



## Criação de Modelos Faciais 3D Através de Fotografias

JOÃO FILIPE DOS SANTOS LEAL

Outubro de 2017

# **Criação de Modelos Faciais 3D Através de Fotografias**

**João Filipe dos Santos Leal**

**Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática, Área de Especialização em  
Sistemas Gráficos e Multimédia**

**Orientador: João Paulo Pereira**

**Co-orientador: Prof. Doutor António Vieira Castro**

Porto, outubro 2017



*A família e amigos*



# Resumo

Nos dias de hoje, a qualidade da semelhança entre as personagens em videojogos e o seu correspondente em carne e osso está cada vez mais elevada. A necessidade de uma modelação rápida, eficaz e económica é então cada vez mais requisitada. A solução que melhor conjuga essas 3 qualidades é sem dúvida a modelação facial através de fotografias.

O primeiro artigo publicado nesta área foi em 1999, e têm sido feitas muitas melhorias aos sistemas desenvolvidos, no entanto, ainda existem muitas falhas. Falhas quer em termos de qualidade, semelhança e tempo. Com este projeto de dissertação pretendia-se então combater essas lacunas desenvolvendo uma ferramenta capaz de criar modelos faciais através de fotografias e posterior alteração de diversos parâmetros faciais.

Os objetivos foram maioritariamente cumpridos, apresentando-se então uma biblioteca capaz de criar modelos 3D a partir de uma fotografia com um rosto, e alterar expressões faciais no mesmo, bem como manipular visualmente o modelo e gravá-lo para um ficheiro. Os utilizadores que testaram a aplicação gráfica apreciaram a sua rapidez na criação no modelo, no entanto não se mostraram tão satisfeitos relativamente à quantidade de ferramentas disponíveis para a sua edição.

**Palavras-chave:** modelação facial 3D, identificação facial, deteção características faciais, reconstrução facial, 3DMM, *blendshapes*, interface gráfica



# Abstract

Nowadays, the similarity quality between video game characters and their real-life counterpart is getting higher and higher. The demand for fast, efficient and cheap 3D modeling is being highly required. The solution that best combines these aspects is undoubtedly the modeling through the use of face pictures.

The first article published in this area was in 1999, and many improvements have been made ever since, but there were still many flaws to cover in those systems. The flaws include quality issues, low similarity, and high execution time. Therefore, with this work, it's intended to combat those flaws, by creating a framework capable of creating such 3D models and also adjust various facial parameters.

The majority of the objectives drawn were accomplished, presenting a framework capable of creating 3D models by using a facial picture, and changing their facial expressions, as well as visually manipulating the 3D model and writing it to a file. The users that tested the graphical application appreciated the fast creation of the model, but were not so convinced by the number of editing tools at their disposal.

**Keywords:** 3D facial modeling, facial identification, facial features detection, facial reconstruction, 3DMM, *blendshapes*, graphical interface



# Agradecimentos

Quero em primeiro lugar agradecer ao ISEP, por este percurso de 5 anos, que para além de boas ferramentas para trabalhar me deu bons amigos e muitas memórias

Um grande agradecimento ao meu orientador João Paulo Pereira, por me suportar e ajudar no desenvolvimento da dissertação.

Um agradecimento especial aos meus familiares e amigos por me apoiarem e motivarem a continuar a trabalhar na dissertação quando por vezes tudo parecia falhar.

Um último, um agradecimento a Patrik Huber por disponibilizar uma biblioteca muito útil ao público em geral, e pela simpatia e ajuda nos *e-mails* que trocamos.



# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto	1
1.2	Problema	2
1.3	Objetivo	3
1.4	Resultados Esperados	3
1.5	Análise de Valor	3
1.6	Abordagem Preconizada	4
1.7	Organização do Documento	4
<b>2</b>	<b>Contexto e Estado da Arte</b>	<b>5</b>
2.1	Detalhes sobre o Contexto e o Problema	5
2.1.1	Processo de Reconhecimento Facial	5
2.1.2	Prototipagem de Máscaras em Personagens de Filmes	5
2.1.3	Criação de Modelos de Personagens Reais	6
2.1.4	Criação do Avatar à Própria Imagem num Videojogo	6
2.2	Análise de Valor	6
2.2.1	Modelo de Desenvolvimento de Novos Conceitos	6
2.2.2	Valor	8
2.2.3	Proposta de Valor	9
2.2.4	Modelo <i>Canvas</i>	9
2.3	Estado da Arte	10
2.3.1	Categorias de Modelação Facial 3D	10
2.3.2	Métodos e Artigos Publicados na Área de Modelação Facial 3D	13
2.4	Tecnologias	16
2.4.1	OpenCV	16
2.4.2	DLib	16
2.4.3	Visual Studio	17
2.4.4	Qt	17
2.5	Conclusão	17
<b>3</b>	<b>Design da Solução</b>	<b>19</b>
3.1	Arquitetura	19
3.2	Modelação	20
3.3	Interface Gráfica	22
<b>4</b>	<b>Implementação da solução</b>	<b>23</b>
4.1	Modelo 3DMM	23
4.1.1	Identificação de Pontos Faciais	23
4.1.2	Geração do Modelo	24

4.1.3	Criação do Modelo de Cor.....	24
4.1.4	Rotação e <i>Zoom</i> do Modelo 3D.....	26
4.1.5	Alteração de Expressões Faciais no Modelo 3D.....	26
4.1.6	Alteração de Parâmetros Faciais .....	27
4.2	Geração da biblioteca .....	28
4.3	Interface Gráfica .....	28
4.3.1	Configurações Iniciais.....	28
4.3.2	Interface .....	28
4.3.3	Criação do Modelo.....	32
<b>5</b>	<b>Avaliação .....</b>	<b>35</b>
5.1	Avaliação do Tempo de Execução do Algoritmo .....	35
5.2	Avaliação do Protótipo por Utilizadores.....	36
5.3	Conclusão .....	40
<b>6</b>	<b>Conclusões Finais.....</b>	<b>41</b>
6.1	Balanco do Trabalho Desenvolvido .....	41
6.2	Trabalho Futuro.....	42
6.2.1	Manipulação do Modelo 3D .....	42
6.2.2	Melhoria Modelo 3D .....	42
6.2.3	Acabamento do Modelo 3D .....	42
6.2.4	Otimização .....	42
<b>7</b>	<b>Bibliografia.....</b>	<b>43</b>

# Lista de Figuras

Figura 1: Avatar do jogador Cristiano Ronaldo no jogo FIFA 2016 .....	2
Figura 2: Modelo de negócio <i>Canvas</i> .....	9
Figura 3: <i>Scanner</i> da Cyberware .....	11
Figura 4: Figura ilustrativa do processo descrito em (Park, Zhang and Vezhnevets, 2005) .....	12
Figura 5: Figura ilustrativa do processo utilizado em (Patel and Smith, 2009).....	13
Figura 6: Figura ilustrativa do método utilizado no artigo acima descrito .....	14
Figura 7: Comparação entre os métodos SNO e MFF .....	15
Figura 8: Aplicação do modelo num rosto .....	15
Figura 9: Exemplo de aplicação da DLib.....	17
Figura 10: Arquitetura do sistema .....	19
Figura 11: Diagrama de classes .....	20
Figura 12: Diagrama de sequência da aplicação gráfica a desenvolver.....	22
Figura 13: Excerto de código referente às variáveis da classe Mesh.....	25
Figura 14: Desenho da malha do modelo 3D.....	25
Figura 15: Excerto de código para rodar cada vértice da malha.....	26
Figura 16: Excerto de código para escalar cada ponto da malha .....	26
Figura 17: Excerto de código para criar o novo modelo com a nova expressão facial. ....	26
Figura 18: Excerto de código para encontrar os índices de todos os vértices pertencentes a uma determinada região.....	27
Figura 19: Exemplo execução da aplicação.....	29
Figura 20: Exemplo de rotação do modelo .....	29
Figura 21: Expressão de raiva.....	30
Figura 22: Expressão de nojo .....	30
Figura 23: Expressão de medo .....	31
Figura 24: Expressão de felicidade.....	31
Figura 25: Expressão de tristeza.....	31
Figura 26: Expressão de surpresa .....	32
Figura 27: Criação do modelo 3D.....	33
Figura 28: Exemplo de exportação do modelo 3D para o Blender .....	33
Figura 29: Tempo de execução médio em relação ao tamanho do rosto .....	36
Figura 30: Distribuição de idades dos inquiridos .....	37
Figura 31: Avaliação da semelhança do modelo em relação à foto .....	37
Figura 32: Avaliação da rapidez de criação do modelo.....	38
Figura 33: Avaliação da qualidade final do modelo .....	38
Figura 34: Avaliação da qualidade de manipulação do modelo .....	39
Figura 35: Avaliação da quantidade de ferramentas de manipulação .....	39
Figura 36: Avaliação da utilidade do produto final .....	40



# Lista de Tabelas

Tabela 1: Tempos de execução do algoritmo de criação do modelo facial .....	35
---	----



# Acrónimos e Símbolos

## Lista de Acrónimos

<b>3DMM</b>	<i>3D Morphable Model</i>
<b>BFM</b>	<i>Basel Face Model</i>
<b>MFF</b>	<i>Multi-Features Fitting</i>
<b>SIFT</b>	<i>Scale Invariant Feature Transform</i>
<b>SNO</b>	<i>Stochastic Newton Optimization</i>

## Lista de Símbolos

°	Graus
---	-------



# 1 Introdução

Nesta primeira seção pretende-se contextualizar o tema da dissertação, identificar quais os problemas que levaram à necessidade de soluções, qual o objetivo do projeto e quais os resultados esperados. Irá também conter um pequeno resumo da análise de valor, e uma breve descrição da abordagem pretendida para a resolução do problema.

## 1.1 Contexto

Em alguns videojogos é por vezes oferecida ao jogador a possibilidade de personalizar um avatar (a sua representação virtual no videojogo) segundo vários atributos. Um desses principais atributos é a customização facial, que inclui a cor dos olhos, do cabelo, do tipo e tamanho do nariz, olhos, boca, queixo, orelhas, etc. Quando se pretende criar um avatar com as feições idênticas às do jogador, por mais que se ajustem os parâmetros disponíveis, é muito difícil conseguir alcançar um resultado satisfatório (no mínimo algo com o que o jogador se consiga identificar). É com base no interesse do jogador em ter uma interface que facilite a personalização de um avatar à sua imagem, que surge esta ideia de tese.

Nos dias de hoje, a modelação de personagens é das tarefas essenciais à produção de um jogo, ou de um filme. Tomando por exemplo os jogos de futebol como o FIFA<sup>®1</sup> e o Pro Evolution Soccer<sup>®2</sup>, é possível verificar que todos os jogadores de todas as equipas intervenientes no jogo são modelados com o maior rigor de modo a associar o avatar com o jogador real.

---

<sup>1</sup> <http://www2.ea.com/fifa>

<sup>2</sup> <https://www.konami.com/wepes/2017/eu/en/>



Figura 1: Avatar do jogador Cristiano Ronaldo no jogo FIFA 2016 <sup>3</sup>

Na imagem da Figura 1 é possível observar o detalhe associado ao avatar do jogador Cristiano Ronaldo.

Outro exemplo são os videojogos adaptados de filmes, onde têm de ser modelados os personagens principais da forma mais fiel possível, muitas vezes com base na imagem dos atores que os interpretam.

Um processo que permitiria fazer essa personalização de modo simplificado para o utilizador seria personalizar automaticamente o avatar apenas recorrendo a fotografias do seu rosto.

Sistemas que criam modelos 3D da face com base apenas em imagens 2D têm várias aplicações, tais como a criação automática de personagens reais e avatares de videojogos, fácil prototipagem de máscaras/maquilhagem em personagens de filmes, impressão 3D ou o reconhecimento facial.

Com o avanço rápido das tecnologias é necessário desenhar produtos com qualidade crescente, querendo isto dizer que os modelos terão de ser mais detalhados. Os consumidores também se tornam cada vez mais exigentes, procurando que a representação virtual seja o mais fiel possível. Este contexto faz com que a concorrência aperte e a qualidade visual do jogo/filme tenha de ser cada vez mais elevada.

## 1.2 Problema

A modelação facial 3D pode ser feita manualmente, partindo de um modelo base e ajustando diversos parâmetros até se chegar a um modelo satisfatório; contudo, este método revela-se muito demorado. Uma alternativa passaria pelo uso de um *scanner* 3D, o que implica a presença física da pessoa para a recolha das suas características faciais, sendo, portanto, um método com custos de equipamento e deslocações da pessoa a fazer o *scanning*. Se a modelação for feita apenas com base em imagens do rosto da pessoa, este processo deixa de ser demorado ou

---

<sup>3</sup> Imagem disponível em <https://i.ytimg.com/vi/bihHTg1XBuE/maxresdefault.jpg>

dispendioso, tornando-se claramente vantajoso. Atualmente já existem alguns sistemas que oferecem esta possibilidade ao utilizador, mas estes ainda sofrem de muitos problemas, quer a nível de resultado final (deformação do modelo, fraca semelhança, perdas de informação), quer a nível de quantidade de *input* necessária à criação do modelo (demasiadas imagens necessárias, necessidade de seleção manual de pontos faciais e outros ajustes), quer a nível de tempo de processamento necessário para produzir o modelo (normalmente estes sistemas produzem melhores resultados, mas não podem ser utilizados em sistemas de tempo real).

### **1.3 Objetivo**

O objetivo desta dissertação é o de contribuir para a resolução do problema exposto, contribuição que inclui o desenvolvimento de uma solução para dois setores alvo: modelação automática de avatares dos jogadores em videojogos e criação de um modelo facial 3D para personagens reais.

Para o primeiro pretende-se desenvolver uma ferramenta que seja integrada com o próprio videojogo para permitir ao jogador criar um avatar à sua imagem através de fotografias suas.

