

PE2LGP

PE2LGP: Do Texto à Língua Gestual (e vice-versa)

Ruben Emanuel Ramires dos Santos

Dissertação para obtenção do Grau em Mestre em Engenharia
Informática e de Computadores

Mestrado Bolonha em Engenharia Informática e de Computadores

Orientadores: Prof^a. Maria Luísa Torres Ribeiro Marques da Silva Coheur
Prof. João António Madeiras Pereira

Júri

Presidente: Prof. Miguel Nuno Dias Alves Pupo Correia
Orientador: Prof^a. Maria Luísa Torres Ribeiro Marques da Silva Coheur
Vogal: Prof. Alfredo Manuel dos Santos Ferreira Júnior

Novembro 2016

Conteúdo

1	Introdução	16
1.1	Problema e Motivação	17
1.2	Objetivos	17
1.3	Contribuições	18
1.4	Organização do Documento	18
2	PE2LGP - From European Portuguese to Portuguese Sign Language	19
2.1	Visão Geral	20
2.1.1	Gloss Translate	21
2.1.2	Lookup	21
2.1.3	Animate	21
2.2	Implementação	21
2.2.1	Blender	21
2.2.2	Filosofia dos Addons	21
2.2.3	Processamento e Tradução da Língua Natural	22
2.2.4	Lookup	23
2.2.5	Personagem 3D	24
2.2.6	Animações	24
2.3	Avaliação	24
3	Trabalho Relacionado	25
3.1	Uso de Avatares para Língua Gestual	26
3.1.1	Avatares na Língua Gestual Portuguesa	26
3.1.2	Avatares na Língua Gestual Brasileira	28
3.1.3	Avatares noutras Línguas Gestuais	30
3.1.4	Relação com o PE2LGP	31
3.2	Desafios na Criação de Avatares para Língua Gestual	31
3.3	Criação e Manipulação de Objetos Usando o Kinect	32
3.3.1	Softwares	33
3.3.2	Relação com o PE2LGP	33
3.4	Rastreamento das Mãos (Hand Tracking)	33
3.4.1	Métodos Discriminativos	34

3.4.2	Métodos Generativos	34
3.4.3	Métodos Híbridos	35
3.4.4	Leap Motion	37
3.4.5	Relação ao PE2LGP	37
3.5	Recursos da Área de Processamento de Língua Natural	38
3.5.1	Natural Language Toolkit	38
3.5.2	A Gramática da LGP	38
3.5.3	Dicionário de Língua Gestual Portuguesa	39
3.5.4	Relação ao PE2LGP	39
4	Sistema 1: Modificação da Arquitetura	40
4.1	Exportação do Modelo para o Unity	41
4.1.1	Problemas Encontrados e Soluções	41
4.1.2	Sistema de Criação de Estados	44
4.1.3	Interface	44
4.2	Componente de Língua Natural	46
4.2.1	Componente de Processamento de Texto	46
4.2.2	REST API	48
4.2.3	Ligação à Interface	49
4.2.4	Alocação dos Dados	50
5	Sistema 2 - Criação Manual de Gestos	51
5.1	Criação do Ficheiro de Animação	52
5.2	Seleção da Configuração das Mãos	52
5.3	Criação da Animação	53
5.3.1	Feedback de Seleção	54
5.3.2	Feedback e Limitação da Rotação	55
5.3.3	Criação dos Frames	56
5.3.4	Interface	56
5.3.5	Conclusão da Animação	57
5.4	Plugin para Conversão de Animações	57
6	Sistema 3 - Criação Automática de Gestos	60
6.1	Introdução ao Sensor Kinect One	61
6.2	Mapeamento do Skeleton Tracking ao Modelo	62
6.3	Comandos de Voz	62
6.4	Melhorias e Extras	63
6.4.1	Limitações no Movimento Capturado	63
6.4.2	Múltiplas Câmaras para Feedback	63
6.4.3	Câmara Principal Associada ao Modelo	64

6.4.4	Contador Decremental na Criação da Animação	64
6.4.5	TPose	65
6.4.6	Nome da Animação	66
6.4.7	Seleção da Configuração das Mãos	66
6.5	Interface	66
7	Unificação dos Sistemas	67
7.1	Menu Principal	68
7.2	Visualizador das Animações	68
7.3	Usabilidade e Design	69
7.4	Conclusão	70
8	Avaliação	71
8.1	Métodos	72
8.1.1	Sistema 1	72
8.1.2	Sistemas 2 e 3 - Primeiro Momento	72
8.1.3	Sistemas 2 e 3 - Segundo Momento	73
8.2	Participantes	74
8.3	Métricas	74
8.3.1	Sistema 1	74
8.3.2	Sistemas 2 e 3 - Primeiro Momento	75
8.3.3	Sistemas 2 e 3 - Segundo Momento	75
8.4	Resultados	76
8.4.1	Sistema 1	76
8.4.2	Sistema 2 - Primeiro Momento	77
8.4.3	Sistema 3 - Primeiro Momento	79
8.4.4	Sistema 2 - Segundo Momento	81
8.4.5	Sistema 3 - Segundo Momento	82
8.4.6	Comparação entre os Sistemas 2 e 3	83
8.5	Conclusão	83
9	Conclusão	84
9.1	Dificuldades	85
9.2	Trabalho Futuro	86
	Bibliografia	87
	Apêndice A Anexos	93
A.1	Questionários	94
A.1.1	Modelo do Questionário do Sistema 1	94
A.1.2	Modelo do Questionário 1 dos Sistemas 2 e 3	95

A.1.3	Modelo e Resultados do Questionário 2 dos Sistemas 2 e 3	97
A.1.4	Resultados dos Questionários do Sistema 1	98
A.1.5	Resultados dos Questionários do Sistema 2	101
A.1.6	Resultados dos Questionários do Sistema 3	104

Lista de Figuras

2.1	“Pose Mode”.	20
2.2	Gesto “mulher”.	20
2.3	Interface (addon).	20
2.4	Visão Geral da Arquitetura de Alto Nível (Figura do Trabalho de Almeida [2014]).	20
2.5	Arquitetura Interna Relativa ao Processamento do Texto (Figura do Trabalho [Almeida, 2014]).	22
2.6	Correspondência da Base de Dados às Ações (Figura do Trabalho de Almeida [2014]).	23
3.1	Avatar de Bento et al. [2014].	26
3.2	Plataforma do Web de Bento et al. [2014].	26
3.3	App. Móvel do “Hugo”.	29
3.4	Integração do “Hugo” no Mundo Web.	29
3.5	Avatar da ProDeaf.	29
3.6	Plataforma Web Online da ProDeaf para Criar Gestos.	29
3.7	Avatar da Brava.	30
3.8	Processo de Criação da Brava.	30
3.9	Avatar do Projeto Sign 360.	31
3.10	Avatar da MocapLab.	31
3.11	Resultado do Trabalho de Sharp et al. [2015].	35
3.12	Pipeline do Sridhar et al. [2013].	35
3.13	Pipeline do Tagliasacchi et al. [2015].	36
3.14	Resultado do Tagliasacchi et al. [2015].	36
3.15	Exemplo do Algoritmo ICP.	36
3.16	Alinhamento do Algoritmo ICP.	36
3.17	Dispositivo UNI da à MotionSavvy.	37
3.18	Set que Incorpora o Leap Motion.	37
4.1	Arquitetura do PE2LGP Original.	41
4.2	Arquitetura do PE2LGP Atual.	41
4.3	Mapeamento Incorreto 1.	42
4.4	Mapeamento Incorreto 2.	42
4.5	Resultado Final.	42

4.6	Mapeamento Incorreto do <i>Rig</i> .	43
4.7	Mapeamento Correto do <i>Rig</i> .	43
4.8	Estados Base.	44
4.9	Estados do Input: “João”.	44
4.10	Estados do Input Exemplo.	44
4.11	Interface do PE2LGP Original.	45
4.12	Interface do PE2LGP Atual.	45
4.13	Pipeline do PE2LGP Original (Figura do Trabalho de Almeida [2014]).	46
4.14	Fluxo da Estrutura das Classes da Componente Linguística.	46
4.15	Funcionamento da Ligação entre a Componente de Língua Natural e a Interface.	49
4.16	Fluxo do Funcionamento dos Pedidos à REST.	50
5.1	Interface Relativa à Criação do Ficheiro de Animação.	52
5.2	Interface Relativa à Seleção da Configuração das Mãos.	53
5.3	Representação dos Pontos Onde é Possível Movimentar o Modelo.	54
5.4	Gizmo na Zona Seleccionada do Modelo (Ombro Esquerdo).	54
5.5	Objecto Utilizado para Feedback.	55
5.6	<i>Feedback</i> de Rotação do Ponto Seleccionado.	55
5.7	Painel Principal do Sistema 2.	56
5.8	Painel 1	57
5.9	Painel 2	57
5.10	Painel 3	57
5.11	Interface do Plugin para Converter Animações.	58
5.12	Animação Antes da Conversão.	59
5.13	Animação Depois da Conversão.	59
6.1	Exemplo do Tracking.	61
6.2	Referência às Duas Câmaras.	64
6.3	Exemplo do Contador Decremental.	65
6.4	Exemplo da TPose.	65
6.5	Interface para Controlar o Kinect.	66
7.1	Painéis do Menu Principal.	68
7.2	Interface Principal da Lista de Animações.	69
8.1	Resultados das Perguntas 4, 5 e 8, do Questionário do Sistema 1.	76
8.2	Resultados das Perguntas 1, 2, 3, 6, 7, 9 e 10, do Questionário do Sistema 1.	77
8.3	Resultados das Perguntas 1, 3 e 5, do Questionário do Sistema 2.	78
8.4	Resultados das Perguntas 1, 2, 3, 6, 7 e 9, do Questionário do Sistema 2.	78
8.5	Tempo de Criação, em Minutos, de Um gesto por Pessoa.	79
8.6	Resultados das Perguntas 1, 3 e 5, do Questionário do Sistema 3.	80

8.7	Resultados das Perguntas 1, 2, 3, 6, 7 e 9, do Questionário do Sistema 3.	80
8.8	Tempo de Criação, em Minutos, de Um gesto por Pessoa.	81
8.9	Número de Pessoas que Adivinharam os Gestos do Questionário do Sistema 2.	81
8.10	Resultados da Classificação dos Gestos do Sistema 2.	81
8.11	Resultados do Nível de LGP dos Avaliadores do Sistema 2.	82
8.12	Número de Pessoas que Adivinharam os Gestos do Questionário do Sistema 3.	82
8.13	Resultados da Classificação dos Gestos do Sistema 3.	82
8.14	Resultados do Nível de LGP dos Avaliadores do Sistema 3.	83
A.1	Modelo do Questionário 1 (Capa).	94
A.2	Modelo do Questionário 1 (2 de 4).	94
A.3	Modelo do Questionário 1 (3 de 4).	95
A.4	Modelo do Questionário 1 (4 de 4).	95
A.5	Modelo do Questionário 2 e 3 (2 de 4).	96
A.6	Modelo do Questionário 2 e 3 (3 de 4).	96
A.7	Modelo do Questionário 2 e 3 (4 de 4).	96
A.8	Modelo do Questionário 4 (Capa).	97
A.9	Seção 1 de 3 do Questionário 4.	97
A.10	Seção 2 de 3 do Questionário 4.	97
A.11	Seção 3 de 3 do Questionário 4.	97
A.12	Resultado do Questionário do Sistema 1 - Pessoa 1 de 6.	98
A.13	Resultado do Questionário do Sistema 1 - Pessoa 2 de 6.	98
A.14	Resultado do Questionário do Sistema 1 - Pessoa 3 de 6.	99
A.15	Resultado do Questionário do Sistema 1 - Pessoa 4 de 6.	99
A.16	Resultado do Questionário do Sistema 1 - Pessoa 5 de 6.	100
A.17	Resultado do Questionário do Sistema 1 - Pessoa 6 de 6.	100
A.18	Resultado do Questionário do Sistema 2 - Pessoa 1 de 6.	101
A.19	Resultado do Questionário do Sistema 2 - Pessoa 2 de 6.	101
A.20	Resultado do Questionário do Sistema 2 - Pessoa 3 de 6.	102
A.21	Resultado do Questionário do Sistema 2 - Pessoa 4 de 6.	102
A.22	Resultado do Questionário do Sistema 2 - Pessoa 5 de 6.	103
A.23	Resultado do Questionário do Sistema 2 - Pessoa 6 de 6.	103
A.24	Resultado do Questionário do Sistema 3 - Pessoa 1 de 5.	104
A.25	Resultado do Questionário do Sistema 3 - Pessoa 2 de 5.	104
A.26	Resultado do Questionário do Sistema 3 - Pessoa 3 de 5.	105
A.27	Resultado do Questionário do Sistema 3 - Pessoa 4 de 5.	105
A.28	Resultado do Questionário do Sistema 3 - Pessoa 5 de 5.	106

Abreviações

LGP Língua Gestual Portuguesa

SVO Sujeito-Verbo-Objeto

OSV Objeto-Sujeito-Verbo

VSO Verbo-Sujeito-Objeto

SOV Sujeito-Objeto-Verbo

OVS Objeto-Verbo-Sujeito

Agradecimentos

Em primeiro lugar, gostaria de deixar o meu agradecimento à minha família: os meus pais e o meu irmão. O meu percurso, profissional e pessoal, não seria possível sem o seu apoio incondicional.

Seria imprescindível não mencionar a orientadora desta tese, a Professora Luísa Coheur. O seu apoio, dedicação, e constante disposição em querer ajudar, foram fundamentais.

Agradeço à Professora Ana Mineira e ao seu aluno de Doutoramento, Tanyã Naves, ambos da Universidade Católica Portuguesa, envolvidos na LGP. Foram importantes durante o desenvolvimento da tese e também na avaliação. Além disso, sempre se mostraram disponíveis em ajudar.

Por fim, e não menos importante, deixo o meu agradecimento a todos os restantes envolvidos: aos colaboradores Pedro Fialho e Sérgio Curto do L2F/INESC-ID, a todos as pessoas que participaram na avaliação, e ao investigador do User Centered Technologies (UCT) research institute, Rumen Filkov.

Abstract

In Portugal there are about 33,000 deaf people of various degrees, including people suffering from congenital deafness. LGP is the sign language used by the deaf community in Portugal. It is executed mainly with hand movements, although facial and body expressions are also considered. In order to eliminate or reduce the communication barriers of deaf people, PE2LGP (From European Portuguese to Portuguese Sign Language) was built, which through a 3D character, translates, in real time, European Portuguese (in textual form).

In this thesis we improved and extended PE2LGP. Changes were made in terms of its architecture and two systems were developed in order to allow the introduction of new signs, without the need of having advanced knowledge in the animation area. In the first, allows to manually add signs to PE2LGP by interacting directly with the avatar through the mouse using an interface; in the second, signs are created with the help of Kinect One in order to automate the entire process of the first system, where the user's movements are converted into gestures.

PE2LGP was also exported to the game engine Unity, and the natural language component was converted into a REST API, breaking its dependency to Blender, which led to a more flexible system.

