. Li, T. L. Kunii, B. Guo, B. Zhang, N. Magnenat-Thalmann, Sumedha, D. Thalmann, and M. Gutierrez, "Emerging web graphics standards and technologies," *IEEE Comput. Graph. Appl.*, vol. 23, pp. 66–75, 2003.

[59] A. Raouzaiou, N. Tsapatsoulis, K. Karpouzis, and S. Kollias, "Parameterized facial expression synthesis based on mpeg-4," in *EURASIP Journal on Applied Signal Processing*, vol. 2002, pp. 1021–1038, 2002.

[60] T. Weise, H. Li, L. V. Gool, and M. Pauly, "Face/off: Live facial puppetry," *Eurographics*, 2009.

[61] D. Cosker, E. Krumhuber, and A. Hilton, "Perception of linear and nonlinear motion properties using a facs validated 3d facial model," in *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, APGV '10, pp. 101–108, ACM, 2010.

[62] J. Mihalik, "Modeling of human head surface by using triangular b-splines," *Radioengineering*, vol. 19, no. 1, pp. 39–45, 2010.

[63] J. Chen, "Automatic face animation with linear model," tech. rep., Rensselaer Polytechnic Institute, Troy, NY, 2008.

[64] Z.-L. Sun and K.-M. Lam, "Depth estimation of face images based on the constrained ica model," *IEEE Transactions on information forensics and security*, vol. 6, no. 2, pp. 360–370, 2011.

[65] M. David, *HTML5: Designing Rich Internet Applications*. Visualizing the Web, Focal Press, 2010.

[66] I. Kemelmacher-Shlizerman and R. Basri, "3d face reconstruction from a single image using a single reference face shape," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 394 – 405, 2011.

[67] I. Kemelmacher-Shlizerman and S. M. Seitz, "Face reconstruction in the wild," in *ICCV*, 2011.

[68] S. Garchery and N. Magnenat-Thalmann, "Designing mpeg-4 facial animation tables for web applications," in *Multimedia Modeling 2001*, pp. 39–59, 2001.

[69] Z. Liu, Y. Shan, and Z. Zhang, "Expressive expression mapping with ration imagex," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '01, pp. 271–276, ACM, 2001.