Para o segundo alvo pretende-se usar essa mesma ferramenta, mas adicionar-lhe uma camada de interface gráfica para a criação dos modelos faciais, em que também será permitido realizar alguns ajustes ao modelo final.

### **1.4 Resultados Esperados**

Com esta dissertação pretende-se obter um sistema capaz de criar um modelo facial 3D, com apenas uma imagem do rosto do utilizador. Esse sistema será integrado numa *framework* em que será possível além de criar o modelo facial, facilitar a possibilidade de ajustar alguns parâmetros faciais e realizar eventuais modificações desejadas para que o avatar final seja mais satisfatório. É também esperado que a ferramenta permita ser integrada noutras aplicações, nomeadamente na personalização do avatar de jogos, mas poderá ser integrada em qualquer outro tipo de *software*.

### **1.5 Análise de Valor**

O sistema proposto tem como principal mais-valia dar ao consumidor a possibilidade de integrar a modelação automática de um avatar de acordo com as suas características faciais ou usar a mesma ferramenta para criação de modelos faciais 3D baseados em personagens reais com a ajuda de uma interface gráfica.

## **1.6 Abordagem Preconizada**

A abordagem preconizada é o desenvolvimento de um sistema de modelação facial 3D e a sua integração numa *framework*. O sistema será comparado a sistemas existentes para verificar se houve melhorias no resultado final e comprovar a sua aplicabilidade.

## **1.7 Organização do Documento**

Este documento está organizado em vários capítulos da seguinte forma. No capítulo 2 será abordado com mais detalhe o contexto e o problema, realizada uma análise de valor ao produto e descrito o estado da arte e tecnologias. No capítulo 3 será demonstrado através de diagramas, uma proposta da arquitetura e interface da solução. O capítulo 4 irá ter a implementação da solução detalha. No capítulo 5 será avaliado o produto final, e no último capítulo serão dadas algumas conclusões sobre o projeto e perspetivas de trabalho futuro.

## **2 Contexto e Estado da Arte**

Nesta secção irá ser contextualizado o problema, procedendo-se à análise de valor mais detalhada e à descrição do estado da arte na modelação facial.

### **2.1 Detalhes sobre o Contexto e o Problema**

A modelação facial 3D tem imensas aplicações nos dias de hoje. A seguir encontram-se descritos alguns exemplos e qual a sua relação com o âmbito desta tese.

#### **2.1.1 Processo de Reconhecimento Facial**

A modelação facial está a ser aplicada cada vez mais a esta área, dado que com apenas algumas fotografias, é possível modelar a face de um indivíduo, e dependendo dos sistemas utilizados neste processo, a qualidade pode ser muito elevada. A vantagem de utilizar modelação facial 3D para o reconhecimento, é que é obtido o modelo total do rosto, sendo facilitado o reconhecimento da mesma pessoa, numa perspetiva completamente diferente, dado que o resultado do modelo deverá ser muito semelhante.

#### **2.1.2 Prototipagem de Máscaras em Personagens de Filmes**

Esta tarefa é realizada com mais frequência em filmes de ficção científica, onde um personagem fictício é criado a partir de um personagem real. É criada uma máscara para o ator usar, no entanto essa máscara é criada e moldada perfeitamente para o ator específico. Ao ser criado um modelo facial 3D do ator, é possível desenhar mais facilmente uma máscara que se adapte a este com maior perfeição, sendo assim necessário pouca interação com o ator e evitando sucessivas iterações para correção da máscara.

### **2.1.3 Criação de Modelos de Personagens Reais**

Esta é uma das mais importantes tarefas do desenho de um videogame com personagens reais: a sua aparência. A modelação facial 3D automática de personagens reais facilita muito o trabalho dos *designers*, de forma que estes precisam apenas de retocar ligeiramente o modelo para o deixar mais polido e mais semelhante ao modelo real. No entanto, estes sistemas apenas permitem a obtenção de um modelo da face, pelo que o resto do corpo das personagens terá de ser criado através de outras técnicas.

### **2.1.4 Criação do Avatar à Própria Imagem num Videogame**

São inúmeros os videogames que oferecem ferramentas de personalização facial do avatar que irá representar o jogador no decorrer do jogo. No entanto não existe nenhum jogo que permita a personalização automática do avatar através do uso de fotografias. A adição dessa ferramenta iria permitir aos jogadores criarem um personagem semelhante a eles a nível facial.

Apesar de a modelação facial 3D ter outras aplicações, para o desenvolvimento desta tese apenas se irá focar nas duas últimas referidas: criação de modelos de personagens reais para filmes e criação do avatar à própria imagem num videogame.

Para apresentar às empresas de videogames uma solução que permita a personalização do avatar, é necessário desenvolver código específico para cada uma das empresas, dado que cada videogame pode usar um modelo facial diferente, e que suporta diferentes parametrizações.

## **2.2 Análise de Valor**

Nos subcapítulos seguintes vai ser analisado o valor do produto proposto.

### **2.2.1 Modelo de Desenvolvimento de Novos Conceitos**

Com base no modelo de desenvolvimento de novos conceitos (NCD) de Peter Koen, destacam-se neste projeto os seguintes elementos chave (Koen, 2004) (Koen *et al.*, 2002).

#### **2.2.1.1 Identificação de Oportunidade:**

A identificação de oportunidade ocorre quando uma empresa deteta oportunidades para melhoramento numa tecnologia ou produto já existente. Tais oportunidades podem incluir simplificação de operações, maior rapidez, redução de custos ou uma aplicação da tecnologia/produto a outros mercados não explorados.

Algumas das técnicas mais utilizadas são o *design* de um *roadmap* (mapa temporal onde são identificadas as metas de o desenvolvimento de uma tecnologia/produto), análise das tendências da tecnologia e dos consumidores ou pesquisa de mercado.

No planeamento deste projeto foi identificada uma oportunidade no mercado de videojogos, onde não existe uma interface em nenhum jogo que permita efetuar a personalização facial do avatar à imagem do jogador de uma forma simplificada e precisa.

#### 2.2.1.2 Análise de Oportunidade

Na análise de oportunidade, tal como o nome sugere, é feita uma pesquisa mais aprofundada sobre a real necessidade por parte das empresas da ideia proposta e onde se avalia se irá compensar avançar com a ideia.

As metodologias para esta fase são as mesmas do que para a fase anterior, mas o esforço deve ser em dobro e as pesquisas mais detalhadas.

Neste projeto foram identificadas várias falhas a nível de sistemas que se pudessem aplicar ao objeto de estudo da tese. Tais falhas incluíam a baixa qualidade do modelo facial gerado, perda de informação facial, resultados com muito pouca semelhança à imagem real e muito tempo para o processamento. Assim, verifica-se que existe ainda margem para o desenvolvimento de melhores sistemas que vão ao encontro das exigências e necessidades dos utilizadores.

#### 2.2.1.3 Geração de Ideias e Enriquecimento

Esta é a fase em que ocorre o desenvolvimento e a maturação da ideia. Contacto com os consumidores e outras empresas que estejam a desenvolver aplicações na mesma área são de importância fundamental nesta fase. É também aqui que ocorre o *brainstorm*, onde surgem e são debatidas várias ideias e se misturam algumas delas com a ideia principal.

Para se conseguir obter sucesso, é fundamental o envolvimento do consumidor e as suas preferências, a investigação das perspetivas de mercado, identificação de novas tecnologias e troca de experiências e de ideias na rede de trabalho.

Neste projeto, a geração de ideias partiu de uma pesquisa aprofundada sobre técnicas de modelação facial através de imagens e sobre reconstrução de objetos 3D através de imagens.

#### 2.2.1.4 Seleção da Ideia

Esta é uma das fases mais importantes, pois é aqui que são escolhidas quais as ideias que vão fazer avançar o projeto, e com isso a vida e direção do mesmo.

Comparar as metodologias selecionadas em diversos fatores tais como a probabilidade de sucesso técnico e comercial, a recompensa e o enquadramento estratégico é uma das técnicas mais utilizadas nesta fase.

Para este projeto seguir para a fase de desenvolvimento, foram selecionadas algumas das ideias consideradas mais relevantes. As ideias escolhidas permitem mais que uma abordagem para o problema, de modo a testar quais as que obtêm melhores resultados.

#### 2.2.1.5 Definição do Conceito

A última fase consiste na definição do conceito, onde é definido o que é necessário para a solução final, tal como os objetivos, mercado, risco técnico e comercial, recursos, com a

finalidade de ter um documento que justifique o investimento necessário e a passagem para o desenvolvimento da solução.

As metodologias mais usadas nesta última fase são a definição dos objetivos e resultados esperados do projeto, bem como quais são os critérios que descrevem um projeto atrativo.

Neste projeto, o conceito final seria a obtenção de uma ferramenta capaz de criar um modelo facial 3D, com apenas uma imagem da cara do utilizador. A ferramenta deve permitir alguns ajustes do modelo final e além disso deve ser possível integrar a ferramenta num *software* já existente.

## **2.2.2 Valor**

Nos próximos subcapítulos será dada a definição de valor, valor percebido e valor para o cliente, bem como os benefícios, sacrifícios e o valor da solução.

### **2.2.2.1 Valor**

O conceito de valor representa aquilo que o produto ou tecnologia tem para oferecer em troca daquilo que tem de consumir. Se essa troca for positiva, estamos perante algo com valor.

### **2.2.2.2 Valor Percebido**

O valor percebido representa aquilo que o utilizador entende que é o valor do produto. Cada consumidor tem a sua perceção do valor percebido, baseado na perceção do que é dado e recebido na troca de valores.

### **2.2.2.3 Valor para o Cliente**

O valor para o cliente representa a perspetiva que o consumidor tem do produto/tecnologia. São pesados os benefícios (atributos ou resultados) e os sacrifícios (monetários ou materiais) quer a curto, quer a médio prazo.

Neste projeto em específico, pode ter-se em conta o seguinte balanço:

### **2.2.2.4 Benefícios**

- Criação de modelos faciais 3D de uma forma rápida e simples;
- Criação de modelos faciais 3D de boa qualidade;
- Sem mais gastos económicos a cada modelação.

### **2.2.2.5 Benefícios para Desenvolvedores de Jogos com Configuração de Avatar**

- Possibilidade de oferecer aos jogadores ferramentas para a criação de um avatar com as feições à sua imagem.

### **2.2.2.6 Sacrifícios**

- Custo inicial do produto

### 2.2.2.7 Valor do Produto

Numa fase inicial terá que ser sacrificado algum valor monetário para a compra do produto ao qual está associada uma licença de utilização. Depois desse ponto, o consumidor poderá desfrutar de todos os benefícios: usufruir da tecnologia e criar modelos faciais 3D ilimitadamente, ou até incorporar o produto no seu próprio *software* (caso da personalização do avatar no jogo).

### 2.2.3 Proposta de Valor

Pretende-se que este sistema permita a criação de um modelo facial 3D apenas com fotografias do rosto. As empresas desenvolvedoras de jogos podem usar o sistema para permitir ao jogador fazer a personalização facial do avatar à sua imagem. Este sistema também oferece a possibilidade de poder modelar personagens reais de uma forma rápida, simples, sem custos adicionais, e com elevada qualidade.

### 2.2.4 Modelo *Canvas*

Nesta secção será descrito o modelo de negócio de acordo com o modelo *Canvas*.

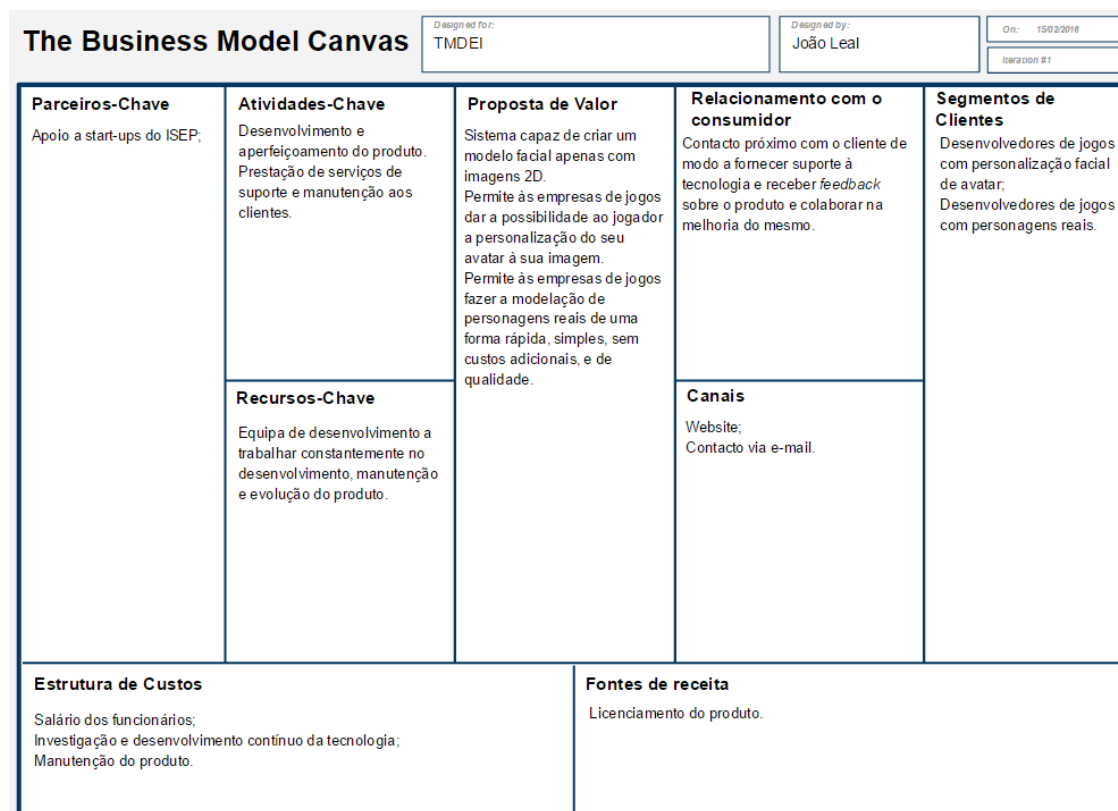


Figura 2: Modelo de negócio *Canvas*

Os Segmentos de Clientes principais da ideia de negócio são desenvolvedores de jogos nos quais há possibilidade de personalização de um avatar, ou onde são utilizados personagens reais.