These systems were evaluated in order to understand if they could be used by people with different profiles, and also if the signs had quality enough to be understood by people that speak LGP. Both systems were easily used by the evaluators, but the produced signs were not totally understood by LGP users. Thus, many improvements are still needed before reaching a final product.

Keywords

Systems to Create Gestures, Portuguese Sign Language, Text Analysis, Animations, Unity

Resumo

Estima-se que em Portugal existam cerca de 33.000 surdos de vários tipos, incluindo as pessoas que sofrem de surdez congénita. A LGP é a língua gestual utilizada pela comunidade não ouvinte em Portugal e é executada através de movimentos com as mãos, tendo ainda em conta expressões corporais e faciais. No sentido de eliminar ou reduzir a barreira na comunicação destas pessoas, nasce o PE2LGP (do Português Europeu para a Língua Gestual Portuguesa), que através de um modelo 3D, traduz, em tempo real, Português Europeu em formato textual para LGP.

Nesta tese foi feita uma evolução do PE2LGP ao nível da arquitetura com o objectivo de agilizar o modo de introdução de novos gestos, sem que para isso sejam precisos conhecimentos avançados na área de animação. Assim, foram criados dois sistemas: um manual e outro automático. O primeiro permite ao utilizador criar novos gestos manipulando diretamente o modelo através de interações com o rato, usando para isso uma interface. O segundo tenta automatizar o processo de criação do primeiro, utilizando o sensor Kinect One, ou seja, a interação do utilizador com o modelo é maioritariamente feita através do sensor, onde os movimentos do utilizador são convertidos em gestos.

Além disso, o modelo e as respectivas animações do PE2LGP foram exportadas para o motor de jogo Unity, onde foi recriada e melhorada a interface, e a componente de língua natural foi convertida para uma REST API. Estas alterações permitiram quebrar a dependência que existia ao Blender.

A avaliação concentrou-se nos dois sistemas de criação de gestos, sendo que o principal objetivo foi perceber se eram acessíveis, para utilizadores com diferentes perfis, no que diz respeito à usabilidade e, posteriormente, perceber se os gestos teriam qualidade, em particular para a comunidade surda. Verificou-se que a usabilidade é adequada e que ambos os sistemas são importantes no contexto da LGP, contudo precisam de diversas melhorias para chegar a um produto final.

Palavras Chave

Sistemas para Criar de Gestos, Língua Gestual Portuguesa, Análise do Texto, Animações, Unity

1

Introdução

Conteúdo

1.1	Problema e Motivação	17
1.2	Objetivos	17
1.3	Contribuições	18
1.4	Organização do Documento	18

Estima-se que em Portugal existam cerca de 33.000 surdos¹ de vários tipos, incluindo as pessoas que sofrem de surdez congénita (à nascença).

A LGP é a língua gestual utilizada pela comunidade não ouvinte em Portugal e é executada através de movimentos com as mãos, tendo ainda em conta expressões corporais e faciais². Ao contrário da escrita, a LGP e todas as línguas gestuais são desenvolvidas de forma espontânea e independente entre comunidades surdas por todo o mundo [McNeill and Duncan, 2005]. A LGP é uma língua relativamente recente, pois nasceu na primeira escola de surdos, em 1823, na Casa Pia de Lisboa³. No final dos anos 70 começou a investigação da LGP, sobretudo ao nível das descrições linguísticas⁴. No entanto, a informação existente acerca da LGP é limitada e, por vezes, inconsistente.

O sistema PE2LGP (Português Europeu para Língua Gestual Portuguesa) [Almeida, 2014], começou a ser desenvolvido em 2014 por Inês Almeida no âmbito da sua tese de Mestrado em Engenharia Informática e de Computadores. O PE2LGP, traduz, em tempo real, Português Europeu em formato textual para LGP, sendo que os gestos são realizados por uma personagem virtual em 3D, a Catarina. O sistema contém uma interface que permite a interação com o personagem. O projeto consistiu no desenvolvimento de duas componentes. A primeira relativa à produção animações para o personagem 3D que correspondem a gestos na LGP. A segunda diz respeito à criação de uma componente de análise e processamento do texto inserido pelo utilizador através da interface. Nesta tese foi feita uma evolução ao PE2LGP.

1.1 Problema e Motivação

O protótipo do trabalho de Inês Almeida contém uma pequena quantidade de gestos. Isto significa que produz poucas frases em LGP. Para agravar, a introdução de gestos, uma vez que envolve a criação de animações 3D, a primeira versão do PE2LGP requer conhecimentos avançados nas áreas de animação e, em especial, da ferramenta Blender. Além disso, o tempo de produção desta tarefa é também um problema, pois criar um único gesto no protótipo poderá atingir até várias horas de trabalho, dependente da complexidade. Deste modo, o objetivo desta tese é eliminar ou reduzir estas limitações.

1.2 Objetivos

O principal foco desta tese reside na criação de dois sistemas capazes de agilizar o modo de introduzir novos dados, sem que para isso sejam necessários conhecimentos avançados na área de modelação e animação 3D. Os sistemas são independentes e tentam produzir o mesmo resultado, porém de maneiras diferentes, mas que se podem complementar. Por um lado, temos um sistema manual, que fará uso apenas de uma interface para a criação de gestos. Por outro, temos um sistema

¹<http://associacaosurdosaltaestremadura.org/index.php?id=44>

²<http://www.disabled-world.com/disability/types/hearing/communication/>

³http://www.apsurdos.org.pt/index.php%3Foption%3Dcom_content%26view%3Darticle%26id%3D41%26Itemid%3D8

⁴ver nota 3

automático, que tenta explorar as capacidades do sensor Kinect e, assim, automatizar o processo na sua totalidade.

1.3 Contribuições

De uma maneira geral, as contribuições feitas no PE2LGP foram:

1. Modificação da Arquitetura Atual:
 - (a) Exportação do Protótipo para o Unity.
 - (b) Enriquecimento da Componente Linguística.
2. Criação do Sistema de Animações Manual.
3. Criação do Sistema de Animações Automático.
4. Unificação dos Sistemas.
5. Avaliação dos Sistemas.

1.4 Organização do Documento

O [Capítulo 2](#) apresenta o protótipo inicial do PE2LGP. O [Capítulo 3](#) faz referência a todos os trabalhos relacionados investigados, com ênfase na produção de avatares e no uso do sensor Kinect.

Os capítulos seguintes ([Capítulo 4](#), [Capítulo 5](#), [Capítulo 6](#) e [Capítulo 7](#)) correspondem aos itens 1-4 da lista definida na seção anterior. No [Capítulo 4](#) apresenta-se a modificação da arquitetura do PE2LGP com destaque na introdução do motor de jogo Unity e na criação de uma *REST API* que serve como comunicação à componente linguística. Depois, no [Capítulo 5](#) e no [Capítulo 6](#) são apresentados ambos os sistemas de criação de animações: um manual e o outro automático. O [Capítulo 7](#) foca-se na unificação dos 3 sistemas resultantes dos 3 capítulos anteriores.

Por fim, e de modo a concluir o trabalho, no [Capítulo 8](#) são apresentados os métodos de avaliação e os respectivos resultados e, no [Capítulo 9](#), são listadas as contribuições e são definidos os objetivos para o trabalho futuro.

2

PE2LGP - From European Portuguese to Portuguese Sign Language

Conteúdo

2.1	Visão Geral	20
2.2	Implementação	21
2.3	Avaliação	24

Este capítulo tem como objetivo apresentar o PE2LGP realizado por Inês Almeida em 2014 que está na base desta tese (detalhes sobre o protótipo original podem ser encontrados em [Almeida, 2014], [Almeida et al., 2015a] e [Almeida et al., 2015b]).

2.1 Visão Geral

O PE2LGP traduz, em tempo real, Português Europeu¹ em formato textual para LGP, sendo os gestos realizados por um agente virtual, a Catarina (Figura 2.1 e Figura 2.2). O sistema contém uma interface (Figura 2.3), que permite a interação com um personagem 3D através de texto.



Figura 2.1: "Pose Mode".



Figura 2.2: Gestos "mulher".

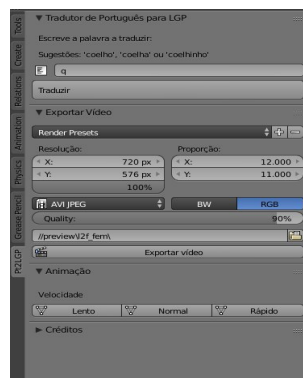


Figura 2.3: Interface (addon).

Este projeto divide-se em três componentes: a primeira é relativa ao tratamento de dados introduzidos através da interface (gloss-translate), a segunda refere-se à conexão entre o texto inserido pelo utilizador e as animações (lookup) e, por fim, a terceira componente diz respeito à animação do personagem 3D (animate). A Figura 2.4 representa a visão geral da arquitetura atual do protótipo. Em seguida segue uma descrição de cada componente.

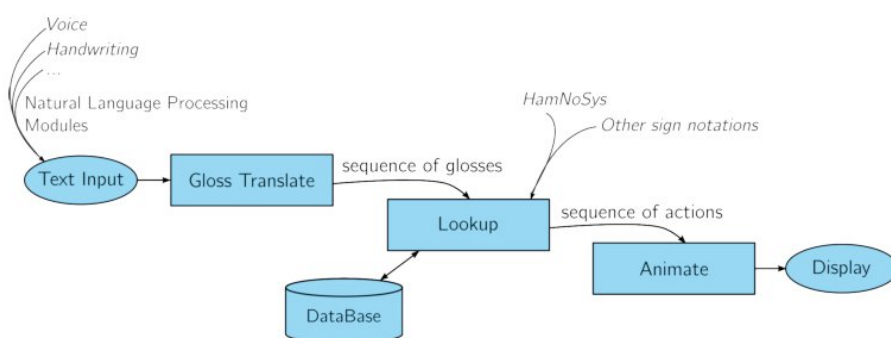


Figura 2.4: Visão Geral da Arquitetura de Alto Nível (Figura do Trabalho de Almeida [2014]).

¹Português Europeu é a designação dada à variedade linguística da língua portuguesa falada em Portugal continental, nas regiões autónomas da Madeira e dos Açores, e pelos emigrantes portugueses espalhados pelo mundo, englobando os seus dialectos regionais, vocabulário, gramática e ortografia.

2.1.1 Gloss Translate

O protótipo segue uma abordagem *gloss-based*, quer isto dizer que as palavras são associadas ao seu significado através de um dicionário. Na *gloss notation* cada gesto é denotado com uma palavra (*gloss*) que melhor corresponde ao significado do gesto. Os *glosses* são calculados de acordo com as regras da gramática da LGP (que tem como base o trabalho de Amaral et al. [1994]), como por exemplo, a ordem SVO. Por exemplo, utilizando a ordem SVO, a sequência de gestos para a frase “qual é o teu nome” seria “teu nome qual”. Assim sendo, esta componente é responsável pelo cálculo dos *glosses*.

2.1.2 Lookup

O *Lookup* tem como objetivo obter uma sequência de identificadores de ações a partir de uma sequência de *glosses*, para que posteriormente possa ser possível fazer a conexão às animações. Um dos obstáculos enfrentados por esta componente é o fato de muitas palavras Portuguesas não terem um único significado em LGP. Uma palavra pode significar diferentes palavras na LGP dependendo do contexto da frase.

2.1.3 Animate

Esta componente recebe um conjunto de ações para serem compostas numa animação fluída, seguido de um conjunto de sugestões de como fazê-lo. Por exemplo, o *gloss* pode ser *fingerspelled* ou então ter outro tratamento. O *fingerspell* é responsável por representar a palavra, letra a letra, sendo que, para cada um, existe uma animação correspondente. O outro tratamento diz respeito a animações criadas exclusivamente para determinadas palavras.

2.2 Implementação

Apresenta-se, nas seguintes sub-seções, a prova de conceito da arquitetura, descrevendo as escolhas e particularidades do sistema.

2.2.1 Blender

Para a animação e modelação do personagem 3D, foi escolhido o programa Blender². Esta escolha deve-se ao fato de ser um projeto *open-source*, com uma vasta comunidade de artistas e desenvolvedores. Para além disso, contém uma API flexível que permite operar sobre toda a informação através de *scripting*.

2.2.2 Filosofia dos Addons

O uso do Blender obriga a outras escolhas, como a ferramenta de criação da interface para comunicação com o personagem 3D. O Blender contém uma forma de interação com toda a sua

²<https://www.blender.org/>

informação usando para isso a API escrita na linguagem Python³ para scripts. Existe também uma interface para o mesmo objetivo, que pode ser customizável, onde se pode adicionar novos botões e painéis. A esta customização dá-se o nome de *addon*. No fundo, um *addon* é um conjunto de alterações na interface e portanto neste sentido foi criado um para a comunicação com o personagem 3D (Figura 2.3).

2.2.3 Processamento e Tradução da Língua Natural

O processamento da Língua Natural Portuguesa, como por exemplo, a ordem das palavras, é diferente da LGP. Por essa razão, são precisas algumas transformações de modo a que através da primeira possamos chegar à segunda. Para este efeito foi seguido um pipeline (Figura 2.5) que se foca sobretudo no processamento do texto, através de ferramentas da área de processamento da Língua Natural. Em seguida explicam-se algumas das componentes do pipeline, que recebe como entrada uma palavra ou frase introduzida pelo utilizador (*Text input*).

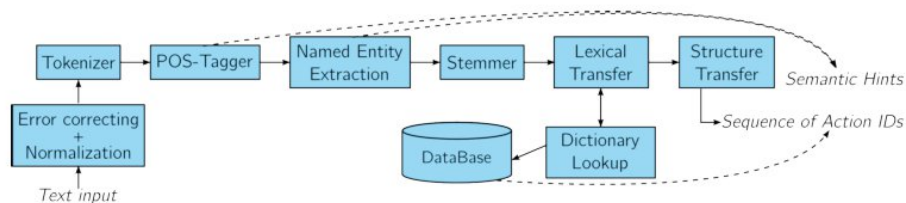


Figura 2.5: Arquitetura Interna Relativa ao Processamento do Texto (Figura do Trabalho [Almeida, 2014]).