No Relacionamento com o Consumidor é muito valorizado o contato direto com o cliente, para que se consiga perceber a sua opinião sobre o produto e prestar-lhe suporte. Além disso, também é importante tentar inclui-lo no desenvolvimento e evolução do produto.

Os Canais a ser utilizados serão o *website* onde será dado a conhecer o produto ao consumidor e o *e-mail*, através do qual serão contactados potenciais clientes em específico.

A Proposta de Valor deste projeto é um sistema capaz de gerar modelos faciais 3D com apenas fotografias do rosto. Este sistema poderá ser utilizado na modelação de personagens reais de um jogo, ou integrado na personalização facial do avatar de um jogador. Essa modelação é simples, rápida, sem custos adicionais e de qualidade.

As Atividades-Chave são o desenvolvimento e melhoramento do produto, e os serviços de suporte e manutenção aos clientes.

Os Recursos-Chave são uma equipa de desenvolvimento constantemente a trabalhar no desenvolvimento, melhoria e evolução do produto.

Como Parceiros-Chave, o projeto contará com a colaboração de empresas *start-up* do ISEP.

A Estrutura de Custos irá incluir o salário dos funcionários, gastos com o desenvolvimento e investigação da tecnologia e a manutenção do produto.

As Fontes de Receita que se irão obter resultarão do licenciamento do produto.

## **2.3 Estado da Arte**

Nesta secção serão explicadas as categorias da modelação facial e descritos alguns dos métodos do estado da arte mais relevantes.

### **2.3.1 Categorias de Modelação Facial 3D**

Atualmente existem várias abordagens para tentar solucionar o problema da modelação facial. De seguida serão brevemente abordados os tipos de modelação utilizados atualmente. Esta separação, apesar de não ser absoluta, abrange um ou mais métodos semelhantes numa só categoria, uma vez que alguns métodos diferem de uns para os outros em apenas pequenos pormenores.

#### **2.3.1.1 Modelação Através de uma Interface Gráfica**

Este tipo de modelação é muito primário e muito pouco utilizado hoje em dia, dada a quantidade elevada de tempo necessário para produzir um único modelo. A técnica consiste em

utilizar uma ferramenta de *design* interativa (e.g. Blender) e partir de um modelo facial genérico (ou por vezes sem modelo inicial), até se atingir o resultado desejado.

#### 2.3.1.2 Modelação através de um *Scanner*

Este modelo, como o título sugere, é criado recorrendo a um *scanner* que projeta um laser e roda em volta da face num ângulo de 360°. Além disso também é capturado um vídeo de dois pontos de vista da face (podem ser mais pontos de vista, dependendo do scanner e dos requisitos associados ao método que irá utilizar os seus dados). Este método, além da inevitável deslocação ao local onde irá decorrer a recolha das características, requer a imobilização total do indivíduo durante alguns segundos (menos de 30 segundos, na maioria dos casos). No entanto consegue produzir resultados com alta precisão e qualidade.

Como este método recorre a equipamentos externos e não a imagens para produzir o modelo, não vai ser objeto de estudo ao longo do documento. Optou-se pela sua inclusão apenas por ser uma das técnicas de modelação facial utilizadas atualmente com melhores resultados (Alomari *et al.*, 2011).



Figura 3: *Scanner* da Cyberware<sup>4</sup>

#### 2.3.1.3 Modelação Obtida a Partir de um Modelo Facial Genérico 3D

Sistemas deste género criam o modelo facial do utilizador através da deformação de um modelo facial genérico de modo a encaixar os vértices 3D em alguns pontos faciais retirados de imagens da pessoa (normalmente uma imagem frontal e uma de perfil). Depois dessa modelação primária são levados a cabo alguns procedimentos (esses procedimentos variam conforme o método em questão) para melhorar os resultados obtidos inicialmente.

---

<sup>4</sup> Imagem disponível em <http://cyberware.com/products/scanners/ps.html>

Este método é muito simples e rápido podendo assim ser utilizado em sistemas de tempo real. No entanto não produz resultados muito precisos, dado o número muito disperso dos vértices do modelo<sup>5</sup>. Além disso, o modelo produzido é também um modelo estático, o que implica que não é possível animá-lo, ou pelo menos tem uma animação muito limitada (Alomari *et al.*, 2011).

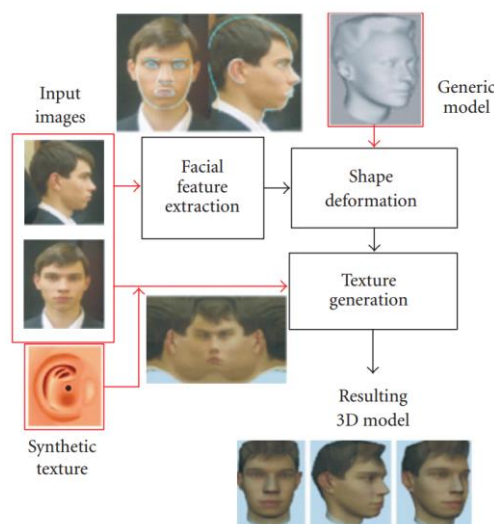


Figura 4: Figura ilustrativa do processo descrito em (Park, Zhang and Vezhnevets, 2005)

#### 2.3.1.4 Modelação com Base num Modelo Morfo 3D

Modelo 3D morfo advém de *3D morphable model* em inglês (doravante designado por 3DMM neste documento) e significa que é um modelo dinâmico, sendo assim possível simular movimentos faciais e expressões (animações), ao contrário do que acontece com os modelos estáticos. Este tipo de modelação é baseado num 3DMM que é composto pela forma do modelo facial e pela textura correspondente (cor da face ao longo do modelo). Esse modelo foi previamente treinado com outros modelos faciais para se chegar a um modelo genérico facial.

A partir desse modelo genérico e com uma imagem da face fornecida, o sistema faz a sua correspondência e é obtido o modelo final do rosto. Apesar de este método produzir resultados muito bons e apenas necessitar de uma imagem do rosto, tem um custo de computação muito elevado, o que o torna inutilizável em sistemas de tempo real<sup>6</sup>(Alomari *et al.*, 2011).

<sup>5</sup> Citação traduzida e adaptada pelo autor. No original: “3D face modeling by deforming a 3D face generic model to fit the 3D vertex of some facial landmarks is relatively simple and fast in processing, but cannot obtain an accurate 3D face model since the adopted 3D generic model has a sparse number of vertices and triangles”.

<sup>6</sup> Citação traduzida e adaptada pelo autor. No original: “[...] for a new input 2D face image, the 3D face modeling method fits the 3D morphable model into the new input 2D face image. The 3D morphable model-based 3D face modelling is very accurate and needs just single 2D face image. To produce a 3D face model, but it costs a prohibitive computation time to generate a 3D face model for real-time processing.”

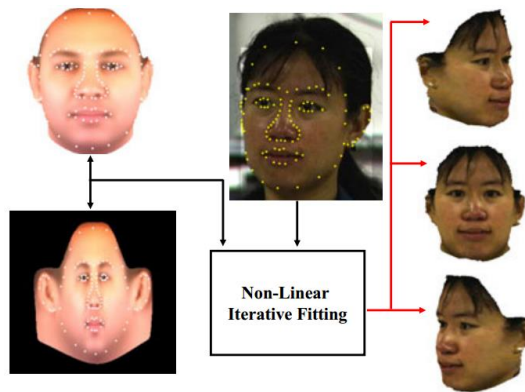


Figura 5: Figura ilustrativa do processo utilizado em (Patel and Smith, 2009)

### 2.3.1.5 Modelação feita com uma Sequência de Imagens

Estes sistemas capturam uma sequência de imagens, e em cada uma das imagens são identificados os pontos faciais, a partir dos quais é construído o modelo facial. Este modelo gerado também é normalmente um 3DMM.

## 2.3.2 Métodos e Artigos Publicados na Área de Modelação Facial 3D

No subcapítulo anterior foram vistas as categorias em que se enquadram a maioria dos métodos atualmente existentes para a modelação facial 3D. De seguida serão analisados e discutidos alguns artigos escritos sobre a modelação facial 3D com maior relevância para esta tese, nomeadamente métodos que trabalhem com 3DMM e não necessitem de uma sequência de imagens.

### 2.3.2.1 *A morphable model for the synthesis of 3D faces* (Blanz and Vetter, 1999)

Escrito e desenvolvido por V. Blanz e T. Vetter em 1999, este foi o primeiro método a utilizar um 3DMM para modelar faces em 3D.

O modelo morfo é extraído a partir de um conjunto de modelos faciais presentes num base de dados de faces previamente digitalizadas por um *scanner*. Para esse modelo ser desenhado é necessário estabelecer uma correspondência entre as múltiplas faces digitalizadas. Essa correspondência é feita a partir da indicação manual de pontos faciais em cada uma das faces (esse número de pontos faciais é variável conforme a aplicação e pode ir de 50 a 300) (Vetter, Jones and Poggio, 1997).

De seguida é feita a correspondência do 3DMM com uma imagem 2D frontal. Inicialmente o modelo é colocado sobre a imagem e manualmente alinhado. Depois é feita a correspondência automática entre o modelo morfo e a imagem 2D (Jones and Poggio, 1998). Por fim é feita uma correção de textura e iluminação ao modelo para o deixar mais realístico (Pighin *et al.*, 1998). Também foi desenvolvida uma ferramenta para modificar o modelo conforme diversos parâmetros.

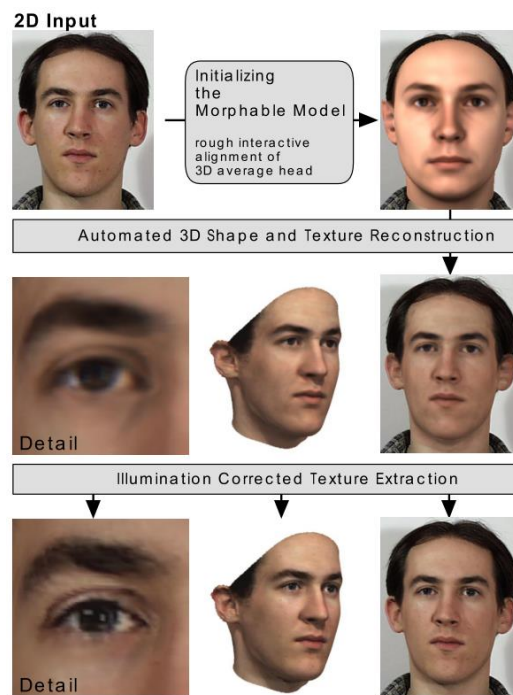


Figura 6: Figura ilustrativa do método utilizado no artigo acima descrito

### 2.3.2.2 *Stochastic Newton Optimization* (SNO) (Blanz and Vetter, 2003)

Este é um método utilizado para sobrepor uma imagem 2D no modelo 3DMM. Normalmente os métodos deste género utilizam funções de custo para fazer esse enquadramento, o SNO tenta minimizar essa função. O SNO evita os mínimos locais por pesquisar numa região maior reduzindo também assim o tempo de computação (Blanz and Vetter, 2003).

Este processo demora 4.5 minutos a executar num processador de 2.0GHz Pentium IV.

### 2.3.2.3 *Multi-Features Fitting* (MFF) (Romdhani and Vetter, 2005)

Este também é um método para enquadrar uma imagem 2D no modelo 3MM.

À semelhança do SNO, também tenta combater o problema do mínimo local; no entanto, utiliza uma função de custo para várias características, em vez de só uma. Essas características são a intensidade dos pixels, arestas da imagem, iluminação especular, características pré-gaussianas e limites de textura.

Este processo demora 70 segundos a executar num processador de 3.0GHz Pentium IV e produz um resultado superior ao método SNO.

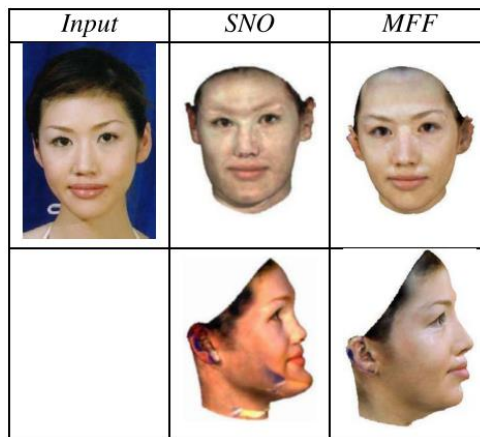


Figura 7: Comparação entre os métodos SNO e MFF

Como podemos ver na figura acima, o método MFF consegue obter um bom resultado, ao passo que o SNO apresenta alguns artefactos provenientes do fundo da imagem.

#### 2.3.2.4 *Basel Face Model (BFM)* (Paysan *et al.*, 2009)

Este é um modelo 3DMM que foi criado a partir de 200 faces (100 do sexo masculino e 100 do sexo feminino) digitalizadas com um *scanner* de alta precisão. Entre elas é estabelecida uma correspondência entre os mesmos pontos faciais, através de um processo chamado de registo (Amberg *et al.*, 2007). A textura da cara é obtida através de 3 fotografias tiradas pelo *scanner*. Por fim o 3DMM é definido por 2 vetores (um para a posição dos vértices e outro para a textura) de 3 dimensões.

Além disso, nesta publicação, são disponibilizados os scans usados, bem o como o modelo 3DMM criado.



Figura 8: Aplicação do modelo num rosto

A figura acima é resultante dos testes efetuados ao modelo numa base de imagens. À esquerda, a imagem 2D inicial, no centro a reconstrução do modelo sobre a imagem, e à direita o modelo 3D resultante.

### 2.3.2.5 *Local Features based Fitting* (Huber et al., 2015)

Este método começa por utilizar os ângulos e a posição espacial do rosto presentes na imagem 2D para projetar o 3DMM na imagem. Depois seleciona características locais (secções da imagem com alguma relevância) usando SIFT (*Scale Invariant Feature Transform*) (Lowe, 2004) para as extrair e ajustar o modelo. Este processo é repetido com novos parâmetros.

Este método apenas demora 200 milissegundos; não é referido qual é o processador utilizado para a medição, mas sendo o artigo de 2015, presume-se que seja um modelo recente.

### 2.3.2.6 *A Multiresolution 3D Morphable Face Model and Fitting Framework* (Huber et al., 2016)

Este artigo publica um 3DMM construído a partir de 169 scans faciais de alta resolução de pessoas de diferentes idades e raças. O modelo facial consiste em 3448 vértices (29587 no caso da obtenção de licença). É então disponibilizada uma biblioteca, denominada *eos*, que consiste em ferramentas que permitem a criação de um modelo facial a partir de uma imagem inicial e de um conjunto de pontos faciais.

## 2.4 Tecnologias

Aqui serão brevemente descritas duas tecnologias de extrema importância na área de modelação facial.

### 2.4.1 OpenCV

O OpenCV<sup>7</sup> (*Open Computer Vision*) é uma biblioteca livre para o desenvolvimento de aplicações na área de visão computacional. A biblioteca disponibiliza módulos de processamento de imagem e vídeo, estruturas de dados, álgebra linear e alguma interface gráfica. Além disso também permite a identificação de rostos e outros elementos através do uso de Haar Cascades<sup>8</sup>. A biblioteca está disponível para os sistemas operativos mais utilizados tais como Windows, Linux, Android e OS X.

### 2.4.2 DLib

A DLib<sup>9</sup> também é uma biblioteca ligada à visão computacional, que contém algoritmos de machine learning (aprendizagem automática por computador para o reconhecimento de padrões complexos e tomada decisões inteligentes baseadas em dados), processamento de imagem, interfaces gráficas e outras utilidades convenientes.

---

<sup>7</sup> <http://opencv.org/>

<sup>8</sup> [https://docs.opencv.org/master/d7/d8b/tutorial\\_py\\_face\\_detection.html](https://docs.opencv.org/master/d7/d8b/tutorial_py_face_detection.html)

<sup>9</sup> <http://dlib.net/>

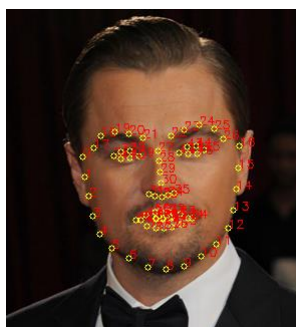


Figura 9: Exemplo de aplicação da DLib<sup>10</sup>

### 2.4.3 Visual Studio

O Visual Studio é um IDE (Integrated Development Environment) que permite criar aplicações em diversas linguagens e que neste projeto será usada para desenvolver em C++.

### 2.4.4 Qt

O Qt é uma ferramenta de *design* de interfaces gráficas. Também existe um *plug-in* do Qt que permite criar projetos Qt diretamente no Visual Studio, que será utilizado neste projeto.

## 2.5 Conclusão

Nesta secção foi contextualizado o problema e feita a análise de valor do produto. Foi também descrito o estado da arte na modelação facial categorizando as várias categorias de modelação facial existente, bem como alguns artigos e métodos mais relevantes. Concluiu-se que seguir a abordagem de criação do modelo facial através de um 3DMM seria o mais vantajoso.

---

<sup>10</sup> Imagem disponível em <https://i.stack.imgur.com/prClc.png>



## 3 Design da Solução

Neste capítulo irá ser descrita a arquitetura a utilizar no sistema e a modelação do projeto.

### 3.1 Arquitetura

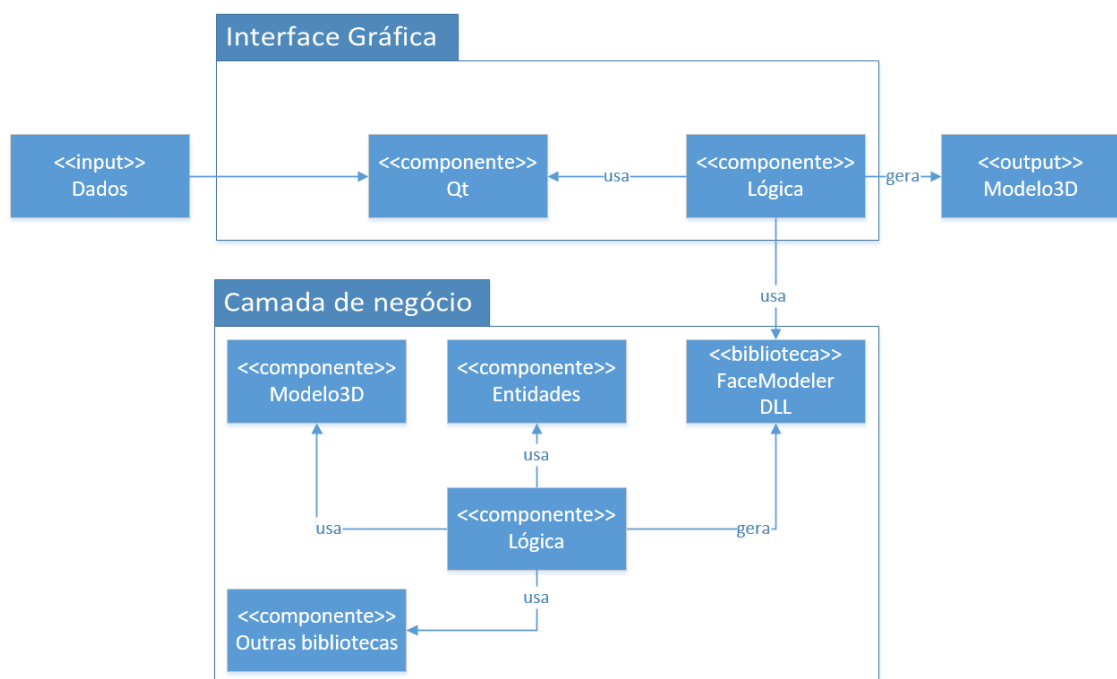


Figura 10: Arquitetura do sistema

A Figura 10 representa uma possível arquitetura do sistema. A camada de negócio utilizará o Modelo 3D e outras bibliotecas para gerar a biblioteca FaceModeler. Na interface gráfica será

utilizado a biblioteca gerada no modelo de negócio, a ferramenta de *design* gráfico Qt e o *input* de dados por parte do utilizador. Esse *input* pode ser uma imagem selecionada pelo utilizador ou uma fotografia tirada diretamente na aplicação. Em ambos os casos a imagem terá que apresentar um rosto virado para a frente. Por fim a aplicação pode exportar o modelo 3D gerado através da imagem inserida e com as alterações que o utilizador efetuar.

### 3.2 Modelação

O produto final da aplicação é uma biblioteca que permite utilizar algumas ferramentas tais como:

- Criação de um modelo facial quer a partir de uma fotografia, quer a partir da câmara;
- Alteração de emoções no modelo;
- Manipulação do modelo 3D (rotação nos vários ângulos e *zoom*);
- Alteração de parâmetros faciais (e.g. alteração da forma do nariz, boca).

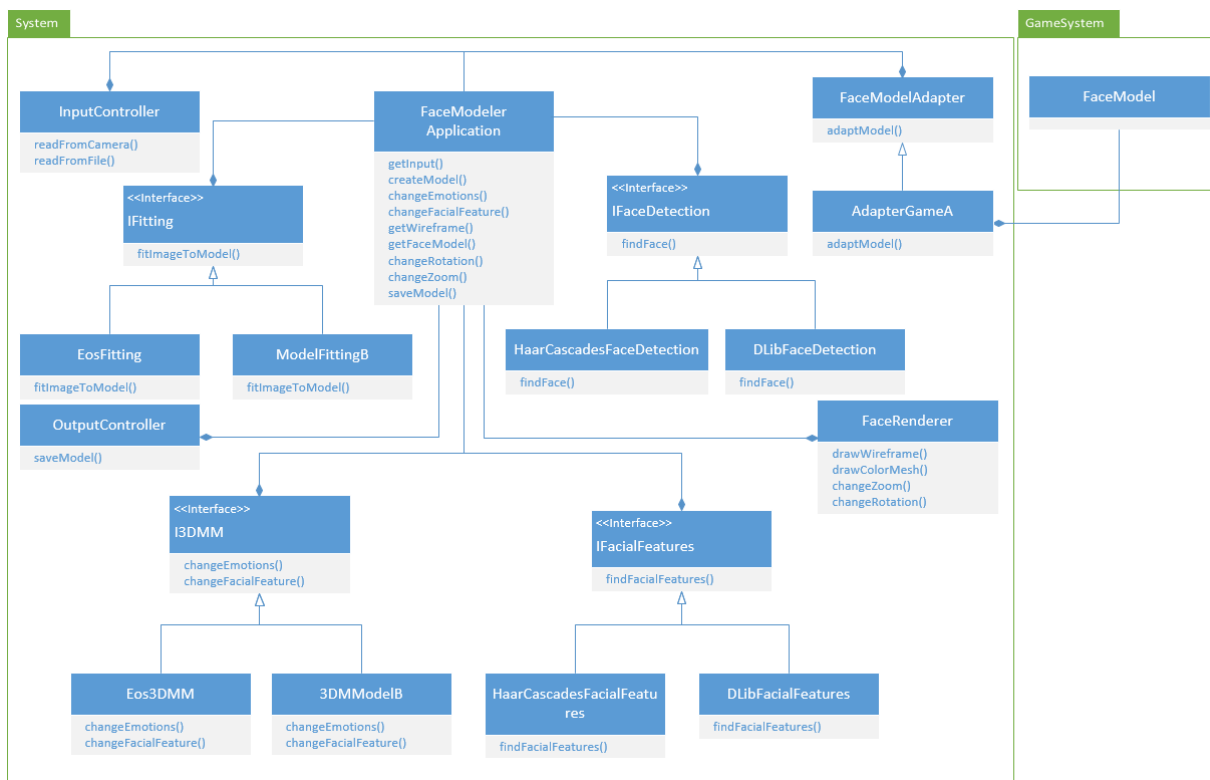


Figura 11: Diagrama de classes

Na Figura 11 encontra-se representado o diagrama de classes da biblioteca desenvolvida e como pode ser observado, todos os módulos estão ligados à *FaceModelerApplication*:

- **InputController**: É responsável pela leitura das imagens fornecidas ou obtidas diretamente da câmara;
- **OutputController**: Exporta o modelo facial para um ficheiro;
- **IFaceDetection**: Classe base responsável pela deteção facial. As classes derivadas desta utilizam Haar Cascades ou a biblioteca DLib para essa deteção;
- **IFacialFeatures**: Classe base responsável pela identificação de características faciais necessárias à reconstrução facial. As classes derivadas desta também utilizam Haar Cascades ou a DLib para proceder a essa identificação;
- **Eos3DMM**: Classe derivada de I3DMM que representa o modelo facial e é responsável pela mudança de emoções e ajuste de parâmetros faciais no modelo;
- **EosFitting**: Classe derivada de IFitting e é responsável pela reconstrução facial;
- **FaceRendering**: Classe responsável por realizar todas as operações de visualização do modelo depois de este estar criado, tais como o desenho, rotação e *zoom*;
- **FaceModelAdapter**: Classe base responsável por adaptar o modelo 3D gerado a modelos em outros formatos.

### 3.3 Interface Gráfica

Também vai ser criada uma aplicação gráfica em Qt que permite utilizar a biblioteca desenvolvida.

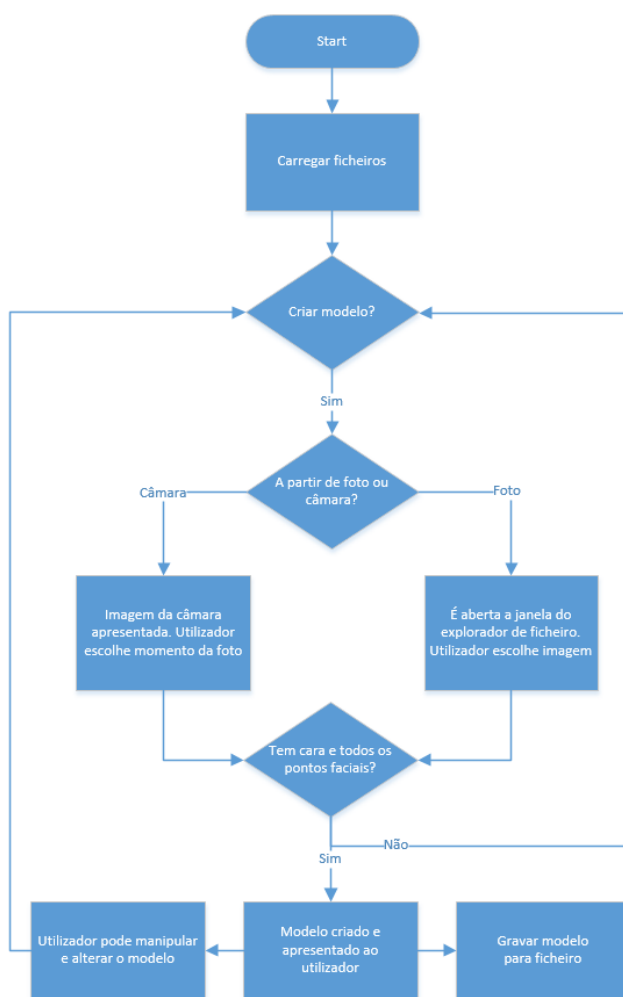


Figura 12: Diagrama de sequência da aplicação gráfica a desenvolver

A Figura 12 representa o diagrama de sequência da aplicação gráfica. No diagrama podemos observar que primeiramente são carregados os ficheiros necessários para a criação do modelo. O utilizador pode criar um modelo quer através de uma fotografia, quer através da câmara. Caso a imagem fornecida tenha um rosto e seja possível identificar todos os pontos faciais então o modelo pode ser criado. Por fim, o utilizador pode interagir com o modelo, fazer as alterações que desejar e gravar o modelo para um ficheiro, que poderá ser lido numa ferramenta de *design* gráfico.