- **Error Correcting + Normalization:** Procede à correção de erros ortográficos e à normalização do *input*, que pretende colocar o texto numa única forma, por exemplo, colocando tudo em letras minúsculas.
- **Tokenizer:** Separa o texto inserido através da tokenização das palavras e de pontuações. Por exemplo, a frase “O João come a sopa” depois de ter passado pela fase anterior, ficaria na seguinte forma:

[“o” , “joao” , “come” , “a” , “sopa”]

- **POS-Tagger:** São usados *taggers* (classificadores) baseados em n-gramas da NLTK⁴, começando com um *tagger* de bigramas e usando como classificação por omissão a categoria “nome”, por ser a classe mais comum em Português. O corpus usado para treino dos taggers é o da floresta sintáctica [Afonso et al., 2002]. O exemplo anterior com a frase “O João come a sopa” ficaria:

[(“o” , art) , (“joao” , prop) , (“come” , v-inf) , (“a” , art) , (“sopa” , n)]

³<https://www.python.org/>

⁴<http://www.nltk.org/>

- **Name Entity Extraction:** Identifica nomes que são entidades, de modo a poderem ser tratados de forma diferente. Esta identificação é feita através da consulta a um ficheiro de texto que contém um nome por linha de cada entidade. Esse ficheiro é construído manualmente. Por exemplo, dado como *input* “Pedro Passos Coelho”, o sistema não interpretaria “Coelho” como um animal, mas sim como um nome pertencente à sequência e, por essa razão, o gesto é diferente de “coelho” como animal.
- **Stemmer:** Analisa os sufixos e prefixos de uma palavra. Por exemplo, na palavra “*coelhinha*”, seria detetado o sufixo “inha”, que nos permite saber que é algo de pequenas dimensões e, através da terminação “a”, que tem o gênero feminino. Assim, se quiséssemos traduzir a palavra “*coelhinha*” para LGP, esta resultaria em 3 palavras diferentes: “*coelho*”, “*mulher*” e “*pequeno*”, seguindo esta ordem na LGP e na animação.
- **Lexical Transfer:** Converte a lista de palavras anotadas para os seus respetivos *glosses* usando um dicionário. O funcionamento deste processo é feito através da referência à base de dados, utilizando para isso o seguinte formato de dados:

[“GLOSS”, [“sopa”]] e [“FINGERSPELL”, [“j”, “o”, “a”, “o”]]

- **Structure Transfer:** Suporta reordenação de adjetivos e quantidades no final, afetando o nome. Por exemplo, o *input* “*dois coelhos*” resultaria em:

[[“GLOSS”, [“coelho”]], [“NUMERAL”, [“2”]]]

Além disso, o protótipo também suporta reordenação na sequência “nome-verbo-nome”, tentando assim converter frases na forma SVO usada na Língua Natural Portuguesa⁵, na estrutura mais comum, a OSV (até à data de 2014), usada na LGP.

2.2.4 Lookup

Este é o passo final que faz a ligação do texto processado ao personagem 3D, através de uma base de dados, de modo a fazer os gestos correspondentes. A base de dados consiste num conjunto de *glosses* e num conjunto de ações. As ações são identificadas por ID's, que são depois mapeadas em animações no Blender. Cada palavra está associada a uma ou mais ações que, por sua vez, representam animações no personagem 3D, tal como se pode verificar através da Figura 2.6.

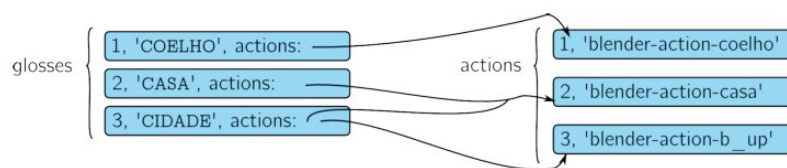


Figura 2.6: Correspondência da Base de Dados às Ações (Figura do Trabalho de Almeida [2014]).

⁵<http://www.portaldalinguaportuguesa.org/?action=terminology&act=view&id=1454>

2.2.5 Personagem 3D

Depois de modelado, a configuração do personagem consiste no *Rigging* e do *Skinning*⁶. A melhor maneira de animar um objeto complexo é através de *Armatures*⁷. Estas atuam como um esqueleto, ou seja, é possível mover os ossos desse esqueleto formando a animação do personagem. O processo de construção de uma *Armature* é chamado *Rigging*. Basicamente são os ossos de um modelo 3D, sendo o *Skinning* o processo de associar a *Armature* ao modelo.

2.2.6 Animações

Todas as animações foram feitas keyframe a keyframe⁸. Existe ainda uma animação comum a todas as outras, a “pose mode” (Figura 2.3). Cada animação começa e acaba na “pose mode”. É importante salientar que existem apenas 2 tipos de animações para cada palavra. Isto significa que, ou a palavra introduzida tem uma animação específica ou então esta é processada através do *fingerspell*. Para exemplificar, dado como *input* “coelho” como um animal, o gesto será apenas um, mas se considerarmos como uma entidade, será representado pelo conjunto das animações dos caracteres “c”, “o”, “e”, “l”, “h” e “o”.

2.3 Avaliação

Foram consultados membros da comunidade de surdos e de 2 associações, entre eles intérpretes e surdos. Ambas as associações pareceram céticas em relação à possibilidade de se conseguir traduções precisas ou animar um vocabulário suficiente para um produto final. Ainda assim gostaram da ideia, mesmo que o sistema fosse limitado e só funcionasse para palavras soltas. A interface provou ser utilizável, mas não é robusta o suficiente, devido ao fato de, por vezes, se saltar para menus que não fazem parte da interface de comunicação com o personagem. Por essa razão na maioria das vezes não conseguiam voltar atrás. O personagem 3D fazia um gesto e era pedido para identificarem esse gesto. Todos responderam corretamente e a qualidade na tradução mereceu um feedback positivo. No entanto, todos mencionaram a falta de expressões faciais.

⁶http://wiki.blender.org/index.php/Doc:2.4/Books/Essential_Blender/07.1.Rigging_and_Skinning:_Hands_On

⁷<http://wiki.blender.org/index.php/Doc:2.4/Manual/Rigging/Armatures>

⁸Uma keyframe em animação e produção cinematográfica é um desenho que define os pontos de início e fim de qualquer transição entre imagens

3

Trabalho Relacionado

Conteúdo

3.1	Uso de Avatares para Língua Gestual	26
3.2	Desafios na Criação de Avatares para Língua Gestual	31
3.3	Criação e Manipulação de Objetos Usando o Kinect	32
3.4	Rastreamento das Mãos (Hand Tracking)	33
3.5	Recursos da Área de Processamento de Língua Natural	38

Neste capítulo apresenta-se a investigação relativa ao trabalho relacionado. Em primeiro lugar, são apresentados avatares utilizados no âmbito da língua gestual de vários países, com particular foco no Brasil. Depois, apresento mecanismos para manipular e criar avatares a partir do sensor Kinect. Em seguida, é feita uma análise aos mecanismos do rastreamento das mãos, utilizando o Kinect e o Leap Motion. Por fim, são descritos recursos para manipular e processar texto em língua natural, muitos deles já utilizados no PE2LGP.

3.1 Uso de Avatares para Língua Gestual

Os avatares, no contexto digital, são personagens virtuais que têm como objetivo executar tarefas realizadas por humanos. São cada vez mais utilizados em diversas áreas: publicidade, redes sociais, cinema, língua gestual, entre outras. Assim, no que diz respeito à língua gestual, os avatares são úteis porque parte das pessoas portadores de deficiência auditiva têm problemas significativos na leitura e, portanto, o conteúdo de, por exemplo, páginas web não está totalmente acessível [Kipp et al., 2011]. Uma alternativa aos avatares poderá ser o uso de vídeos gravados. Porém, estes são estáticos e não podem ser utilizados em cenários dinâmicos, assim como é a Internet.

Em seguida, identifico os principais avatares de diferentes países.

3.1.1 Avatares na Língua Gestual Portuguesa

O trabalho criado por José Bento em 2014 intitulado “Avatares em Língua Gestual Portuguesa” [Bento et al., 2014], teve como principal objetivo criar um intérprete virtual para expressar Língua Gestual Portuguesa. Pretendia-se interagir com o intérprete virtual através de uma plataforma Web que permite dar a conhecer os mais diversos gestos básicos da Língua Gestual Portuguesa utilizados no dia-a-dia, como por exemplo, o nome das cores, frutas e outros.

A captura do movimento (facial, corporal e ao nível das mãos, mais propriamente ao nível dos dedos) foi feita através da primeira versão do Kinect. Posteriormente, algumas falhas na captura, nomeadamente ao nível do detalhe dos dedos foram reparadas recorrendo a diferentes ferramentas e algoritmos.



Figura 3.1: Avatar de Bento et al. [2014].

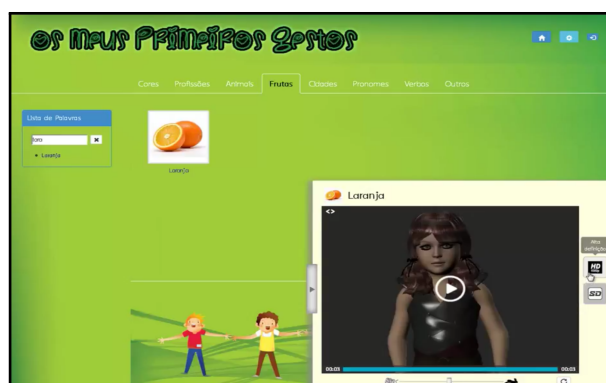


Figura 3.2: Plataforma do Web de Bento et al. [2014].

Foram criados 35 gestos através de um método híbrido que utiliza duas fases: uma automática

e outra manual. Numa primeira fase, é feita a captura de movimentos do corpo. Na segunda fase é feita uma animação manual para descrever o movimento da mão e ambos criam uma animação que corresponde a um gesto. Os 35 gestos criados incluem pronomes, substantivos, verbos entre outros. A partir desses gestos é possível formar diversas frases, através da regra [OSV](#).

3.1.1.A Ferramentas de Modelação e Animação 3D

Para a modelação do personagem 3D foi utilizado o programa Autodesk 3DS MAX e para as animações, o Autodesk Motionbuilder¹. O Motionbuilder é um software de animação usado para animar modelos 3D, ou seja, depois de criado o esqueleto do personagem no 3DS MAX, este foi depois animado no Motionbuilder. Por fim, foi usado o Brekel Kinect Pro Body², que é uma aplicação exclusiva para o sistema operativo Windows, que em conjunto com o dispositivo Kinect, permite aos animadores 3D fazerem uma captura do corpo e depois manipular todos os dados capturados.

3.1.1.B Tipos de Animações

Foram utilizados dois tipos de animações. O primeiro, usando apenas o Motionbuilder que consiste na criação manual das animações, ou seja, o processo é feito todo pelo animador, passo a passo. O segundo é assistido, utilizando o Brekel Kinect Pro Body num processo de produção mais complexo, isto é, não basta capturar os dados através do programa e utilizá-los diretamente no intérprete virtual, é preciso passar pelos seguintes passos:

1. Captura do movimento.
2. Ligação dos dados do Kinect ao Avatar.
3. Tratamento de dados (Data Cleaning).
4. Adição da animação da mão à animação corporal do avatar.

Os primeiros 2 passos são feitos através do Brekel Kinect Pro Body, onde são produzidos ficheiros em BVH. Estes consistem num formato de dados próprio, para captura de movimento, onde é armazenada informação hierárquica do movimento. Basicamente contêm uma representação nas coordenadas [x, y, z] de cada articulação, incluindo a rotação e translação. Quanto à etapa do Data Cleaning, esta consiste em determinar a posição correta de cada articulação no conjunto de instantes de captura (corrigir posições que possam estar erradas). Além disso, é necessário corrigir posições que ficam erradas ao passar para o intérprete virtual (devido à fisionomia do esqueleto não ser totalmente igual à do avatar e depender da pessoa a quem se captura o movimento). A ligação do Brekel Kinect Pro Body ao Autodesk Motionbuilder é feita carregando o ficheiro com a extensão *.BVH* gerado que atua automaticamente no intérprete virtual. Foi necessário algum trabalho extra para conseguir detetar as mãos, sobretudo ao nível do detalhe dos dedos, isto porque a primeira versão do Kinect não tem captura ao nível da posição dos dedos. Assim sendo, utilizando o Kinect, foram aplicados dois algoritmos com as seguintes propriedades:

¹<https://www.google.pt/search?hl=pt&q=motionbuilder>

²<http://brekel.com/brekel-kinect-pro-body/>

1. Assume que as mãos são as extremidades mais próximas do sensor.
2. Desenha elipses à volta da articulação de cada mão.

Para complementar estes dois algoritmos foi utilizada a framework de processamento de imagem e Inteligência Artificial, a AForge.NET³. Além disso, são usadas várias técnicas de modo a capturar o contorno dos dedos. Uma dessas técnicas chama-se *finger tracking* semiautomático, ou seja, o processo de deteção envolve o utilizador para marcar as posições das articulações dos dedos.

3.1.1.C Captura da Expressão Facial

A expressão facial foi capturada de forma semelhante à postura corporal, utilizando um software da mesma entidade, chamado Brekel Kinect Pro Face⁴. Este programa consiste exatamente no mesmo que o Brekel Pro Body mas este dedica-se à captura facial. Além do Brekel Pro Face, foi usado o FaceShift Studio⁵, que é um software de captura de movimentos faciais, que analisa o movimento de um ator e descreve-o como uma mistura de várias expressões básicas, incluindo a orientação da cabeça e a direção dos olhos. Esta descrição gerada pelo software é depois utilizada para animar personagens virtuais, como é o caso. Segundo a análise do autor, as expressões faciais não são muito importantes face às palavras soltas, mas em relação às frases fazem toda a diferença [Bento et al., 2014]. As expressões faciais foram integradas nos pronomes interrogativos, expressões de surpresa entre outros.

3.1.2 Avatares na Língua Gestual Brasileira

Em seguida serão apresentadas 3 entidades que desenvolvem avatares para a Língua Gestual Brasileira (Libras) no Brasil: HandTalk, ProDeaf e BRAVA.

3.1.2.A HandTalk

A HandTalk⁶ é uma startup brasileira que consiste numa plataforma de tradução de Português para Libras utilizando um personagem 3D, denominado “Hugo”. A HandTalk tem uma base semelhante ao PE2LGP, com a particularidade de traduzir Português do Brasil para a Língua de Sinais Brasileira (Libras) em vez de traduzir Português Europeu para Língua Gestual Portuguesa. O personagem 3D e as respetivas animações foram criadas no Blender.

Neste momento a HandTalk tem como foco ajudar a traduzir o mundo Web e móvel para os mais incapacitados auditivamente⁷, ou seja, o personagem consegue integrar-se em dispositivos móveis⁸ (Figura 3.3) e no mundo Web (Figura 3.4), sendo que neste último traduz todo o conteúdo das páginas em que está integrado.

Em 2012 foi eleita como a empresa mais inovadora do Brasil e em 2013 como a melhor aplicação social do mundo, no WSA-mobile, evento organizado pela Organização das Nações Unidas (ONU).

³<http://www.aforge.net.com/>

⁴<http://brekel.com/brekel-kinect-pro-face/>

⁵<http://www.faceshift.com/>

⁶<http://www.handtalk.me/>

⁷<http://projetodraft.com/hand-talk-a-startup-de-alagoas-que-esta-ajudando-os-surdos-brasileiros-se-comunicarem-com-o-mundo/>

⁸<https://itunes.apple.com/pt/app/hand-talk-tradutor-para-libras/id659816995?mt=8>



Figura 3.3: App. Móvel do “Hugo”.

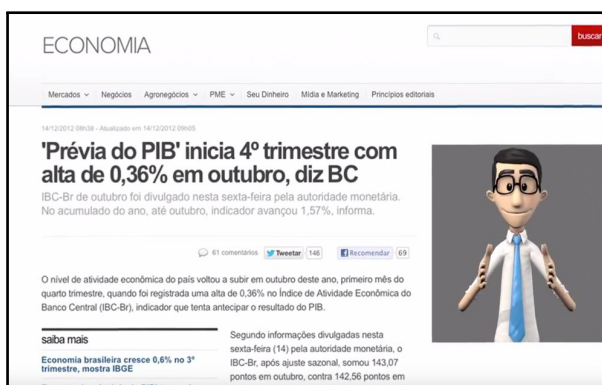


Figura 3.4: Integração do “Hugo” no Mundo Web.