## 4 Implementação da solução

Neste capítulo será refletida a fase de implementação e as alterações que foram necessárias fazer ao projeto.

### 4.1 Modelo 3DMM

O modelo 3DMM a ser usado é o elemento mais importante do projeto, pois é nele que se baseia todo o desenvolvimento posterior.

No ponto de situação anterior tinha sido referido que o modelo a utilizar seria o BFM, no entanto a linguagem em que este estava desenvolvido (MatLab) é diferente daquela em que se pretendia desenvolver o projeto (C++), gerando algumas dificuldades na sua integração no mesmo. Foi então necessário recorrer a outras soluções e alargar o estudo do estado da arte, tendo-se decidida a utilização da biblioteca maioritariamente desenvolvida por Patrik Huber: *eos*<sup>11</sup>.

Como já foi também referido neste documento esta biblioteca cria um modelo a partir de uma fotografia e da indicação de 68 pontos faciais. A DLib é a biblioteca mais indicada para identificar esses pontos.

#### 4.1.1 Identificação de Pontos Faciais

Para proceder à criação do modelo facial, é necessário numa primeira fase identificar um rosto na imagem submetida pelo utilizador; após a identificação facial, feita usando Haar Cascades, é fornecida a imagem inicial e o retângulo da face obtido na identificação à DLib. São obtidos

---

<sup>11</sup> <https://github.com/patrikhuber/eos>

então todos os pontos faciais, que são posteriormente guardados num vetor para serem utilizados na criação do modelo.

#### 4.1.2 Geração do Modelo

Aqui será descrito todo o tratamento efetuado com o *eos*.

Primeiramente é feito o carregamento do ficheiro "sfm\_shape\_3448.bin", que contém a forma como estão organizados os vértices que compõem o rosto. Caso fosse possível obter uma licença, seria utilizado outro ficheiro (um ficheiro com um modelo mais complexo) em vez deste, permitindo armazenar a informação relativa a 29587 vértices, em vez dos 3448 disponíveis.

Em seguida carrega-se o ficheiro "ibug\_to\_sfm.txt", encarregado de fazer a correspondência entre os pontos faciais e os vértices do modelo.

Depois, o ficheiro "expression\_blendshapes\_3448.bin", que contém as *blendshapes* para as expressões de raiva, nojo, medo, tristeza, alegria, tristeza e surpresa. As *blendshapes* são implementações de técnicas utilizadas para deformar uma malha e chegar com a mesma a outra malha diferente, sendo também possível obter várias formas intermédias entre as duas malhas. Assim sendo, a *blendshape* da alegria, por exemplo, poderá ter um valor entre 0 e 1, sendo que em 0 não haverá qualquer alteração à malha inicial, e em 1 a malha representará a alegria máxima que a *blendshape* pode produzir. Ao valor que esta *blendshape* tem chama-se o coeficiente.

O ficheiro "sfm\_3448\_edge\_topology.json" é utilizado para acelerar o processamento do encaixe dos contornos faciais onde parte da face está ocultada.

Por fim, os ficheiros "model\_contours.json" e "ibug\_to\_sfm.txt" são utilizados para ajudar no encaixe do contorno da face virada para a frente.

Para finalmente gerar o modelo, são enviados todos estes componentes juntamente com as coordenadas dos pontos faciais e a imagem para o método *fit*, que retorna a malha do modelo.

#### 4.1.3 Criação do Modelo de Cor

A versão do *eos* sem licença apenas permite a criação de um modelo de forma, não contendo informação de cor. O modelo de cor além da vantagem óbvia da cor, também não é influenciado pela luz. Ou seja, o modelo gerado seria semelhante caso sejam tiradas duas fotos à mesma pessoa em ambientes de iluminação diferente.

Apesar de não ser então possível obter a cor do modelo diretamente, existe uma forma indireta de o fazer, que irá ser descrita a seguir.

```
std::vector<glm::vec4> vertices; ///< 3D vertex positions.
std::vector<glm::vec2> texcoords; ///< Texture coordinates for each vertex.
std::vector<std::array<int, 3>> tvi; ///< Triangle vertex indices
```

Figura 13: Excerto de código referente às variáveis da classe Mesh

No excerto de código da Figura 13 estão os componentes da classe *Mesh* do *eos*, que representa o modelo 3D. Um vetor com as coordenadas 3D dos vértices, um vetor com as coordenadas da imagem a partir de onde foi extraído o modelo, e um vetor com conjuntos de 3 de índices (denominados de triângulos de agora em diante). A ligação em todos os triângulos dos seus vértices forma a malha (*mesh*).

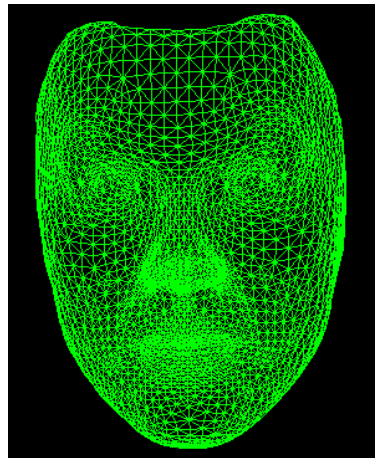


Figura 14: Desenho da malha do modelo 3D

A imagem da Figura 14 é obtida através da projeção (projeção essa fornecida pelo *eos*) dos pontos 3D num plano e traçando uma linha entre cada vértice do mesmo triângulo.

Para criarmos um modelo de cor, é necessário utilizar a imagem inicial e as coordenadas da textura, que correspondem a cada vértice do modelo. Como o número de vértices é relativamente reduzido, “pintar” cada vértice com a cor da coordenada correspondente não produziria uma imagem muito compreensível, pelo que se revela necessário preencher os espaços interiores dos triângulos. No entanto apenas temos a informação da cor nos pontos dos vértices, e como as projeções do modelo e da imagem inicial são diferentes, não é possível obter diretamente a informação requerida. É necessário, portanto calcular uma transformação para cada triângulo: usando as coordenadas na textura, e a projeção das coordenadas do modelo 3D, calcula-se a sua transformação e posteriormente a sua inversa. Com essa transformação, é possível fazer corresponder a todos os pontos dentro do triângulo projetado a coordenada correspondente na imagem inicial, preenchendo assim todos os triângulos da malha.

#### 4.1.4 Rotação e Zoom do Modelo 3D

As tarefas de rotação e *zoom in* e *zoom out* do modelo são simplificadas pela biblioteca *glm*<sup>12</sup> que permite fazer alterações à projeção inicial, tanto a nível de rotações em qualquer um dos três ângulos, como na escala (*zoom*).

```
if (rotationX != 0) {  
    float ang = to_rad(rotationX);  
    projectionRotated = glm::rotate(projectionRotated, ang, glm::vec3(1.0, 0, 0));  
}
```

Figura 15: Excerto de código para rodar cada vértice da malha

```
if (currentZoom != 1) {  
    glm::vec3 zoom(currentZoom, currentZoom, currentZoom);  
    projectionRotated = glm::scale(projectionRotated, zoom);  
}
```

Figura 16: Excerto de código para escalar cada ponto da malha

Como se pode observar nos excertos de código das figuras 15 e 16, para o caso da rotação, caso ela exista (*rotationX != 0*) converte-se o ângulo de graus para radianos e chama-se o método *glm::rotate* enviando a projeção anterior, o ângulo e um vetor com o valor 1 no ângulo que pretendemos rodar. No caso do *zoom*, caso seja necessário aproximar ou afastar o modelo (*currentZoom != 1*), cria-se um vetor com o valor do *zoom* em todos os eixos, para que a escala seja uniforme, e chama-se o método *glm::scale* com esse vetor e a projeção anterior.

Como o modelo apenas abrange o rosto e não toda a cabeça, não há necessidade de rotação total do modelo. Assim sendo, as rotações estão limitadas a um ângulo máximo de 30° para ambos os lados, nos 3 eixos. Também o *zoom* tem uma limitação na aproximação e distanciação permitida.

#### 4.1.5 Alteração de Expressões Faciais no Modelo 3D

Para criar um modelo com alteração de emoções é necessário criar uma nova malha. Para essa nova malha é necessário multiplicar os vetores das *blendshapes* com os atuais coeficientes das *blendshapes*. A essa multiplicação de vetores soma-se o vetor da forma do modelo. Com esta forma obtida consegue-se desenhar uma nova malha, com as emoções que estavam indicadas nos coeficientes.

```
Eigen::Map<const Eigen::VectorXf> current_blendshapes(blendshape_coeffs.data(), blendshape_coeffs.size());  
Eigen::VectorXf current_combined_shape = blendshapes_as_basis * current_blendshapes + current_pca_shape;  
  
return morphablemodel::sample_to_mesh(current_combined_shape, _3dmm->morphable_model.get_color_model().get_mean(),  
    _3dmm->morphable_model.get_shape_model().get_triangle_list(), _3dmm->morphable_model.get_color_model().get_triangle_list(),  
    _3dmm->morphable_model.get_texture_coordinates());
```

Figura 17: Excerto de código para criar o novo modelo com a nova expressão facial.

<sup>12</sup> <https://glm.g-truc.net/0.9.8/index.htm>

No excerto de código da Figura 17 podemos observar a multiplicação dos vetores das *blendshapes* (*blendshapes\_as\_basis*) com os dos coeficientes das *blendshapes* atuais (*current\_blendshapes*), seguida da sua soma com a dos vetores da forma do modelo (*current\_pca\_shape*).

#### 4.1.6 Alteração de Parâmetros Faciais

Para a alteração dos parâmetros faciais tais como o tamanho do nariz ou da boca, foi seguido um artigo (Yee, 2017) cujo trabalho envolveu a criação de novas *blendshapes*. No artigo é descrito que a *blendshape* do riso foi criada a partir da região da boca da *blendshape* da alegria. Todos os vértices não pertencentes à região da boca seriam colocados a zero, ficando assim retidas apenas as alterações que a *blendshape* fazia na região da boca.

No entanto não se sabe quais os vértices que correspondem a quais regiões do modelo. Pode-se descobrir esses vértices através da projeção do modelo 3D numa imagem e nessa imagem “desenhar” um polígono à volta da região pretendida. Depois basta verificar quais os vértices que pertencem ao polígono e, portanto, quais os vértices dessa região.

```
for (int i =0;i<mesh.vertices.size();i++)
{
    glm::vec3 vertex = { mesh.vertices[i][0], mesh.vertices[i][1], mesh.vertices[i][2] };
    const auto p1 = glm::project(vertex, modelview, projection, viewport);
    cv::Point2f p(p1.x, p1.y);
    if (pointPolygonTest(polygon, p, true) > 0)
        vertexIndexes.push_back(i);
}
```

Figura 18: Excerto de código para encontrar os índices de todos os vértices pertencentes a uma determinada região

No excerto de código da Figura 18 são percorridos todos os vértices da malha, projetando cada vértice num plano e testando se este pertence ao polígono, caso pertença, é colocado num vetor o índice do vértice. A variável “polygon” utilizada foi criada anteriormente através de pontos predefinidos que correspondem à região à volta da boca neste caso.

Este processo é realizado para todas as zonas de interesse tais como os olhos, nariz e boca.

As *blendshapes* têm 3 vezes mais índices que a malha, 3 para cada vértice, indicando a deformação que o vértice sofre em cada eixo. A ordenação dos vértices na *blendshape* é igual à da malha, mas com 3 valores seguidos para o mesmo vértice. Assim sendo, ao índice 0 da malha pertencem os índices 0, 1 e 2 da *blendshapes* e por aí em diante.

O artigo no entanto não refere quais foram as alterações feitas à *blendshape*, deixando assim uma grande questão acerca de como proceder, pelo que não foi mesmo possível concluir a tarefa.

## 4.2 Geração da biblioteca

Estando o projeto em C++ preparado para ser utilizado por outras aplicações, pode ser criada uma biblioteca do mesmo.

Primeiramente define-se quais as classes e métodos que poderão ser utilizados ou chamados. Para isso indica-se com a expressão “`__declspec(dllexport)`”.

Por fim basta indicar ao Visual Studio para gerar uma DLL (Dynamic-Link Library).

## 4.3 Interface Gráfica

Um dos objetivos do projeto era desenvolver uma interface gráfica que pudesse demonstrar todas as funcionalidades da biblioteca desenvolvida. Para além disso, a interface também serve de ferramenta para o programador e utilizadores testarem funcionalidades e avaliarem as mesmas.

### 4.3.1 Configurações Iniciais

A ferramenta de *design* utilizada para desenvolver a interface gráfica foi o Qt e um *plug-in* do mesmo disponível para o Visual Studio. Assim sendo foi criado um projeto Qt, incluindo então a biblioteca gerada anteriormente.