3.1.2.B ProDeaf

A ProDeaf⁹, criada na Universidade Federal de Pernambuco é, tal como a HandTalk, uma startup Brasileira vocacionada no uso de avatares como meio de comunicação. É possível traduzir texto e voz de Português do Brasil para Libras, em que um personagem executa os gestos correspondentes (Figura 3.5). No entanto, a ProDeaf diferencia-se da HandTalk pelo uso da modalidade *Crowdsourcing*. Quer isto dizer, que qualquer pessoa poderá contribuir para o ProDeaf com novos dados através da plataforma disponível online¹⁰ (Figura 3.6). A plataforma dispõe ainda de uma série de opções que permitem criar um gesto, que vão desde a seleção da expressão facial, à posição e orientação de postura corporal, entre outros. A combinação de todas estas opções dão origem a uma animação correspondente a um gesto para uma palavra.

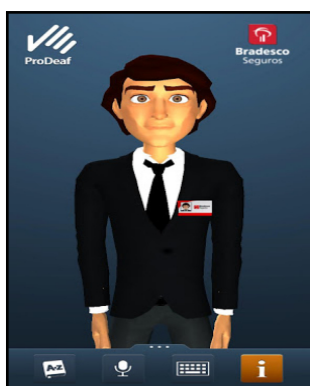


Figura 3.5: Avatar da ProDeaf.

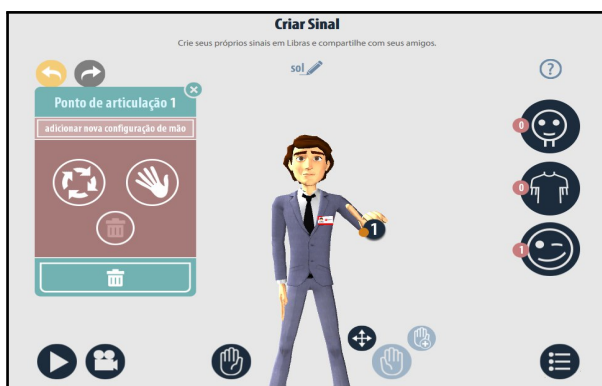


Figura 3.6: Plataforma Web Online da ProDeaf para Criar Gestos.

Neste momento, a ProDeaf em parceria com a Microsoft, está focada em aplicar a tradução entre várias línguas, como, por exemplo, de Libras (Brasil) para ASL (American Sign Language ou Língua Gestual Americana)¹¹.

⁹<http://www.prodeaf.net/>

¹⁰<http://web.prodeaf.net/>

¹¹https://www.youtube.com/watch?v=Pb_xh7MNH54

3.1.2.C BRAVA

A BRAVA é uma empresa de Tecnologia de Informação e Comunicação que desenvolve projetos inovadores de pesquisa que visam proporcionar autonomia a pessoas e dispositivos. Neste contexto surge o avatar da [Figura 3.7](#) que, tal como os anteriores, traduz texto em Português do Brasil para Libras. No entanto, a inovação aqui é o processo de criação, que é feito através de *Motion Capture* (Captura de Movimentos) ([Figura 3.8](#)) em conjunto com um par luvas que contém 12 sensores que permitem capturar com precisão as mãos, bem como os todos os dedos.

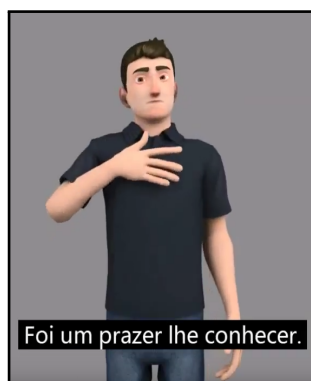


Figura 3.7: Avatar da Brava.

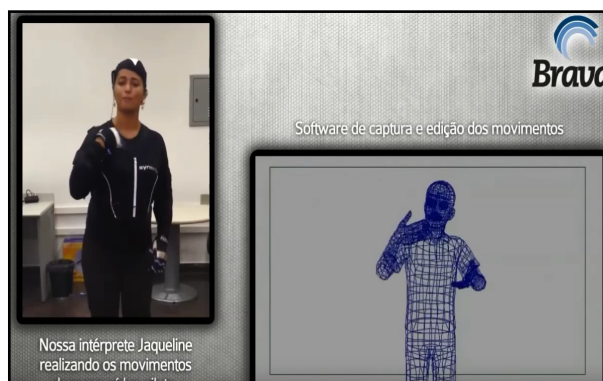


Figura 3.8: Processo de Criação da Brava.

O *Motion Capture* é o processo de gravar um evento de um movimento em tempo real e traduzi-lo em termos matemáticos de modo a obter uma representação 3D do seu desempenho [Menache, 2000]. Por outras palavras, qualquer movimento realizado por uma pessoa é gravado e utilizado para animar avatares. Estas ações, podem, como é o caso, ser capturadas com o auxílio de diversos sensores localizados numa roupa especial usada pelo ator que realiza os movimentos. Desta forma, consegue-se capturar todos os detalhes da pessoa a movimentar-se e reproduzi-los, em tempo real, no avatar.

3.1.3 Avatares noutras Línguas Gestuais

O trabalho intitulado “A prototype Malayalam to Sign Language Automatic Translator” [Joy and Balakrishnan, 2014], apresenta um sistema que recebe texto em Malásio como entrada e gera a língua gestual correspondente utilizando um avatar para a reprodução dos gestos. As animações foram criadas através de um software gratuito do sistema Debian GNU Linux. O utilizador pode interagir com o personagem através de uma caixa de texto. A maioria dos gestos utilizados foram adoptados pela Língua Gestual Indiana. O sistema também pode ser usado como ferramenta de aprendizagem da Língua Gestual Kerala¹², uma vez que existe uma falta de standardização da Língua Gestual Malaia. As fases que incorporam o pipeline que processa o texto são semelhantes às utilizadas no PE2LGP [Almeida, 2014] (Input text, Pos Tagger, Optimizer, Stemming, Animation). Existe uma componente de processamento do texto, dividida em várias tarefas, e outra componente responsável pela animação. Na primeira componente é importante referir que segue a regra SOV,

¹²Querela ou Kerala é um dos 28 estados da Índia.

usada na Língua Gestual Indiana. Este sistema pode ser estendido para converter páginas Web.

A MocapLab¹³ é uma empresa Francesa que é 100 por cento especializada em *Motion Capture* que, além de vender um conjunto de serviços usando esta metodologia, faz investigação e cria nesta área.

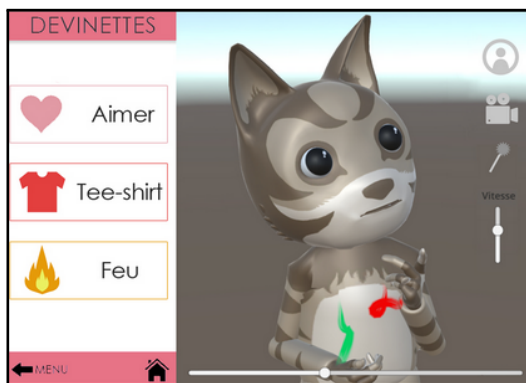


Figura 3.9: Avatar do Projeto Sign 360.

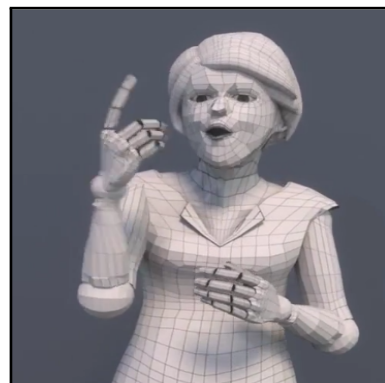


Figura 3.10: Avatar da MocapLab.

Um dos projetos é o Sign 360¹⁴ que consiste numa aplicação (Figura 3.9) para aprender LSF (Língua Gestual Francesa). Porém, o projeto não se fica por aqui, o objectivo é expandir a língua gestual a vários países: ASL (American Sign Language), BSL (British Sign Language), NZSL (New Zealand Sign Language), entre outros. Este projeto destaca-se dos outros por utilizar *Motion Capture* e assim criar avatares com uma qualidade de excelência¹⁵(Figura 3.10).

Existem ainda outros trabalhos noutras línguas que seguem o mesmo conceito: o projeto ATLAS [Barberis et al., 2011] que consiste num sistema de tradução automática de Italiano para Língua Gestual Italiana, o trabalho de Jemni et al. [2013] que traduz Árabe para Língua Gestual Árabe ou o trabalho de Ittisam and Toaditthep [2010] que consiste num editor de animações que representam gestos em Língua Gestual Tailandesa.

3.1.4 Relação com o PE2LGP

Estes trabalhos foram úteis para o PE2LGP uma vez que foi possível verificar quais são as tendências e os mecanismos utilizados atualmente. Por exemplo, o sistema 2 apresentado no Capítulo 5, foi influenciado pela plataforma *crowdsourcing* de criação de gestos da ProDeaf. Além disso, tanto na HandTalk, como na ProDeaf, utilizam o motor de jogo Unity, o que ajudou à escolha. Já o trabalho de Bento et al. [2014] permitiu-me saber quais foram as ferramentas utilizadas no âmbito do *tracking* utilizando o sensor Kinect.

3.2 Desafios na Criação de Avatares para Língua Gestual

Independentemente da língua, existem vários desafios na criação dos avatares [Ossada, 2015] [Kipp et al., 2011], dois quais destaco dois: a criação de animações que aproximem o mais possível

¹³<http://www.mocaplab.com/>

¹⁴<http://www.mocaplab.com/projects/sign-360-2/>

¹⁵<https://www.youtube.com/watch?v=oRM4nZnokow>

os movimentos do avatar aos de um humano e o garantir a compreensão dessas animações pela parte dos portadores de deficiência auditiva. O mais complicado de ultrapassar é o segundo desafio, uma vez que a língua gestual não tem um sistema universal de escrita, nem é coerente entre países, nem mesmo em diferentes regiões do mesmo país. Para complicar, existem poucos estudos sobre a língua gestual, em particular a LGP, e o modo como esta é executada. Além disso, a língua gestual é uma linguagem multimodal onde as mãos, os braços, a cara e o corpo podem ter as mais variadas formas e devem ser sincronizados de acordo com o contexto do gesto. Desta forma, a qualidade de produção de gestos de um avatar varia entre 58-62 por cento, tal como concluiu o trabalho de Kipp et al. [2011].

Quanto ao primeiro desafio, ou seja, tentar com que os movimentos de um avatar se aproximem aos de um humano, com a utilização de *Motion Capture*, é possível obter uma captura bastante precisa até mesmo ao nível das mãos e dos dedos. Porém, o custo de alguns equipamentos é elevado.

3.3 Criação e Manipulação de Objetos Usando o Kinect

Nesta seção apresentam-se projetos que fazem uso do Kinect sobretudo em duas vertentes: a criação e a manipulação. Ambas fazem uso da técnica *Motion Capture* utilizada em diversas áreas tais como: na Medicina [Liu et al., 2014, Das et al., 2011, Gong et al., 2015], na Robótica [Choi et al., 2013, Özgür et al., 2014], nos Transportes [Palacio and Arévalo, 2014], na Educação [Leite and Orvalho, 2013, Bruciati and Rossignol, 2015, Adell and Castañeda Quintero, 2012], e na Realidade Aumentada [Chan et al., 2011] com ênfase nos jogos [Rincon et al., 2016], entre outras. Esta é uma área em que a investigação não tem parado, sobretudo nos últimos anos com o aparecimento dos sensores de profundidade de baixo custo [Herda et al., 2001, Wu et al., 2009, Zhang and Guo, 2013]. O lançamento da segunda versão do Kinect em 2014 contribuiu para este crescimento, isto porque as atualizações do novo equipamento, quer a nível de software, quer a nível de hardware, por omissão, já fazem um *tracking* do corpo humano razoável em relação às versões anteriores.

No contexto da criação de avatares, o mais comum é utilizar o sensor para fazer a reconstrução de objetos ou de pessoas, criando assim um modelo digital. Projetos deste tipo podem utilizar múltiplos sensores como é o caso do [Tong et al., 2012]. Seguindo a mesma vertente da reconstrução, existem programas que tentam fazê-lo, recorrendo, não só ao Kinect como também a outros sensores, como o KSCAN3D¹⁶ e o SKANECT¹⁷.

Quanto à manipulação de avatares usando *Motion Capture*, temos o trabalho de Bholsithi et al. [2014] que pretende controlar avatares a partir do movimento capturado através do Kinect. Para isso, utilizam as bibliotecas de Inteligência Artificial Open NI com o *middleware* NITE Primesense¹⁸. Na medicina, o trabalho de Gong et al. [2015], pretende criar um avatar que represente um terapeuta físico, para assim guiar o processo de reabilitação dos paciente, uma vez que não é necessária a

¹⁶<http://www.kscan3d.com/>

¹⁷<http://skanect.occipital.com/>

¹⁸<http://structure.io/openni>

presença física. O projeto de [Costa et al. \[2014\]](#) consiste em reconstruir partes do corpo humano, nomeadamente o peito do corpo humano feminino, para assim ajudar no planeamento das cirurgias.

3.3.1 Softwares

Brekel Kinect¹⁹ é uma aplicação que usa o sensor Kinect para permitir a captura de objetos tridimensionais e exportá-los de modo a serem usados em ambientes 3D. Também permite o rastreio do esqueleto para modelação e captura do movimento. É uma aplicação gratuita para uso comercial e privado.

Faceshift Studio²⁰ é um software que utiliza a técnica *Motion Capture* com ênfase nas expressões faciais. O software analisa os movimentos faciais do ator e descreve-os como uma mistura de expressões básicas, bem como a orientação da cabeça e a direção dos olhos. Essa descrição é depois utilizada para animar avatares em situações onde a expressões facial é um requisito, como por exemplo na produção cinematográfica, nos jogos e na língua gestual.

O trabalho de [Lugrin et al. \[2016\]](#) combina vários softwares com o objetivo de produzir avatares. Para o *body tracking* usam o Brekel Kinect e para as expressões faciais usam o Faceshift Studio.

3.3.2 Relação com o PE2LGP

O principal objetivo desta investigação foi perceber quais as capacidades, as limitações e as tendências da utilização do sensor Kinect. Como por exemplo, o software Brekel Kinect, que permitiu saber que é possível, não só fazer um tracking razoável do corpo humano, como também manipular esses dados em diversos softwares em tempo real, como o Unity.

3.4 Rastreamento das Mãos (Hand Tracking)

Nesta seção apresentam-se os projetos recentes relativos aos rastreamento das mãos (*hand tracking*), usando para isso sensores de baixo custo, como o Kinect da Microsoft, o Senz3d da Creative²¹ ou o Leap Motion.

Hand Tracking é atualmente considerado um desafio [[Sharp et al., 2015](#)]. As mãos podem formar um conjunto variado de poses complexas com vários níveis de liberdade [[Jones and Lederman, 2006](#)], têm movimentos rápidos, e podem ainda ter diferentes formas e tamanhos. Os dedos podem ser difíceis de desambiguar e podem estar ocultos por outras partes da mão. Assim sendo, muitos projetos limitam o *hand tracking* para *close-range* (distâncias curtas), ou apenas uma mão, como é o caso dos trabalhos apresentados nos trabalhos de [Sridhar et al. \[2013\]](#) e de [Tagliasacchi et al. \[2015\]](#).

Antes do *hand tracking*, que consiste em detetar a trajetória da mão numa sequência de imagens, é necessário detetar quais os pixels que pertencem à mão para cada *frame*. A este processo dá-se o

¹⁹<http://brekel.com/>

²⁰<http://faceshift.com>

²¹Senz3d é também um sensor de movimentos e profundidades, porém em comparação ao Kinect, possui menos componentes de captura. <http://pt.creative.com/p/web-cameras/creative-senz3d>

nome de *hand segmentation*. Para fazer *hand segmentation*, na maioria dos projetos usa-se sempre a mesma técnica, que consiste em tirar partido da informação de profundidade do sensor, assumindo para isso que a mão é o que está mais próximo do sensor e, portanto, tudo o resto é ignorado.

Existem 3 métodos para fazer *hand tracking*: métodos discriminativos que trabalham diretamente com os dados da imagem (por exemplo extrair características e usar técnicas de classificação), métodos generativos (ou baseados em modelos) que utilizam um modelo 3D de uma mão para recuperar a pose e, por fim, métodos híbridos que são uma combinação dos dois anteriores [Erol et al. \[2007\]](#). De seguida, estes métodos serão explicados com mais detalhe.

3.4.1 Métodos Discriminativos

Os métodos discriminativos usam conjuntos de treino (*training sets*) e técnicas de aprendizagem automática para fazerem um mapeamento direto a poses através das características extraídas das imagens de entrada (input). O trabalho de [Athitsos and Sclaroff \[2003\]](#) e o de [Keskin et al. \[2012\]](#) usam métodos discriminativos. Existe uma base de dados com várias imagens de poses de mãos e tentam depois fazer uma correspondência com base no que é capturado pelo sensor.

3.4.2 Métodos Generativos

[Oikonomidis et al. \[2011\]](#) apresentam um trabalho para o problema do *hand tracking* baseado em métodos generativos (também conhecidos como abordagem baseada em modelos (*model-based approach*)) e que é formulado como um problema de otimização que minimiza a discrepância entre um modelo de uma mão 3D usada como hipótese e o que é observado pelo Kinect. Por outras palavras, existe um modelo com 27 parâmetros que correspondem às articulações dos dedos de uma mão 3D e a captura do Kinect através do sensor de profundidade, o objetivo é estimar os valores dos parâmetros do modelo de modo a ficarem próximos da mão correspondente à realidade.

É pretendido minimizar a função que calcula a discrepância entre o observado pelo Kinect e os mapas de profundidade calculados com base no modelo 3D da mão. A otimização é feita através de um algoritmo estocástico de otimização, o Particle Swarm Optimization (PSO). O princípio base do algoritmo é criar um enxame de partículas (*swarm of particles*) que se move num espaço multidimensional pré-definido à procura do seu objetivo (*goal*), ou seja, a posição no espaço que melhor satisfaz as suas necessidades. Em analogia com a natureza, é o mesmo que um bando de pássaros estar à procura do melhor local para dormir (que pode ser a melhor combinação de várias características, como o tamanho espaço, acesso a comida e água, etc) [[Kennedy and Eberhart, 1995](#)].

O algoritmo tem como base 2 conceitos principais. O primeiro é que uma partícula (que pode ser vista como solução do problema) pode determinar a qualidade da sua posição atual. O segundo consiste num fator estocástico que faz as partículas moverem-se. Portanto, temos um enxame de partículas a moverem-se num espaço em que a posição de cada uma é avaliada de acordo com uma função de fitness. A função de fitness depende do problema a ser otimizado. No trabalho de Oikonomidis o algoritmo PSO opera sobre um espaço de 27 dimensões que correspondem aos

parâmetros das articulações da pose 3D da mão, ou seja, para cada *frame*, temos 27 parâmetros devolvidos pelo algoritmo. No entanto, este trabalho não consegue recuperar da perda do movimento.

3.4.3 Métodos Híbridos

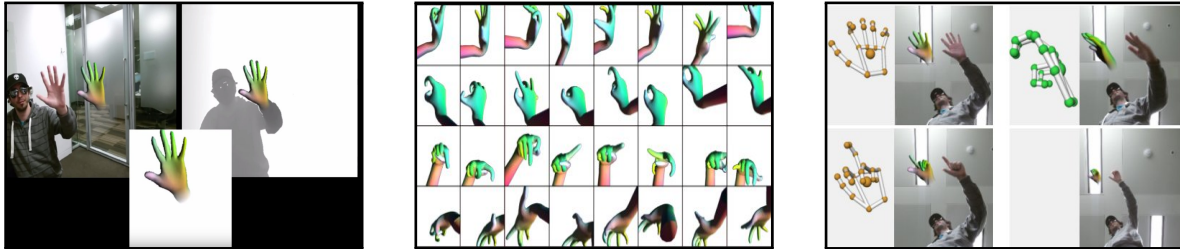


Figura 3.11: Resultado do Trabalho de Sharp et al. [2015].

Os trabalhos mais recentes conhecidos como “Modern pose estimation algorithms” usam abordagens híbridas (por exemplo os trabalhos: Krejov et al. [2015], Sharp et al. [2015] e Sridhar et al. [2013]), também consideradas abordagens baseadas em análise por síntese (analysis by synthesis) em que muitas delas utilizam como base o algoritmo PSO. Primeiro a componente discriminativa faz uma predição direta através da imagem de entrada sobre os parâmetros da pose da mão. Depois, a componente generativa tenta aproximar um modelo 3D da mão com o que é observado através do sensor, em que o principal objetivo é tentar minimizar uma função de energia (energy function). Uma função de energia ideal é o erro de reconstrução, ou seja, a distância entre o observado e a hipótese do modelo.

O trabalho descrito em Sharp et al. [2015] e desenvolvido pela Microsoft apresenta um sistema com uma abordagem híbrida, capaz de resolver algumas restrições descritas anteriormente, como o funcionamento apenas em close-range e sobretudo a perda de movimento. Usam também o algoritmo PSO e uma função de energia, definida pelos autores de “golden energy”. No final permite-lhes obter um modelo 3D da mão o mais próximo possível do que é observado pelo sensor (Figura 3.11).

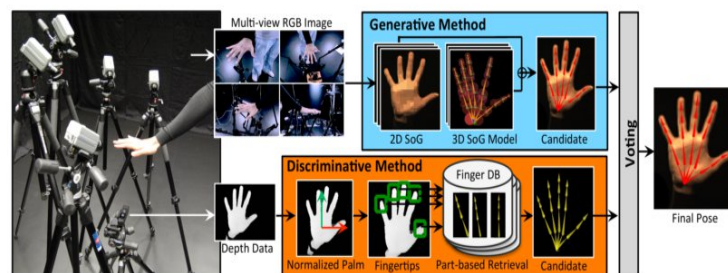


Figura 3.12: Pipeline do Sridhar et al. [2013].

O trabalho descrito em Sridhar et al. [2013] utiliza também uma abordagem híbrida. É seguido um pipeline que faz uso das duas abordagens em paralelo (Figura 3.12). O principal foco deste trabalho é na abordagem discriminativa, que deteta as extremidades dos dedos através da informação de profundidade usando para isso um classificador linear SVM (Support Vector Machine). As extremidades dos dedos são detetadas numa framework de teste-e-hipótese (*hypothesize-and-test*)

que, juntamente com as bases de dados, contém poses de cinco dedos. Assim, obtêm-se múltiplas hipóteses, em que cada uma é testada usando dois critérios. A pose da mão final é a pose que tem o menor erro entre as posições das extremidades dos dedos observadas e estimadas.

A grande contribuição deste trabalho foi o novo método para recuperação de poses em conjunto com imagens base otimizadas da pose. Este método permite recuperar e estabilizar o rastreamento de poses que tenham algumas partes do movimento das mãos oculto.

A grande desvantagem deste trabalho é que utiliza múltiplas câmaras de profundidade tentando assim resolver problemas de oclusão.

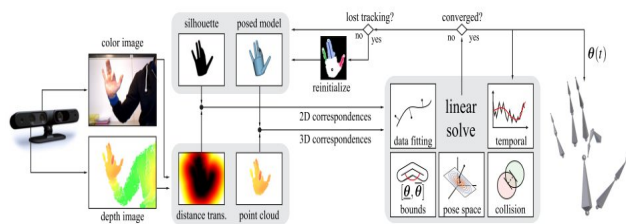


Figura 3.13: Pipeline do Tagliasacchi et al. [2015].



Figura 3.14: Resultado do Tagliasacchi et al. [2015].

O trabalho descrito em Tagliasacchi et al. [2015], ao contrário anteriores, usa um sensor diferente, o Senz3d²² e é focado sobretudo no ICP (Iterative Closest Point) (Figura 3.13). O ICP é um algoritmo de alinhamento, responsável por minimizar a diferença entre conjuntos de pontos de controlo (Figura 3.15). O ICP é geralmente utilizado para reconstruir superfícies 2D e 3D. Os dados de entrada do ICP são dois conjuntos de pontos de controlo (Figura 3.16). Estes conjuntos de pontos podem ser conseguidos através de algoritmos de detecção de contornos²³.

A combinação destas técnicas resulta nas poses ilustradas na Figura 3.14.

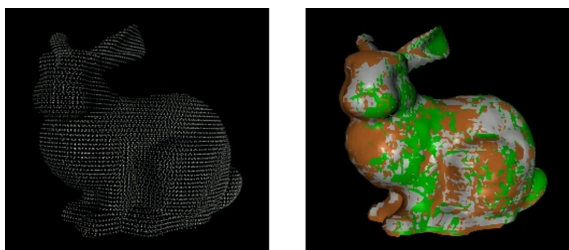


Figura 3.15: Exemplo do Algoritmo ICP.

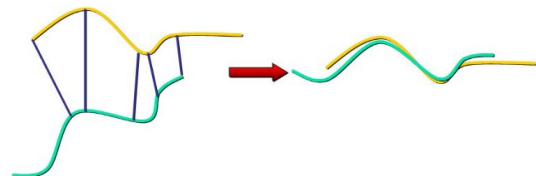


Figura 3.16: Alinhamento do Algoritmo ICP.

As técnicas mencionadas são aquelas utilizadas como base neste tipo de projetos. No entanto, na investigação atual já pouco se usam estas técnicas por si só. O que tem vindo a decorrer ao longo do tempo é a utilização de algoritmos que já tiveram inúmeras intervenções a partir destas técnicas, quebrando e melhorando as suas limitações.

²²<http://pt.creative.com/p/web-cameras/creative-senz3d>

²³<https://github.com/OpenGP/htrack>

3.4.4 Leap Motion

O MotionSavvy²⁴ é um sistema multidireccional desenhado para ajudar pessoas com deficiência auditiva e os ouvintes a comunicarem entre si. Este sistema contém o UNI (Figura 3.17) que é um tablet que incorpora, não só o Leap Motion, como também um software especializado no tracking das mãos. Desta forma, é possível traduzir gestos para texto e vice-versa²⁵.

Neste sentido, surgem diversos projetos com o mesmo intuito, isto é, traduzir gestos para texto. Para isso, fazem reconhecimento através técnicas de Machine Learning. O que difere uns dos outros são as técnicas para reconhecimento, e também a língua. Por exemplo, os trabalhos Kotsidou et al. [2015] e Simos and Nikolaidis [2016] focam-se no reconhecimento da Língua Gestual Grega, sendo que o segundo e mais recente, é apenas focado no alfabeto.

O Leap Motion tem vindo a crescer nos últimos meses, em consequência da enorme adesão e investigação na Realidade Virtual através dos Oculus²⁶. É possível incorporar o dispositivo Leap Motion no set da Oculus (Figura 3.18). Por essa mesma razão, foi lançado em Maio deste ano, a atualização Orion²⁷. O Leap Motion Orion consiste numa atualização ao nível do software focado na Realidade Virtual que melhorou consideravelmente o hand tracking, explorando assim melhor as capacidades do hardware²⁸.



Figura 3.17: Dispositivo UNI da MotionSavvy.



Figura 3.18: Set que incorpora o Leap Motion.

Outros trabalhos estão a ser realizados usando o Leap Motion no contexto da Realidade Virtual, como por exemplo o descrito em Scheggi et al. [2015], que tenta em conjunto da tecnologia háptica dar feedback de força ao utilizador.

3.4.5 Relação ao PE2LGP

Embora estes trabalhos não tenham sido aplicados ao trabalho apresentado, foi uma investigação feita no sentido em perceber como ultrapassar as limitações do Kinect e o Leap Motion, em particular no detalhe da captura das mãos. A investigação foi importante para perceber a complexidade desta tarefa, sendo que é preciso um trabalho dedicado apenas a esta área.

²⁴<http://www.motionsavvy.com/>

²⁵<https://www.youtube.com/watch?v=Kj51RNZAM9k>

²⁶<https://www.oculus.com/>

²⁷<https://developer.leapmotion.com/orion>

²⁸<https://www.youtube.com/watch?v=rnlCGw-0R8g>

3.5 Recursos da Área de Processamento de Língua Natural

Nesta seção pretendo apresentar um dos recursos já usados no PE2LGP, o *NLTK*, e também ferramentas utilizadas para compreender a gramática da LGP. No entanto, esta continua a ser uma área onde a investigação é escassa, dificultando assim a qualidade de aplicações que envolvem recursos linguísticos na LGP, sendo esta a crítica principal das pessoas portadoras de deficiência auditiva.

3.5.1 Natural Language Toolkit

A *NLTK*²⁹ é um conjunto de bibliotecas e programas para a linguagem de programação Python e que tem como objetivo fazer o processamento simbólico e estatístico da Língua Natural. O *NLTK* possui ainda demonstrações gráficas e exemplos de dados.

3.5.2 A Gramática da LGP

Os recursos linguísticos na LGP são limitados. A única gramática existente é de Amaral et al. [1994] e não está acessível. O único estudo recente neste contexto é de Maria Fernanda da Silva Bettencourt, no seu trabalho intitulado “A ordem das palavras na Língua Gestual Portuguesa: Breve estudo comparativo com o Português e outras línguas gestuais” [da Silva Bettencourt, 2015], que teve como principal objetivo perceber se a LGP possuía um padrão de ordem básica das constituintes da frase e, a confirmar-se, qual seria.