Para ser manipulado o modelo 3D foi necessário criar uma classe derivada de uma das classes fornecidas pelo Qt. As classes base do Qt não detetam movimentos do rato; assim sendo foi criada uma classe derivada de QLabel que deteta quando o rato está sobre uma das suas instâncias. Também deteta quando os botões são clicados e qual a distância que se mexe e em que direção e os movimentos da roda do rato.

### 4.3.2 Interface

A interface escolhida para este projeto é constituída por uma *label* do lado esquerdo, onde estará presente o modelo quando este for criado e do lado direito um painel com vários *sliders* onde é possível alterar as várias emoções do modelo.

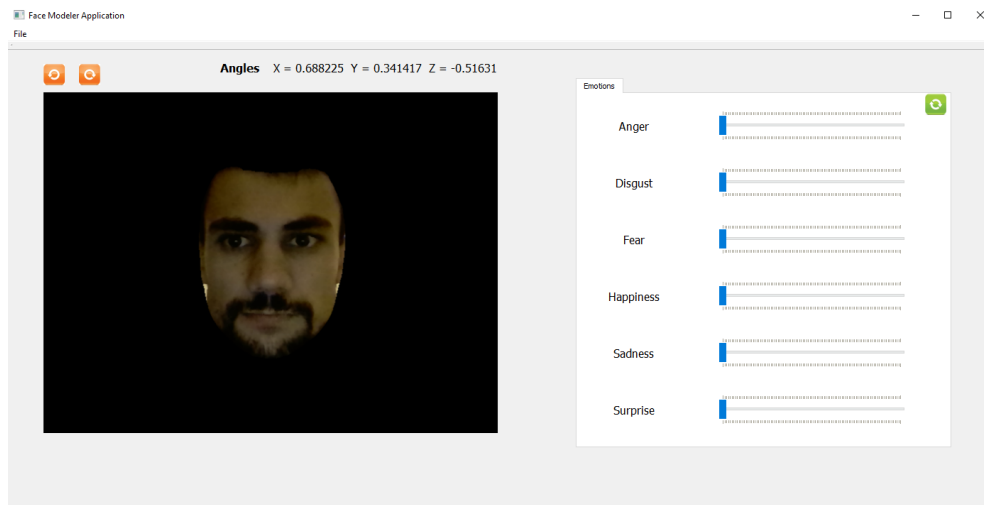


Figura 19: Exemplo execução da aplicação

Tal como dito anteriormente a *label* do modelo deteta os cliques e movimentos do rato. Para rodar o modelo basta clicar e arrastar na direção pretendida, quando o botão for largado, será apresentado o modelo com a nova orientação. Para a rotação no terceiro ângulo, estão presentes dois botões acima do modelo que rodam 5° nesse ângulo. A Figura 19 demonstra um exemplo de como fica a interface depois de criado o modelo 3D.

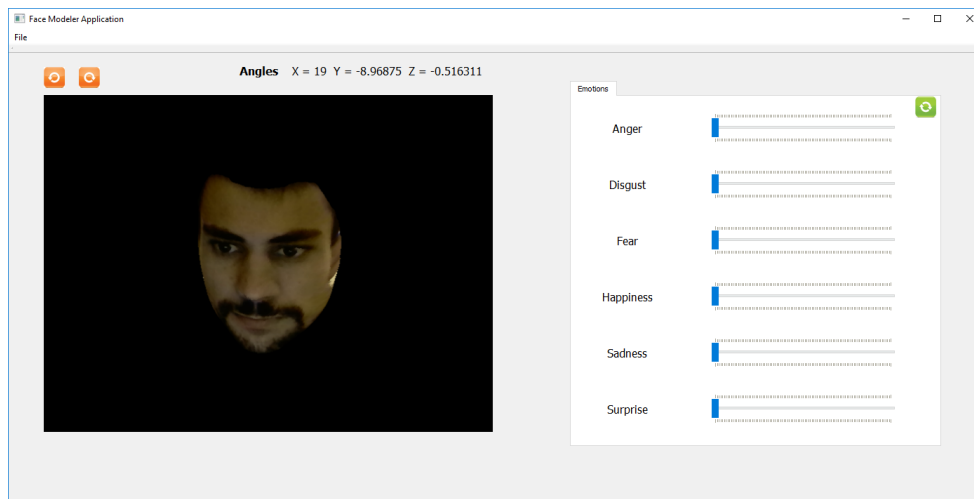


Figura 20: Exemplo de rotação do modelo

Na imagem da Figura 20 está demonstrado um exemplo da rotação do modelo, neste caso de 19° no eixo do X e -9° no do Y.

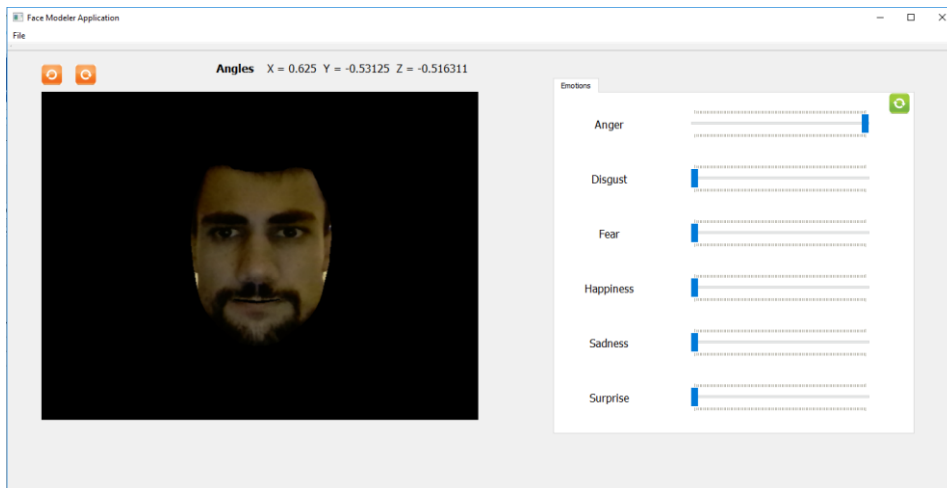


Figura 21: Expressão de raiva

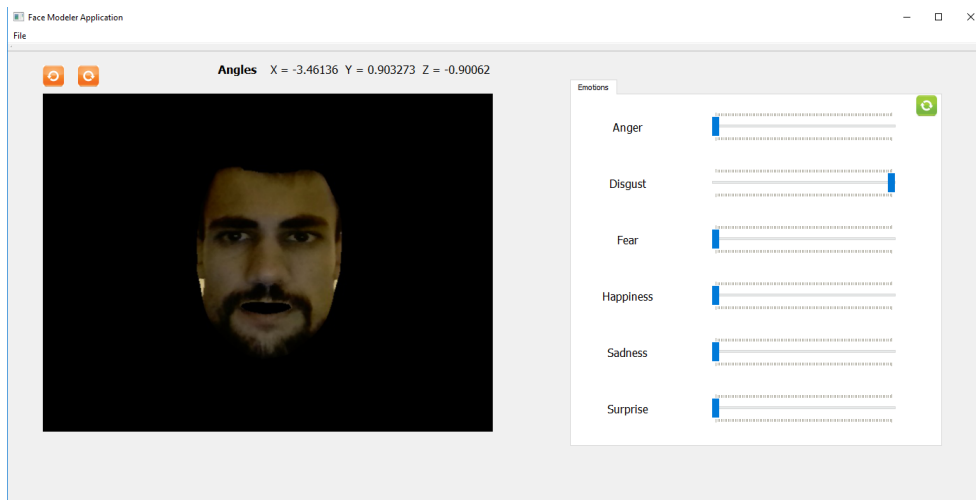


Figura 22: Expressão de nojo

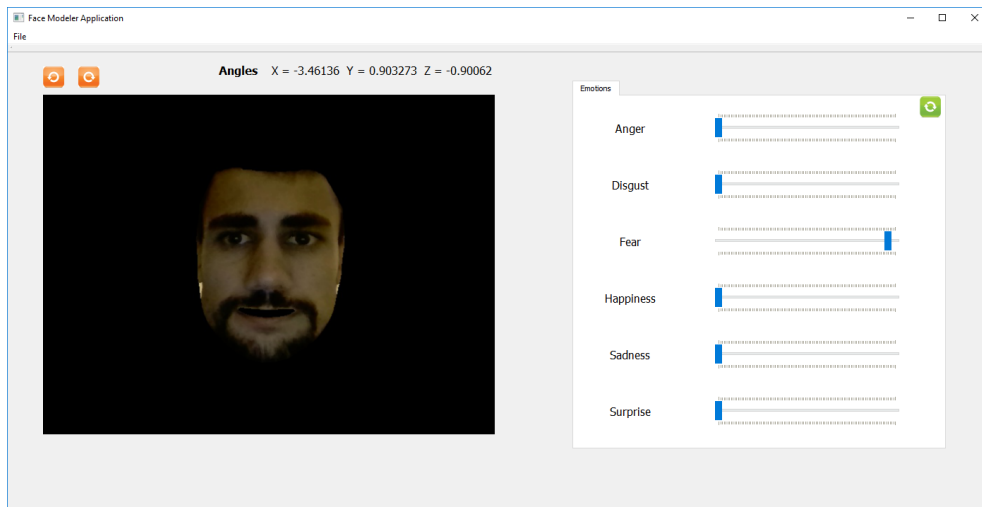


Figura 23: Expressão de medo

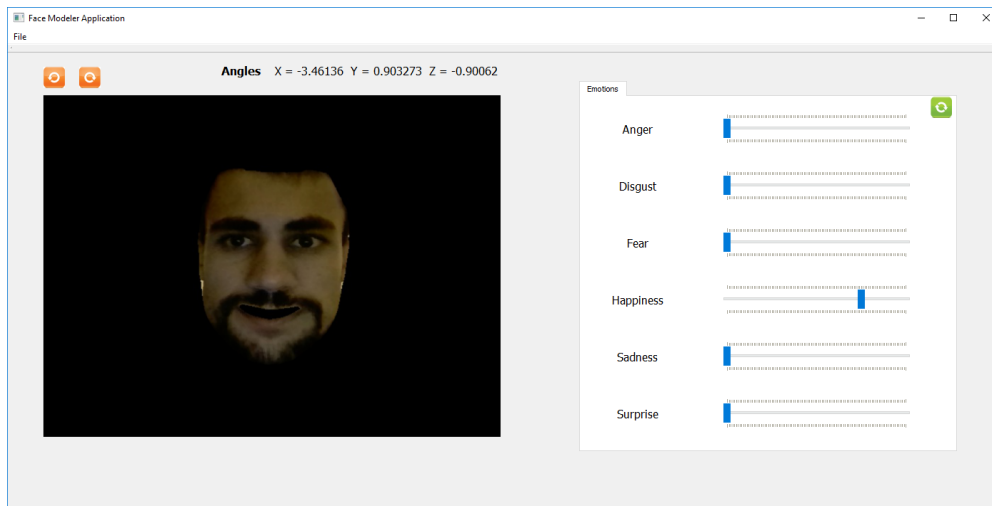


Figura 24: Expressão de felicidade

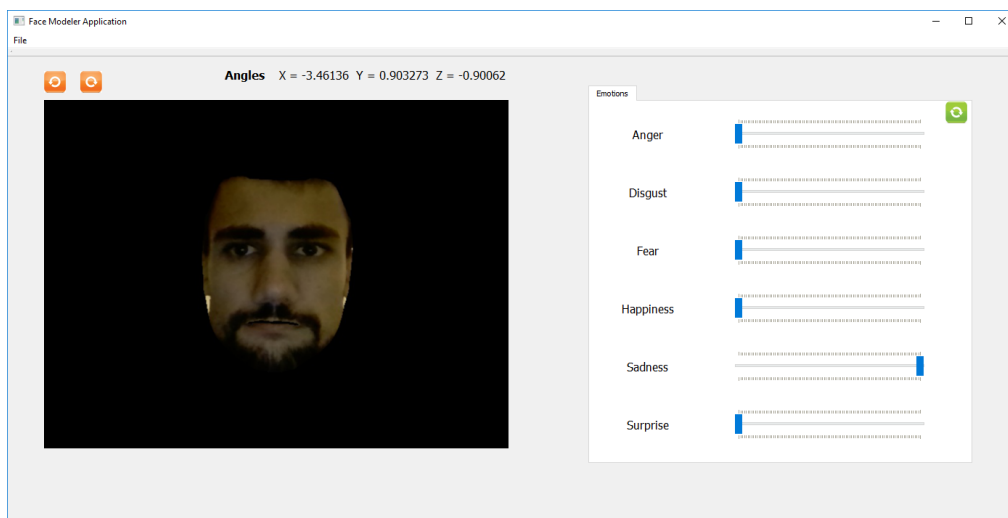


Figura 25: Expressão de tristeza

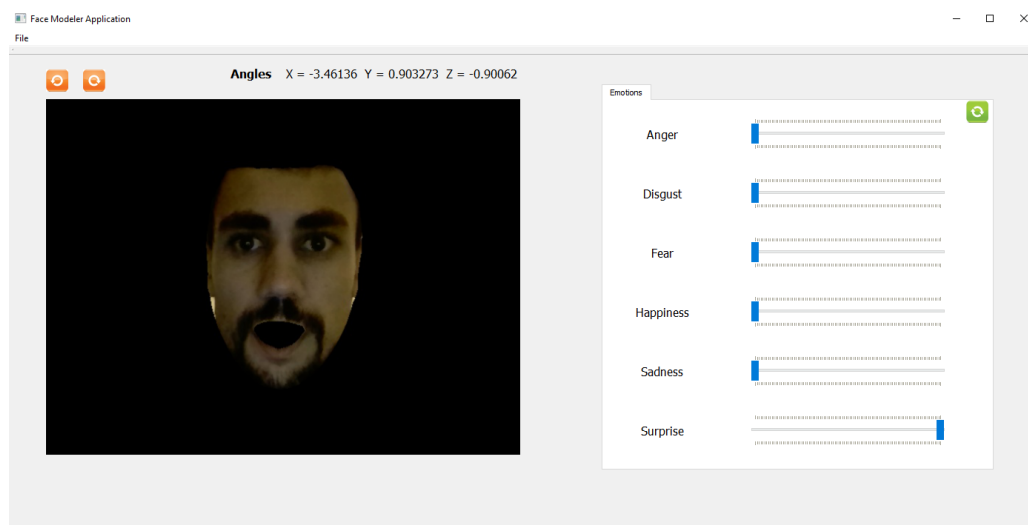


Figura 26: Expressão de surpresa

Desde a figura 21 até à 26 estão ilustradas as diferentes expressões disponíveis na aplicação.

### 4.3.3 Criação do Modelo

Há duas formas para ser criado o modelo, através da câmara ou de uma imagem. Qualquer uma das formas pode ser escolhida na barra de menus no topo da aplicação. A opção de criação através de uma câmara abre uma outra janela com a imagem obtida na câmara e com um botão que permite criar o modelo. A segunda opção abre um explorador de ficheiros que permite escolher uma imagem para importar.

Todos os tipos de exceções são tratadas e caso tenha sido criado o modelo com sucesso, é apresentada uma mensagem ao utilizador e retorna-se à janela anterior.

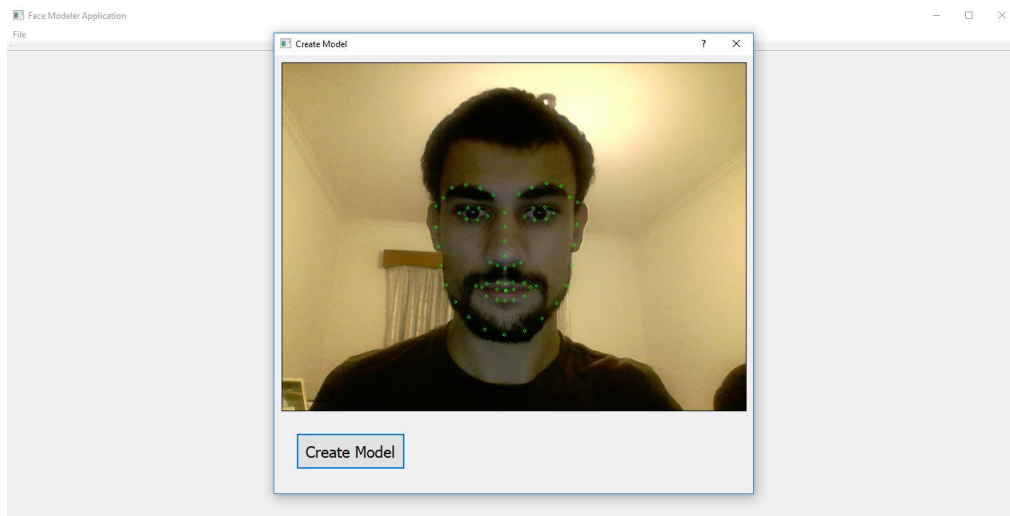


Figura 27: Criação do modelo 3D

Também é possível gravar o modelo para um ficheiro, que pode ser visualizado posteriormente numa ferramenta de *design* interativa (e.g. Blender) (Figura 28).

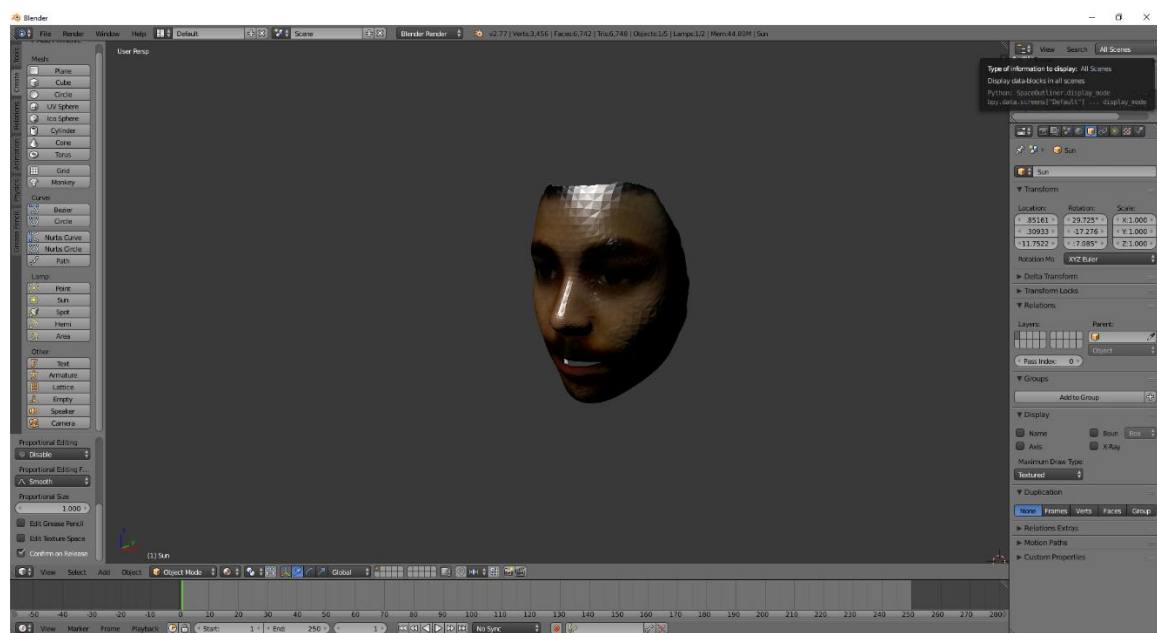


Figura 28: Exemplo de exportação do modelo 3D para o Blender



# 5 Avaliação

Neste capítulo será descrito como foi efetuada a avaliação do projeto, quer a nível de tempo de execução, quer em termos de satisfação do utilizador.

## 5.1 Avaliação do Tempo de Execução do Algoritmo

Foi então efetuado um teste ao tempo de execução do algoritmo da criação do modelo facial. Este teste foi efetuado com várias imagens de rostos de diversas dimensões, sendo cada uma delas testada três vezes de modo a obter um resultado mais correto.

Tabela 1: Tempos de execução do algoritmo de criação do modelo facial

Tamanho do rosto	Tempo 1 (ms)	Tempo 2 (ms)	Tempo 3 (ms)	Tempo médio (ms)
105*105	902	910	906	906
151*151	1029	1022	1046	1032
216*216	1066	1024	1023	1038
311*311	1287	1297	1269	1284
373*373	1349	1319	1350	1339
447*447	1235	1152	1192	1193
925*925	1776	1867	1700	1781

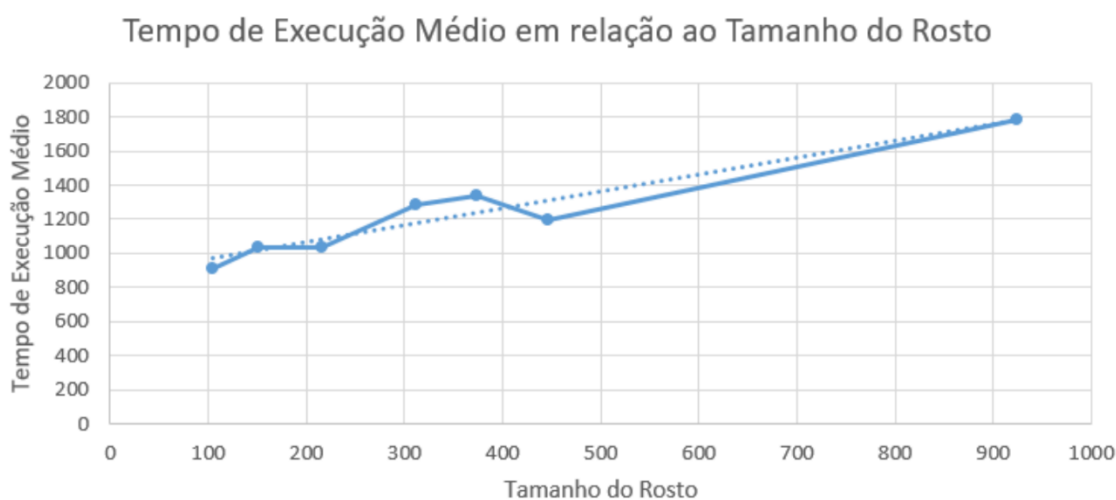


Figura 29: Tempo de execução médio em relação ao tamanho do rosto

Os testes foram executados num processador Intel Core i7-6700HQ @ 2.60Hz. Durante os testes os tempos foram variando, e na maioria dos casos eram mais altos quanto maior fosse a dimensão do rosto. No entanto, como podemos observar traçando um gráfico da relação entre o tempo médio de execução e o tamanho do rosto representado na Figura 29, o aumento não é diretamente proporcional. Mesmo com imagens grandes, o tempo de execução manteve-se abaixo dos dois segundos, o que está abaixo do limite máximo de 3 segundos que se tinha definido com aceitável para uma aplicação deste género.

## 5.2 Avaliação do Protótipo por Utilizadores

De modo a testar a aplicação, foi proposto a vários indivíduos que testassem a aplicação e que seguidamente respondessem a um pequeno inquérito sobre a mesma. Os testes foram efetuados em diversas ocasiões, por familiares e amigos, num total de 22 participantes.

A cada um dos indivíduos era fornecida a contextualização do projeto e explicada a necessidade de testar o mesmo. Cada indivíduo procedeu então à criação do modelo através da imagem da câmara, podendo repetir o processo caso considerasse que a foto não tivesse ficado bem. De seguida era permitido que fizesse as alterações e manipulações que desejasse e quando terminava era-lhe então pedido que respondesse ao inquérito com a máxima sinceridade e informado que o mesmo era anónimo.

O inquérito foi efetuado com o *Google Forms*, que a qualquer momento pode produzir gráficos que demonstram a distribuição das respostas.

As questões foram respondidas numa escala de 1 a 5, com adaptações do significado da escala para que a resposta se adequasse à pergunta (esses diferentes significados nas escalas não aparecem nos gráficos finais).

Em seguida é analisada cada uma das questões.

### Qual a sua idade?

22 responses

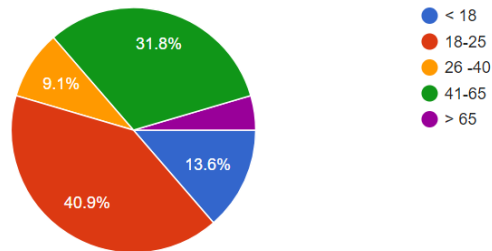


Figura 30: Distribuição de idades dos inquiridos

A Figura 30 ilustra o gráfico da faixa etária dos inquiridos. Como podemos observar, a maioria tinha entre 41 e 65 e entre 18 e 25 anos de idade.

### Quão semelhante achou o modelo, relativamente à foto utilizada?

22 responses

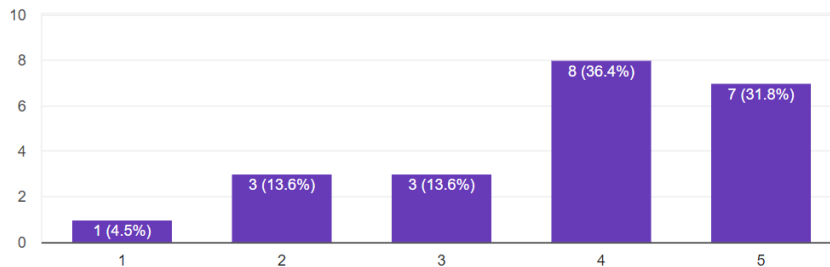


Figura 31: Avaliação da semelhança do modelo em relação à foto

A pergunta colocada relativamente ao protótipo questiona a similaridade entre a foto utilizada e o modelo gerado (Figura 31). Neste caso o valor 1 representa nenhuma semelhança, e o valor 5 muita semelhança. A maioria das respostas foram entre 4 e 5 perfazendo uma média de  $(1 + 2*3 + 3*3 + 4*8 + 5*7) / 22 = 3,78$ .

### Quão rápida achou a criação do modelo 3D?

22 responses

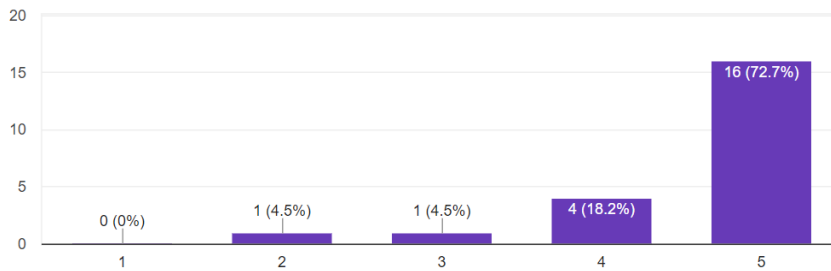


Figura 32: Avaliação da rapidez de criação do modelo

A pergunta cujas respostas estão demonstradas na Figura 32 refere-se ao tempo desde que a foto é tirada até que o modelo seja apresentado. Nesta questão o valor 1 representa muito lenta, e o 5 muito rápida. As repostas dadas foram maioritariamente o valor máximo, obtendo um valor médio de  $(2 + 3 + 4*4 + 5*16) / 22 = 4,60$ .