Os resultados foram obtidos através de um pequeno corpus que não permite generalizações. Porém, os dados demonstram que a LGP tem uma estrutura sintática própria e que o padrão de ordem básica dos seus constituintes é SVO. No livro de Amaral et al. [1994], a autora não afirma qual é a ordem que se deve utilizar, daí a inconsistência. Para agravar, é dito no livro que as seguintes frases com diferentes regras são válidas:

1. Bolo comer rapaz (regra OVS).
2. Comer rapaz bolo (regra VSO).
3. Bolo rapaz comer (regra OSV).

O trabalho conclui que se pode aplicar à LGP, tal como já se aplica a outras línguas gestuais, princípios sintáticos semelhantes aos das línguas orais. Para chegar a tal conclusão, o trabalho focou-se em considerar 3 hipóteses:

1. A ordem dos constituintes da frase varia consoante o tipo de verbos.
2. A compreensão e a produção são marcadas pela ordem dos constituintes.
3. A concordância do verbo com o objeto, influência a ordem dos constituintes.

²⁹<http://www.nltk.org/>

Os dados foram obtidos a partir de informantes com diferentes perfis linguísticos (6 informantes surdos e 2 informantes ouvintes intérpretes de LGP), usando 2 tipos de estímulos para a eliciação de produções verbais em LGP. A primeira referente a imagens com situações diversas. Na segunda foram apresentadas frases com verbos a cada um dos informantes e foi pedido que produzissem em LGP as frases que considerassem corresponder-lhes.

A partir da primeira hipótese (Item 1), verificou-se que há alguns tipos semânticos de verbos que influenciam a ordem dos constituintes da frase. A segunda hipótese (Item 2) ficou confirmada. Para concluir, não foi possível encontrar evidências que comprovem que a concordância do verbo com o objeto possa influenciar a ordem (Item 3). A autora afirma que:

“Quando a ordem básica SVO não é a ordem dos constituintes de uma frase, é necessário usar corretamente mecanismos de concordância (por exemplo MNM³⁰) de modo a não comprometer a inteligibilidade da frase.”

Através da verificação destas 3 hipóteses, pode-se confirmar o que tem sido escrito sobre outras línguas gestuais, como por exemplo, o defendido por Liddell [2003], isto é, que a língua gestual é uma língua sem preposições.

3.5.3 Dicionário de Língua Gestual Portuguesa

O Dicionário de Língua Gestual Portuguesa representa, tanto do ponto de vista social, como cultural, um marco na valorização desta forma de expressão que é, desde 1997, uma das línguas oficiais de Portugal. Foi organizado especificamente para o contexto português por Ana Bela Baltazar, especialista em língua gestual, que exerce funções de intérprete na Rádio Televisão Portuguesa e na Associação de Surdos do Porto e é presidente do Centro de Tradutores e Intérpretes de Língua Gestual, para além de ter a seu cargo a docência de várias disciplinas do âmbito da Língua Gestual³¹.

Salientado que o trabalho de Almeida [2014] tinha este dicionário como ponto de partida.

3.5.4 Relação ao PE2LGP

Esta investigação, serviu para perceber que a investigação nesta área continua sem progressos, pelo que, os recursos utilizados foram os mesmos já teriam sido utilizados no início do PE2LGP. Pela mesma razão, o trabalho apresentado também não progrediu neste contexto.

O dicionário foi utilizado como base para consulta na criação dos gestos através dos sistemas apresentados nos capítulos Capítulo 5 e Capítulo 6. Além disso, também se manteve o uso da NTLK na componente linguística.

³⁰Marcações Não Manuais - Expressão e movimento corporal.

³¹<http://www.portoeditora.pt/produtos/ficha/dicionario-de-lingua-gestual-portuguesa?id=3501376>

4

Sistema 1: Modificação da Arquitetura

Conteúdo

4.1	Exportação do Modelo para o Unity	41
4.2	Componente de Língua Natural	46

Neste capítulo é apresentada a primeira tarefa realizada neste trabalho que surge em virtude da dependência do PE2LGP ao Blender. Tal como se pode verificar através da [Figura 4.1](#), todo o projeto original estava integrado no Blender, inclusive a componente de Língua Natural. De modo a ultrapassar essa dependência, foram feitas duas principais alterações: a exportação do projeto para o motor de jogo Unity ([Seção 4.1](#)) e a criação de um modelo independente de processamento de língua natural ([Seção 4.2](#)).

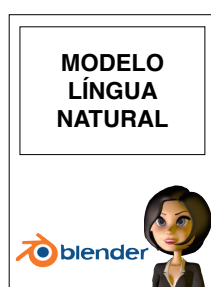


Figura 4.1: Arquitetura do PE2LGP Original.

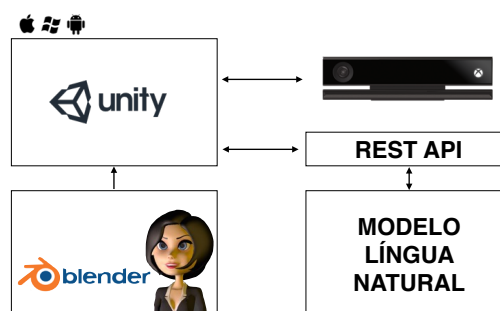


Figura 4.2: Arquitetura do PE2LGP Atual.

Como se pode verificar pela [Figura 4.2](#) ambas as alterações são independentes do Blender, embora comuniquem entre si. Além disso, foi introduzido um sensor, o Kinect, que parte do sistema apresentado no [Capítulo 6](#).

A componente de língua natural pode ser utilizada como sistema único e independente, ou ser integrada noutro sistema, como é o caso.

A escolha da introdução de um motor de jogo surge da necessidade de escalar o projeto. Optar por um motor de jogo permite exportar o projeto para outro formato e, assim, eliminar a dependência ao Blender. Optei pelo Unity uma vez que é o mais simples e flexível, em comparação com outros, no que diz respeito ao escalonamento.

4.1 Exportação do Modelo para o Unity

A exportação pode ser resumida em 2 tarefas: exportar o modelo do Blender para o Unity (incluindo 69 animações) e criar uma interface que permita a interação com o modelo. Contudo, a primeira tarefa revelou-se mais complicada do que o previsto. Em seguida, explico os problemas encontrados e como foram resolvidos.

4.1.1 Problemas Encontrados e Soluções

Muitos modelos criados a partir de diversos programas de modelação e animação 3D, tais como, o Blender ou o Autodesk 3DS MAX, são importados para o Unity sem quaisquer problemas. O Unity internamente resolve possíveis problemas e, quando não consegue, dá ao utilizador hipóteses de escolha para os tentar resolver por si. No entanto, este processo pode não ser tão linear, tal como foi o caso da importação do modelo da Catarina.

4.1.1.A Tempo de Importação Excessivo

A exportação do modelo para o Unity, geralmente é feita de uma de duas formas: ou é gerado um ficheiro no formato .FBX¹ através do programa de modelação ou é usado o próprio ficheiro fonte, que neste caso foi um ficheiro com a extensão .blend vindo do Blender. Em ambos os casos, o tempo de importação era excessivo, isto é, mais de 15 minutos, quando em casos normais dura apenas uns segundos. Salientando que, em alguns casos, o Unity ficava inativo por cerca de 15 minutos, sendo esse o tempo de importação.

O problema era devido a uma componente que o modelo continha e que é gerada através do *Rigify*², que é uma extensão do Blender utilizado apenas como base para a criação do *rig*. Assim que o *rig* está concluído, por norma, essa componente é apagada, por ser excessivamente pesada em termos computacionais. Portanto, bastou eliminar essa componente.

4.1.1.B Mapeamento das Texturas

O mapeamento das texturas e dos materiais também é feito de forma automática pelo Unity. Contudo, podem ocorrer problemas tal como se pode verificar na [Figura 4.3](#) e [Figura 4.4](#). Isto pode acontecer porque: ou o Unity não consegue mapear as texturas no modelo de forma automática ou o ficheiro fonte não contém essas texturas. Neste caso, com o modelo do PE2LGP, aconteceram ambas as situações. Para resolver foi preciso mapear manualmente as texturas que não teriam sido mapeadas corretamente, importar as que não estavam contidas no ficheiro fonte e, por fim, criar as restantes internamente no Unity. Resultando assim no modelo da [Figura 4.5](#).

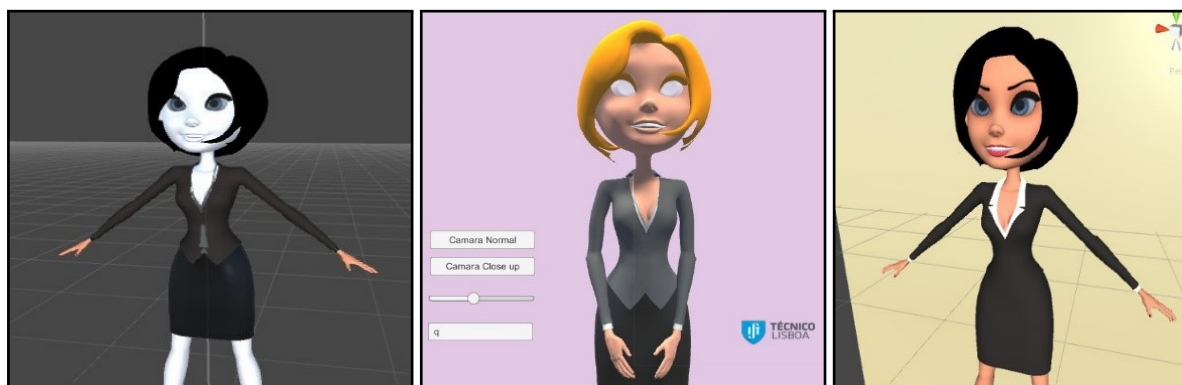


Figura 4.3: Mapeamento Incorreto 1. **Figura 4.4:** Mapeamento Incorreto 2. **Figura 4.5:** Resultado Final.

4.1.1.C Mapeamento do Rig no Sistema de Músculos do Mecanim

O Unity dispõe de um sistema de animações baseado em máquinas de estados chamado Mecanim³. O Mecanim utiliza avatares no formato *Humanoid* para as suas animações. O que difere um avatar *Humanoid* para um de outro tipo, é sobretudo o fato de animações criadas neste formato poderem ser utilizadas num *rig* diferente. Por outras palavras, uma animação criada no Mecanim usando

¹FBX é um formato de ficheiros 3D que, entre várias outras informações, contém os valores das transformações do modelo.

²<https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Rigging/Rigify>

³<http://docs.unity3d.com/460/Documentation/Manual/MecanimAnimationSystem.html>

um modelo X com avatar *Humanoid*, poderá ser utilizada em qualquer outro modelo Y também com avatar *Humanoid*.

Portanto, já que seria este o sistema a utilizar, sempre foi principal preocupação converter o modelo do PE2LGP para um avatar *Humanoid*. Este processo é relativamente simples, bastando mudar as definições no Unity para que o avatar seja *Humanoid*. Depois, internamente, o Unity tenta mapear a estrutura dos *bones* à estrutura do *Humanoid*. Este, por sua vez, utiliza músculos em vez de *bones*. Este processo, mais uma vez, geralmente ocorre corretamente. No entanto, esse não foi o caso, e quando isso não acontece, podemos verificar, através de uma interface, quais as partes do *rig* que não foram mapeadas (Figura 4.6).

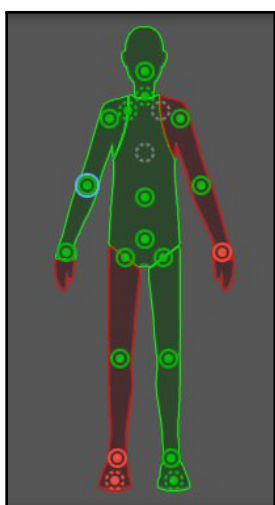


Figura 4.6: Mapeamento Incorreto do *Rig*.

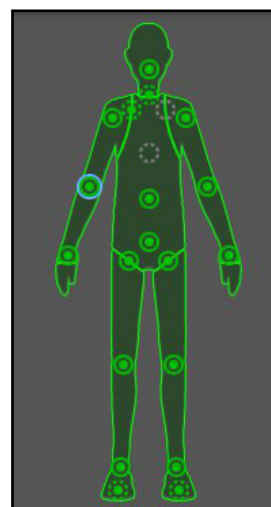


Figura 4.7: Mapeamento Correto do *Rig*.

Este problema acontece quando o Unity não consegue mapear automaticamente o *Rig* nos músculos, devido à estrutura do modelo não ser a esperada. Para resolver o problema, geralmente, o mapeamento terá de ser feito manualmente. No entanto, no caso do modelo no PE2LGP não foi suficiente.

Basicamente, a solução consistiu em corrigir a estrutura do *rig* criada através da extensão *Rigify* e, assim, poder ser aceite no Unity. Esta correção é feita através de um script⁴ escrito em Python, executado internamente no Blender. Deste modo, quando a estrutura é correta e aceite pelo Unity o *feedback* obtido pela interface é semelhante ao da Figura 4.7. Isto significa que o sistema Mecanim está funcional no modelo em questão.

4.1.1.D Importação das Animações

O Unity, por omissão, uma vez mais, tenta fazer a importação automaticamente, neste caso das animações. Porém, não é suficiente. O processo geralmente ocorre da seguinte maneira: encontra todas as animações e junta-as numa só. Depois, é necessário separá-las manualmente. No entanto, no caso do Blender, as animações têm de estar na *timeline*. Posto isto, é necessário anotar o tempo inicial e o tempo final de cada animação definidos no Blender, para que seja possível utilizar esses

⁴Disponível em: <https://goo.gl/f5krh5> ou <https://goo.gl/CA4T4W>

Ultrapassado este obstáculo, foram importadas, através do Blender usando o PE2LGP original, 69 animações para o Unity.

Tal como foi dito anteriormente, o sistema de animações do Unity, Mecanim, é baseado em estados. Cada ação disponível no ambiente é considerado um estado, que, no fundo, representa uma animação. Porém, na maioria dos casos este sistema é aplicado em ambientes estáticos. Por exemplo, num ambiente de um jogo, as ações poderão ser: correr, andar e saltar. No entanto, neste projeto, as ações não são estáticas, pois variam de acordo com o input introduzido. Considerando o input “João”, este deverá resultar em 5 ações possíveis: “j”, “o”, “a” e “o”. Cada ação representa uma animação e é ligada a outra por meio de uma transição.

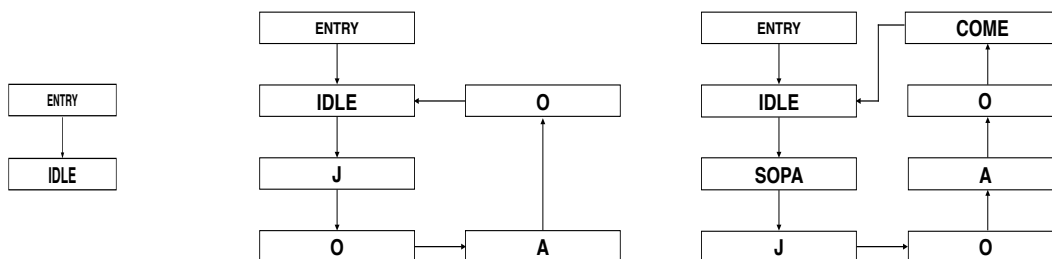


Figura 4.8: Estados Base. **Figura 4.9:** Estados do Input: “João”. **Figura 4.10:** Estados do Input Exemplo.

O sistema de criação de estados baseia-se sobretudo em duas etapas. Primeiro é analisado e processado o input de acordo com as diferentes fases apresentadas na [Seção 4.2](#). Depois, com base no input processado, são criados os estados tendo em atenção dois fatores: se a palavra é *fingerspelled* ou não. De salientar que, quando nenhuma animação está ativada, o sistema apenas executa repetidamente o estado *IDLE*, que corresponde a um estado base, ou seja, quando nada está a ser feito ([Figura 4.8](#)). Para o input “João” teríamos o conjunto de estados da figura [Figura 4.9](#) e para “O João come a sopa” o da figura [Figura 4.10](#). Quando o sistema acaba de passar por todas as transições e estados voltando ao estado base, o *IDLE*, os restantes estados são apagados, ficando apenas na representação da [Figura 4.8](#).

Uma vez importado o modelo e as animações, e criado o sistema de criação de estados, era altura de criar uma interface com a qual o utilizador pudesse interagir com todas as componentes descritas nas secções anteriores. O objectivo era criar uma interface semelhante à existente no protótipo do

sistema original do PE2LGP (Figura 4.11), porém com mais opções, uma vez que só era possível introduzir o texto e obter o seu significado em LGP.

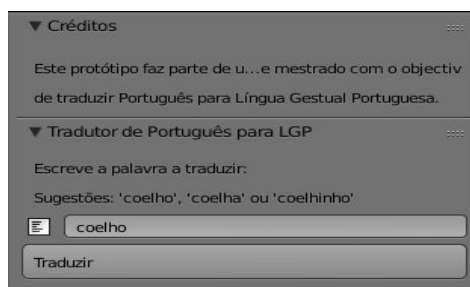


Figura 4.11: Interface do PE2LGP Original.

A interface foi criada recorrendo ao sistema *UI* do Unity⁵, nomeadamente o *Canvas* que é onde os elementos da UI devem estar inseridos (Figura 4.12).

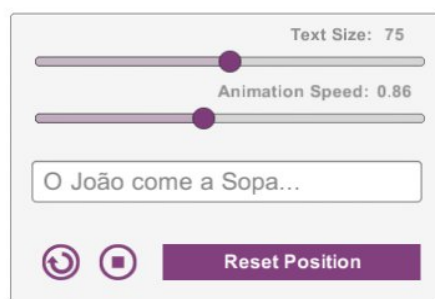


Figura 4.12: Interface do PE2LGP Atual.

Na interface atual o utilizador pode:

1. Escrever uma palavra ou frase e obter o seu significado em LGP.
2. Controlar a velocidade dos gestos.
3. Ver em simultâneo da execução do gesto a sua transcrição.
4. Alterar o tamanho do texto que corresponde à transcrição.
5. Parar a qualquer momento o que estiver a ser executado.
6. Repetir até que mande parar o *input* atual.
7. Movimentar o modelo em torno do seu próprio eixo X.
8. Fazer zoom-in ou zoom-out ao modelo.

⁵<http://docs.unity3d.com/Manual/UISystem.html>

4.2 Componente de Língua Natural

Anteriormente no PE2LGP, o desenvolvimento da componente de processamento de texto era feita através do Blender recorrendo à linguagem Python. Portanto, se quiséssemos modificar essa componente seria necessário, obrigatoriamente, passar pelo Blender. De modo a eliminar essa dependência, foi criada uma REST API (Subseção 4.2.2) escrita em Python que processa pedidos HTTP. Esses pedidos levam como argumento uma *string* que é analisada pela componente de processamento de texto (Subseção 4.2.1), que entretanto também teve modificações relativamente à original. Cada resposta a um pedido HTTP resulta num ficheiro JSON (Listing 4.1) que segue uma estrutura previamente estabelecida com a *string* processada.

4.2.1 Componente de Processamento de Texto

Na arquitetura anterior, o processamento do texto era feito através de um pipeline (Figura 4.13).

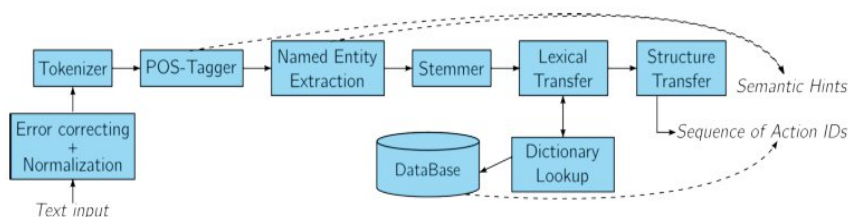


Figura 4.13: Pipeline do PE2LGP Original (Figura do Trabalho de Almeida [2014]).

Agora, o resultado produzido é semelhante, porém, feito recorrendo a vários métodos de duas classes. As principais vantagens desta modificação são: o fato de termos total controlo de cada uma das fases do pipeline e também ter resolvido alguns problemas de comunicação entre a interface e esta componente, como por exemplo, falhas de *output*. Basicamente, o pipeline está implementando recorrendo aos métodos da classe *Phrase* e *Word* (Figura 4.14).



Figura 4.14: Fluxo da Estrutura das Classes da Componente Linguística.

Desta forma, e para facilitar a compreensão, em seguida explica-se o funcionamento de toda a componente recorrendo a um exemplo como entrada: “O João come a sopa”.

Em primeiro lugar, a frase é dividida em várias palavras, em que cada uma corresponde a um objeto e todas estão inseridas numa lista. Portanto, para o exemplo teríamos a seguinte lista com 5 elementos, uma vez que a frase também tem 5 palavras:

[Objecto1, Objecto2, Objecto3, Objecto4, Object5]

Para cada elemento da lista que corresponde a um objecto pode-se usar métodos, como por exemplo, obter o nome da palavra original. No caso do *Object1* o nome original seria a palavra “O”.

O próximo passo consiste em analisar cada palavra da lista de objetos e, por fim, classificá-las. Começando pela limpeza e à normalização dos dados é necessário:

1. Remover os acentos e pontuações.
2. Colocar tudo numa única forma, neste caso, em letras minúsculas.
3. Analisar as terminações de cada palavra de modo a saber se é necessário adicionar uma nova palavra, por exemplo, algo acabado em “inha” remete para “pequeno” e “feminino” na LGP, portanto, neste caso, será necessário adicionar à lista mais duas palavras.
4. Remover as *stops_words* da frase, sobretudo artigos, visto não serem usados na LGP.

Para o exemplo considerado, a lista ficaria reduzida a 3 elementos, pois iriam ser removidos 2 artigos, o “o” e o “a”, resultando assim na seguinte representação:

[Objecto1, Objecto2, Objecto3]

Se ignorarmos a representação por objetos e trocarmos pelo seu valor, teríamos:

[joao, come, sopa]

Até aqui, estes passos podem ser definidos como: normalização e correção de erros. Para finalizar este processo, a lista anterior será classificada, resultando em:

[(joao, n), (come, v-inf), (sopa, n)].

O passo seguinte consiste em alterar a ordem da lista para SVO, visto ser a utilizada na LGP [Almeida et al., 2015a]:

[(sopa, n), (joao, n), (come, v-inf)].

Para concluir, e por questões de eficiência do sistema de criação de estados (Subseção 4.1.2), adiciona-se ao fim de cada elemento da lista os *stemmers* de cada palavra:

```
[ (sopa, n, sop), (joao, n, joa), (come, v-inf, com) ].
```

A classe principal tem 3 objetivos: configurar o servidor *REST*, para cada pedido à *REST* recorrer ao processo descrito anteriormente acedendo à lista e, por fim, converter a lista para o formato JSON. Basicamente, a lista JSON, corresponde a converter o resultado da última fase do pipeline para uma lista de dicionários (Listing 4.1), em que a chave é uma breve descrição que servirá depois para converter o JSON para a lista original de objetos utilizada no Unity.

4.2.2 REST API

A escolha do Python para a criação da *REST API*, deve-se ao facto da componente também ser na mesma linguagem. Desta forma, optei pela *framework* Flask⁶, que é definida como sendo uma *microframework* para Python baseada em *Werkzeug*⁷ e *Jinja2*⁸.

Antes de mais, *REST* é uma abstração da arquitetura da web que ignora os detalhes de como foi feita a implementação das componentes e a sintaxe de protocolo, com o objetivo de se focar nos papéis dos componentes, nas restrições sobre sua interação com outros componentes e na sua interpretação de elementos de dados significantes. É aplicado sobretudo a serviços web fornecendo APIs para acesso a um serviço qualquer na web. Para comunicar utiliza mensagens HTTP. É também uma maneira fácil de ligar várias linguagens de programação.

O funcionamento da *REST API* é simples: recebe um pedido HTTP com um argumento em formato *string*, que pode ser uma letra, uma palavra ou uma frase, e devolve o resultado num ficheiro JSON. A estrutura do JSON foi escolhida de acordo com a necessidade e tem 3 atributos por objecto: “name”, “stemmer” e “w.class”. Estes atributos resultam da aplicação das várias fases do processamento de texto descritas na seção anterior (Seção 4.2). O atributo “name” remete para a palavra tratada final, “stemmer” corresponde ao *stem* da palavra e, por fim, a “w.class” tem a classificação dada pelo classificador à palavra contida em “name”.

Portanto, supondo que a *REST API* se encontra localmente em execução, podemos utilizar o seguinte modelo de endereço para efetuar os pedidos: `http://127.0.0.1:10001/pe2lgp?phrase=%s`, onde “%s” é substituído pela palavra ou frase pretendida. Além disso, o resultado está ordenado pelo “name”, através da regra SVO que, posteriormente serve de entrada no sistema de criação de estados (Subseção 4.1.2), sendo possível descartar a preocupação em ordenar os dados no Unity.

Por exemplo, a frase “O João come a sopa”, poderia ser convertida para um pedido à *REST API*, utilizando a seguinte notação:

```
http://127.0.0.1:10001/pe2lgp?phrase=O João come a sopa
```

O resultado devolvido pela *REST* seria o seguinte JSON:

⁶<http://flask.pocoo.org/>

⁷<http://werkzeug.pocoo.org/>

⁸<http://jinja.pocoo.org/>

Listing 4.1: Resultado em Formato JSON do Input “O João come a sopa” pela Componente de Língua Natural.

```
1 {  
2   {  
3     "name": "sopa", "stemmer": "sop", "w_class": "n"  
4   },  
5   {  
6     "name": "joao", "stemmer": "joa", "w_class": "n"  
7   },  
8   {  
9     "name": "come", "stemmer": "com", "w_class": "v-inf"  
10  }  
11 }
```

Desta feita, bastou adaptar a *REST API* à componente de processamento do texto que, entretanto, foi importada do Blender e melhorada (Subseção 4.2.1).

4.2.3 Ligação à Interface

O conceito desta ligação passa por enviar pedidos à componente linguística e tratar devidamente os resultados retornados, ou seja, esta etapa corresponde à transição da *REST API* ao Unity visível na representação da arquitetura apresentada anteriormente (Figura 4.2).

Tal como se pode verificar pela Figura 4.15 o fluxo começa primeiro com pedidos feitos via HTTP à *REST API* recorrendo à interface criada no Unity. A *REST API*, por sua vez, trata o pedido e retorna o resultado no formato JSON novamente para o Unity. Quando o Unity obtém o JSON, a lista de objetos do JSON é convertida num outro formato, isto é, numa lista de objetos de uma classe que contém os mesmos atributos da estrutura do JSON: *name*, *stemmer* e *w_class*.

Os objetos resultantes dessa conversão servem de entrada ao sistema de criação de estados apresentado na Subseção 4.1.2 que, por sua vez, converte cada um dos objetos em estados.

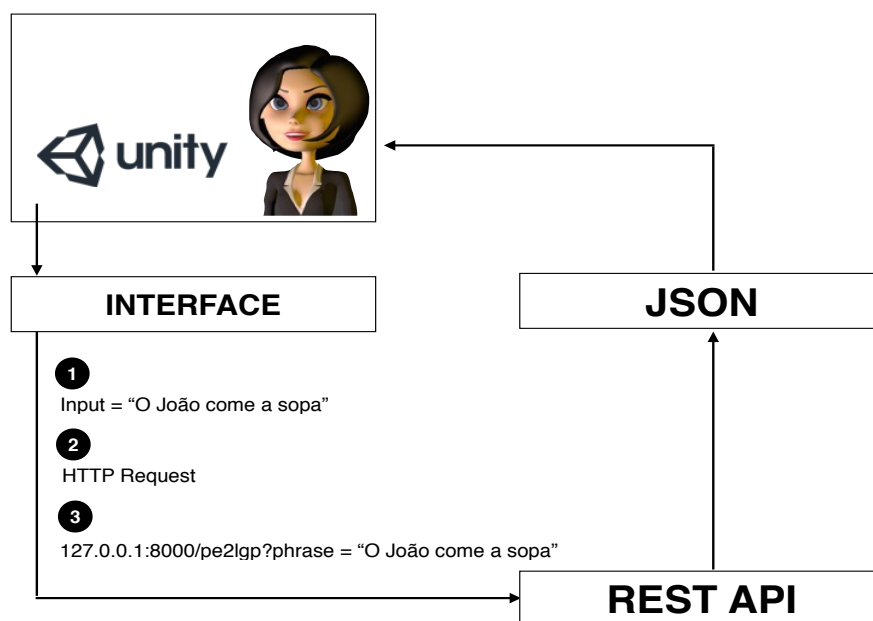


Figura 4.15: Funcionamento da Ligação entre a Componente de Língua Natural e a Interface.

4.2.4 Alocação dos Dados

Sempre foi preocupação tentar não manter os projetos locais, uma vez que acabam por ficar isolados e, por consequência, perdem-se ao longo do tempo. Deste modo, surgiu a necessidade de colocar a *REST API* num servidor. Desta forma, foi alojada num servidor do INESC-ID e é executada através de uma máquina virtual com o sistema operativo Ubuntu⁹. A comunicação ao servidor é feita via SSH, mais propriamente usando Tunnels. Um SSH¹⁰ Tunnel consiste num túnel encriptado criado através do protocolo SSH que pode ser utilizado para transferir o tráfego não encriptado sobre uma rede através de um canal encriptado. Por exemplo, pode-se usar um túnel SSH para transferir ficheiros de uma forma segura entre um servidor FTP¹¹ e um cliente, mesmo que o FTP não seja um protocolo encriptado.

Por outras palavras, o tunnel abre um socket numa determinada porta e todos os pedidos para essa porta irão ser convertidos em pedidos para o servidor. Portanto, um pedido normal à REST API usando o endereço e a notação 127.0.0.1:10001/pe2lgp?phrase="O João come a sopa." seria convertido num pedido SSH ao INESC-ID (Figura 4.16).