### O que achou da qualidade do resultado final?

22 responses

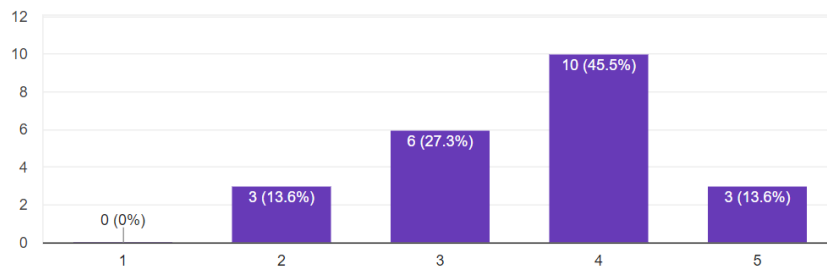


Figura 33: Avaliação da qualidade final do modelo

A questão da Figura 33 questiona a qualidade do modelo apresentado. Neste caso o valor 1 representa muito baixo, e o valor 5 muito alto. Nesta questão os inquiridos debateram-se entre o valor 3 e 4, originando um valor médio de  $(2*3 + 3*6 + 4*10 + 5*3) / 22 = 3,60$ .

O que achou relativamente à qualidade de manipulação do modelo?

22 responses

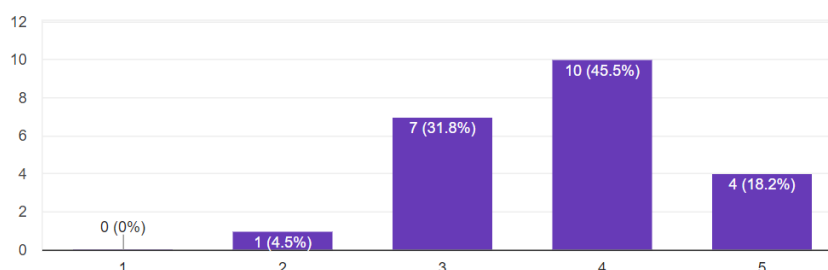


Figura 34: Avaliação da qualidade de manipulação do modelo

Na questão representada na Figura 34 era pedido aos inquiridos que avaliassem a qualidade de manipulação do modelo. Aqui o valor 1 representa muito baixo, e o valor 5 muito alto. Mais uma vez os inquiridos debateram-se entre os valores 3 e 4, dando origem a uma média de  $(2 + 3*7 + 4*10 + 5*4) / 22 = 3,78$ .

O que achou relativamente à quantidade de ferramentas de manipulação do modelo?

22 responses

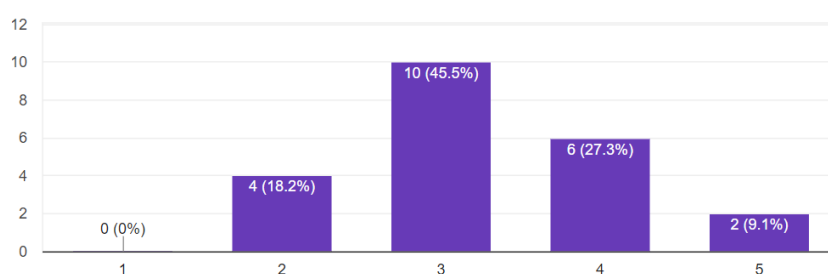


Figura 35: Avaliação da quantidade de ferramentas de manipulação

Esta questão representada na Figura 35 era relativa à quantidade de ferramentas de manipulação do modelo disponíveis. Nesta questão o valor 1 significava pouca quantidade, e o valor 5 muita quantidade. Esta foi a questão que obteve piores pontuações obtendo uma média de  $(2*4 + 3*10 + 4*6 + 5*2) / 22 = 3,27$ .

No seu entender, quão útil é o produto final?

22 responses

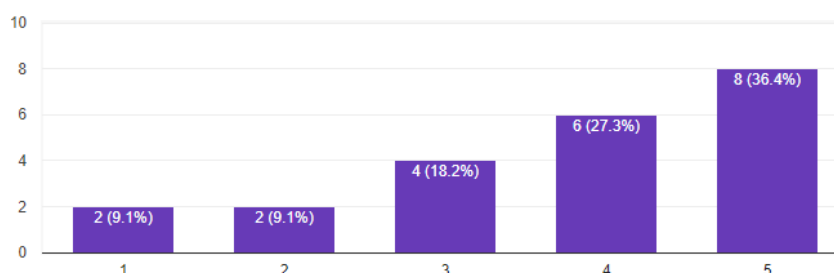


Figura 36: Avaliação da utilidade do produto final

A última pergunta questionava os inquiridos quanto à utilidade do produto desenvolvido. O valor de 1 significava pouco útil, e o de 5 muito útil. A média das respostas é de  $(1*2 + 2*2 + 3*4 + 4*6 + 5*8) / 22 = 3,72$ .

Em geral os resultados dos inquéritos são satisfatórios e atingem as expectativas. A mais provável razão para alguma disparidade em algumas das questões deve-se ao facto da falta de compreensão de e conhecimentos na área e talvez também à idade dos inquiridos. Isso tanto pode ser benéfico como desfavorável.

A questão com maior pontuação foi a da rapidez da criação do modelo que como já foi discutido, foi de sempre abaixo de 2 segundos e em média pouco mais que um segundo. Em adição a isso apenas foi efetuado o desenho do modelo que tem uma duração média entre 80 e 100ms, que se torna pouco significativo.

A questão com pontuação mais baixa foi referente à quantidade de ferramentas de manipulação do modelo disponibilizadas. Tal já seria de esperar dada a não conclusão de algumas das ferramentas previstas, tal como o ajuste de parâmetros faciais tais como a bocas os olhos e o nariz.

## 5.3 Conclusão

Inicialmente tinha sido proposto a integração do sistema a criar num sistema real, tal como uma empresa de videojogos que utilize personagens 3D. Infelizmente tal não foi possível devido à não conclusão do ajuste de parâmetros faciais que tinha sido inicialmente proposto.

Apesar de tudo, a avaliação do produto por parte dos utilizadores, foi de encontro às expectativas. A maioria dos utilizadores valorizou a rapidez da geração do modelo, com uma qualidade final aceitável. Como principal parâmetro a melhorar em projetos futuros, surge a falta de ferramentas de edição do modelo após a sua geração. No geral, os utilizadores acharam interessante e inovador o produto apresentado e mostraram curiosidade acerca do seu desenvolvimento e aplicabilidade no mercado.

## 6 Conclusões Finais

Neste capítulo será refletido o desenvolvimento do projeto no geral e abordados possíveis melhoramentos para o projeto.

### 6.1 Balanço do Trabalho Desenvolvido

Os objetivos traçados para esta dissertação eram o desenvolvimento de um sistema capaz de:

- Criar um modelo facial 3D a partir de imagens;
- Fazer ajustes aos diversos parâmetros faciais;
- Gerar uma *framework* que permita executar essas funções;
- Criar uma interface gráfica que demonstre as funcionalidades do sistema criado.

Assim sendo, pode-se concluir que os objetivos foram maioritariamente atingidos. Foi desenvolvido um sistema capaz de criar um modelo facial a partir de apenas uma fotografia frontal, no tempo máximo de 2 segundos, mesmo para imagens de grandes dimensões. O sistema também é capaz de alterar 6 emoções (raiva, medo, nojo, felicidade, tristeza, surpresa) ao modelo gerado e exportar o modelo 3D para um ficheiro que pode ser lido numa ferramenta de *design* interativo como o Blender. A partir desse sistema foi criada uma biblioteca capaz de executar todas as funções de criação e alteração do modelo e da sua visualização. Por fim, foi criada uma interface gráfica que utiliza a biblioteca gerada e permite demonstrar todas as suas funções em ação.

Apesar de um dos objetivos propostos não ter sido atingido na sua plenitude (ajuste de parâmetros faciais), foi adicionada outra forma indireta de o fazer: alteração de emoções.

Em relação à utilização por parte dos utilizadores da ferramenta desenvolvida, a parte mais apreciada foi a rapidez da criação do modelo, e a menos apreciada foi a quantidade de ferramentas disponível (rotação, *zoom* e alteração de emoções).

## **6.2 Trabalho Futuro**

O trabalho realizado até à data é um bom ponto de partida, no entanto existem outros aspetos que contribuiriam consideravelmente para uma melhor versão do protótipo apresentado.

### **6.2.1 Manipulação do Modelo 3D**

Para o ajuste de parâmetros faciais, que não foi bem-sucedido, podem ser criadas mais *blendshapes* com essa função. Também seria interessante a alteração de outros aspetos do modelo tais como a cor dos olhos, da pele, das sobrancelhas ou, se fosse o caso, a barba.

### **6.2.2 Melhoria Modelo 3D**

O modelo utilizado apenas possui 3448 vértices, e por isso, relativamente baixa definição. Com a obtenção de uma licença e conseqüentemente de um modelo com mais de 29 mil vértices seria possível aumentar muito a qualidade do modelo final, embora reduzindo naturalmente o tempo de execução.

### **6.2.3 Acabamento do Modelo 3D**

O modelo final apresentado não representa na totalidade o indivíduo alvo. Para completar o modelo poderia ser adicionada a modelação do interior da boca e o resto da cabeça em 360°. Outra ideia um pouco mais desafiante seria a adição de cabelo ao modelo, permitindo escolher o tipo e a cor de cabelo ou até mesmo que este fosse deduzido pelo meio de fotografia.

### **6.2.4 Otimização**

Embora a execução da criação do modelo e outras tarefas sejam relativamente rápidas, ainda há lugar para a melhoria nos tempos de execução. A paralelização de tarefas certamente que seria uma alternativa a considerar.

## 7 Bibliografia

Alomari, A. A. *et al.* (2011) 'A state of art on 3D face modeling approaches', *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, (July), pp. 1–5. doi: 10.1109/ICEEI.2011.6021791.

Amberg, B. *et al.* (2007) 'Optimal Step Nonrigid ICP Algorithms for Surface Registration', *CVPR '07*. Available at: [http://gravis.dmi.unibas.ch/publications/2007/CVPR07\\_Amberg.pdf](http://gravis.dmi.unibas.ch/publications/2007/CVPR07_Amberg.pdf).

Blanz, V. and Vetter, T. (1999) 'A Morphable Model For The Synthesis Of 3D Faces'. Available at: <http://gravis.dmi.unibas.ch/publications/Sigg99/morphmod2.pdf>.

Blanz, V. and Vetter, T. (2003) 'Face recognition based on fitting a 3D morphable model', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9), pp. 1063–1074. doi: 10.1109/TPAMI.2003.1227983.

Huber, P. *et al.* (2015) 'Fitting 3D Morphable Models using Local Features'.

Huber, P. *et al.* (2016) 'A Multiresolution 3D Morphable Face Model and Fitting Framework'. Available at: [http://www.patrikhuber.ch/files/3DMM\\_Framework\\_VISAPP\\_2016.pdf](http://www.patrikhuber.ch/files/3DMM_Framework_VISAPP_2016.pdf).

Jones, M. J. and Poggio, T. (1998) 'Multidimensional morphable models: A framework for representing and matching object classes', *International Journal of Computer Vision (IJCV)*, 29(2), pp. 107–131. doi: 10.1023/A:1008074226832.

Koen, P. a (2004) 'Understanding the Front End: A Common Language and Structured Picture', *Stevens Institute of Technology*.

Koen, P. A. *et al.* (2002) 'Fuzzy Front End: Effective Methods, Tools, and Techniques', *Industrial Research*, (The PDMA).

Lowe, D. G. (2004) 'Distinctive Image Features from Scale-Invariant Keypoints', *International Journal of Computer Vision*.

Park, I. K., Zhang, H. and Vezhnevets, V. (2005) 'Image-Based 3D Face Modeling System', *EURASIP Journal on Applied Signal Processing*, 13, pp. 2072–2090. Available at:

[http://download.springer.com/static/pdf/51/art%253A10.1155%252FASP.2005.2072.pdf?originUrl=http%3A%2F%2Fasp.eurasipjournals.springeropen.com%2Farticle%2F10.1155%2FASP.2005.2072&token2=exp=1488587692~acl=%2Fstatic%2Fpdf%2F51%2Fart%25253A10.1155%252FASP.2005.2072.pdf\\*~hmac=0dba67f7c160385887ed08d56c42347c3b87dd2cd3b82d24953570ffef0072ba](http://download.springer.com/static/pdf/51/art%253A10.1155%252FASP.2005.2072.pdf?originUrl=http%3A%2F%2Fasp.eurasipjournals.springeropen.com%2Farticle%2F10.1155%2FASP.2005.2072&token2=exp=1488587692~acl=%2Fstatic%2Fpdf%2F51%2Fart%25253A10.1155%252FASP.2005.2072.pdf*~hmac=0dba67f7c160385887ed08d56c42347c3b87dd2cd3b82d24953570ffef0072ba).

Patel, A. and Smith, W. A. P. (2009) '3D Morphable Face Models Revisited'. Available at: <http://vision.lbl.gov/Conferences/cvpr/Papers/data/papers/0921.pdf>.

Paysan, P. *et al.* (2009) 'A 3D face model for pose and illumination invariant face recognition', *6th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009*, pp. 296–301. doi: 10.1109/AVSS.2009.58.

Pighin, F. *et al.* (1998) 'Synthesizing Realistic Facial Expressions from Photographs'. Available at: <http://kucg.korea.ac.kr/Seminar/2003/src/PA-03-07.pdf>.

Romdhani, S. and Vetter, T. (2005) 'Estimating 3D shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2*, pp. 986–993. doi: 10.1109/CVPR.2005.145.

Vetter, T., Jones, M. J. and Poggio, T. (1997) 'A Bootstrapping Algorithm for Learning Linear Models of Object Classes'.

Yee, L. W. (2017) 'Facial Landmark Tracking Final Report'. Available at: [http://i.cs.hku.hk/fyp/2016/report/final\\_report/LI](http://i.cs.hku.hk/fyp/2016/report/final_report/LI)  
Wing  
Yee\_11424152\_assignsubmission\_file\_final\_report.pdf.