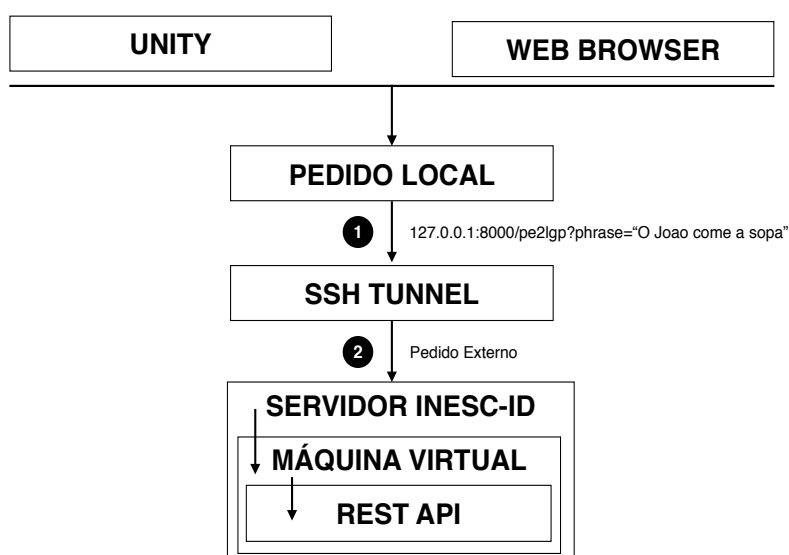


Figura 4.16: Fluxo do Funcionamento dos Pedidos à REST.

⁹<http://www.ubuntu.com/>

¹⁰https://pt.wikipedia.org/wiki/Secure_Shell

¹¹https://pt.wikipedia.org/wiki/File_Transfer_Protocol

5

Sistema 2 - Criação Manual de Gestos

Conteúdo

5.1 Criação do Ficheiro de Animação	52
5.2 Seleção da Configuração das Mãos	52
5.3 Criação da Animação	53
5.4 Plugin para Conversão de Animações	57

A premissa deste projeto sempre foi agilizar o modo de introdução de novos gestos. Desta forma, foram criados dois sistemas: um manual e outro automático.

O sistema manual, denominado de sistema 2, consiste num sistema que, ao contrário do automático apresentado no [Capítulo 6](#), requer a intervenção do utilizador, mas de uma forma menos profissional do que seria esperado caso utilizasse um programação de animação 3D. Através de uma interface, o utilizador pode manipular diretamente o avatar e, assim, criar animações. Assim sendo, conhecimentos na área de animação 3D para o uso deste sistema não são um pré-requisito.

O sistema 2 criado no motor de jogo Unity, produz gestos através de algumas interações com o modelo e uma interface. Desta forma, este capítulo tem como objetivo apresentar o sistema 2.

5.1 Criação do Ficheiro de Animação

Sendo este um sistema de criação de animações, a primeira tarefa passa pela criação do próprio ficheiro de animação. Para isso, o primeiro passo é escolher o nome da animação que pode ser introduzido numa interface ([Figura 5.1](#)). Depois, o nome escolhido será analisado recorrendo a dois métodos: expressões regulares e repetição. O primeiro garante que o utilizador introduz um nome válido, isto é, sem caracteres inválidos como, por exemplo: espaços, vírgulas, pontos, entre outros. E o segundo serve para evitar a criação de um gesto repetido, isto é, cujo o nome já exista.

A interface de criação de ficheiro de animação é apresentada num formato de janela com uma barra de título cinzenta no topo. No canto superior direito da barra de título, há um ícone de casa dentro de um círculo roxo. O corpo principal da interface tem um fundo cinzento claro. No centro, há uma caixa de texto com o texto "asdh1<1,...,a.s" e uma barra de rolagem. Abaixo da caixa de texto, há um botão roxo com o texto "Continuar" em branco. No canto superior esquerdo da interface, há um texto "Por favor, escreva o nome da animação:" em preto, seguido por uma mensagem de erro em vermelho: "O nome da animação contém caracteres inválidos!". No canto inferior direito, há uma assinatura "EP2LGP" em roxo.

Figura 5.1: Interface Relativa à Criação do Ficheiro de Animação.

A interface contém ainda um sistema de *feedback* relativo a erros no caso da introdução de um nome inválido, tal como se pode verificar pela [Figura 5.1](#) a mensagem a vermelho.

5.2 Seleção da Configuração das Mãos

O passo seguinte passa por dar ao utilizador a hipótese de escolher a configuração de ambas as mãos que pretende utilizar para a criação da animação. Desta forma, uma vez mais, o utiliza-

dor dispõe de uma interface (Figura 5.2), onde poderá escolher uma no total de 54 configurações diferentes.



Figura 5.2: Interface Relativa à Seleção da Configuração das Mãos.

A interface Figura 5.2 é o primeiro de dois ecrãs. O segundo é idêntico, mas relativo à escolha da configuração da mão esquerda.

As configurações disponíveis surgem em virtude das já existentes no PE2LGP original, onde posteriormente foram importadas para o Unity (Subsubseção 4.1.1.D). Por sua vez no Unity, através de uma animação, é possível aceder a todos os *frames* da mesma, nomeadamente a qualquer parte do *rig*, sendo possível verificar os valores da posição e rotação por tempo/frame. Desta forma, foi possível aceder apenas aos valores das mãos e assim aplicá-los em tempo real, na altura da seleção, ao modelo.

Contudo, as animações criadas no PE2LGP eram todas para a mão direita e para ter as mesmas animações disponíveis para a mão esquerda foi preciso duplicar todas as animações e ativar, para cada uma, a opção de configuração de animações do Unity, o *mirror*. Como o próprio nome sugere, a opção *mirror* espelha a animação e, portanto, através deste método, as animações da mão esquerda passariam a estar na mão direita.

Depois de escolhido o nome da animação e as respetivas configurações, o utilizador poderá manipular o modelo no terceiro ecrã.

Salientando que este não é o único momento onde é possível escolher a configuração das mãos. A qualquer momento, o utilizador tem hipótese de voltar atrás e mudar.

5.3 Criação da Animação

A premissa deste sistema é o utilizador movimentar o modelo e, a partir desses movimentos, criar animações, *frame* por *frame*. Por outras palavras, a criação da animação corresponde a preencher o ficheiro de animação (Seção 5.1) com *frames*.

Sempre foi uma preocupação questões de usabilidade, tais como: que parte do corpo estaria a ser movimentada, para que direção, e se esses movimentos seriam fáceis de realizar na perspetiva

do utilizador.

5.3.1 Feedback de Seleção

Foram definidos 8 pontos (Figura 5.3), através dos quais o utilizador pode movimentar o modelo. Os pontos são: as mãos, os cotovelos, os ombros, o pescoço e a barriga.

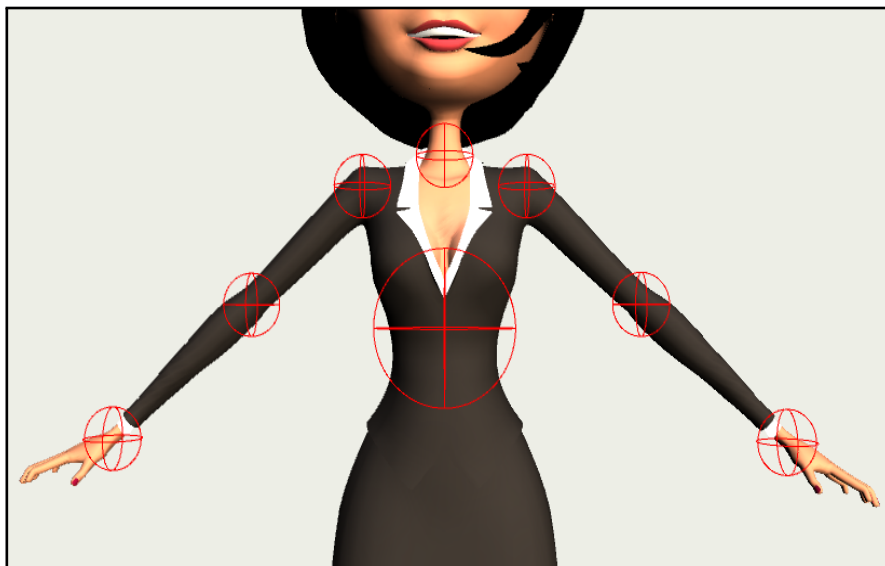


Figura 5.3: Representação dos Pontos Onde é Possível Movimentar o Modelo.

A preocupação na usabilidade começa precisamente depois de definidos estes pontos. Era necessário indicar ao utilizador qual dos pontos estava a ser selecionado. Para isso foram usados os *Gizmos* do Unity, que são desenhados sob modelo, tal como se pode verificar através da Figura 5.4, onde está selecionado o ombro do lado esquerdo do modelo. Os *Gizmos* são utilizados sobretudo como guias visuais para *Debugging* em na *Scene View*, ou seja, no ecrã de criação disponível ao desenvolvedor. Estes guias também podem ser ativados na *Game View* que corresponde ao ecrã onde a câmara é renderizada e que foi o caso.

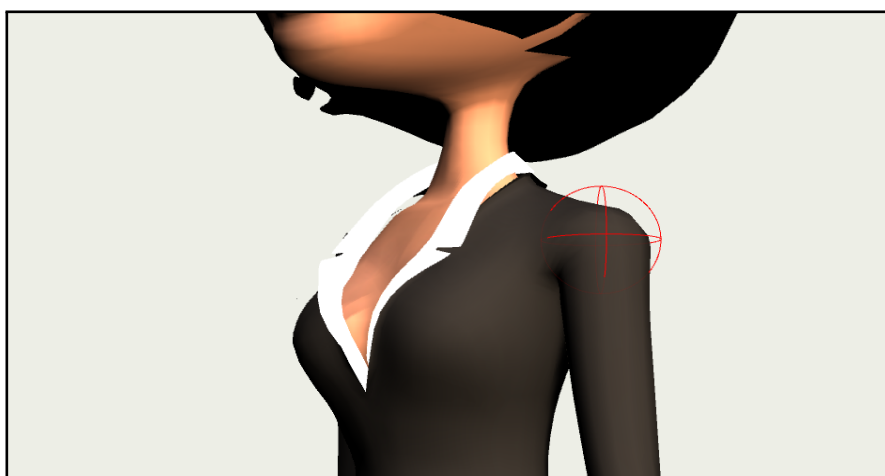


Figura 5.4: Gizmo na Zona Seleccionada do Modelo (Ombro Esquerdo).

5.3.2 Feedback e Limitação da Rotação

Até aqui as informações que o utilizador tinha acerca do ponto do modelo seleccionado era insuficientes. De acordo com este sistema de criação de animações, o utilizador poderia movimentar cada ponto segundo o sistema de coordenadas cartesiano em *World Space* e usando Quaterniões¹. Por isso, o utilizador teria em qualquer altura que tivesse seleccionado um ponto, a hipótese de movimentar para X,Y ou Z. Mas, uma vez usados quaterniões, o sistema de coordenadas está constantemente em movimento, logo não se podia pedir ao utilizador que se seguisse pelo tradicional sistema cartesiano, que é estático. Por essa razão surge a necessidade de dar *feedback* da rotação. Desta forma, foi criado um objecto 3D e aplicados vários materiais, isto de modo a simular as cores representativas dos eixos e associa-los a cada um dos pontos seleccionados, dando assim, em tempo real, *feedback* ao utilizador para onde poderia movimentar o ponto seleccionado. O objecto foi criado no programa de modelação Blender, e consiste na modificação de uma das *meshes* que o Blender disponibiliza por omissão, neste caso a *Torus* (Figura 5.5).

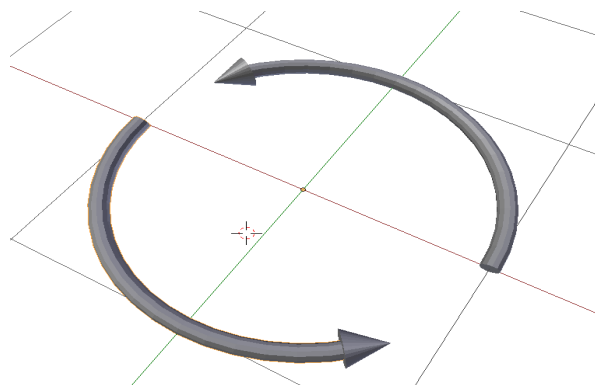


Figura 5.5: Objecto Utilizado para Feedback.

Os materiais criados e aplicados utilizam as cores sólidas: verde, vermelho e azul. Sendo o vermelho representativo do eixo X, o verde do eixo Y e o azul do eixo Z. Assim, o utilizador ao carregar, por exemplo, no ombro direito teria o feedback de seleção e também de direção da rotação por omissão o Y. Contudo, é possível, a qualquer momento, modificar o eixo e obter *feedbacks* diferentes em tempo real, tal como se pode verificar através da Figura 5.6.

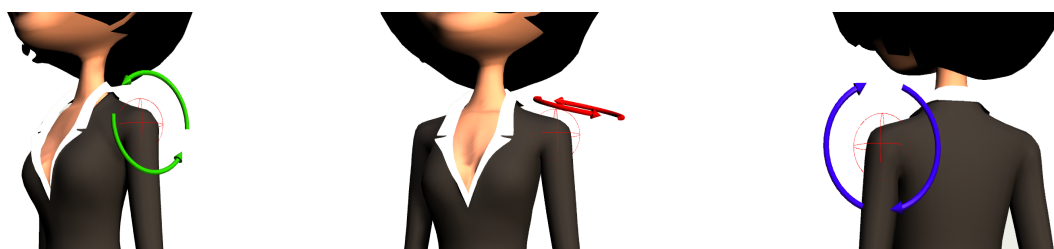


Figura 5.6: Feedback de Rotação do Ponto Seleccionado.

Ao seleccionar um dos pontos, é verificado o valor do centro e instanciado na mesma direção, um dos 3 modelos da Figura 5.6. Além disso, estes objetos instanciados não são estáticos, contêm uma

¹<https://pt.wikipedia.org/wiki/Quatern%C3%B5es>

animação, isto é, permanecem em rotação em torno do seu próprio eixo X.

O utilizador, para movimentar qualquer um dos pontos, tem de manter pressionado o botão esquerdo do rato no local selecionado e arrastá-lo em movimentos horizontais.

Neste altura, o utilizador deveria saber o que estava selecionado e para onde estava a rodar. No entanto, havia que limitar os movimentos. A ideia seria tentar duplicar a rotação do corpo humano. Assim, não deve ser possível, por exemplo, fazer uma rotação de 360 graus em nenhuma parte. Para limitar os movimentos foi definido empiricamente um ângulo de rotação que se aproximasse o mais possível da realidade. Assim, existe uma relação entre o movimento horizontal do rato e o limite definido para cada ponto, em que o valor da rotação vai aumentando ou decrementando até chegar ao limite. Por exemplo, os pontos que correspondem aos cotovelos do modelo, podem rodar no eixo Y, até um limite cerca de 85 graus.

5.3.3 Criação dos Frames

Esta fase consiste em preencher o ficheiro com *frames*. Clarificando que o ficheiro nesta altura não está vazio, mas sim com a posição inicial e as configurações das mãos.

O processo de criação de *frames* passa pelo utilizador movimentar o modelo como quiser, e para criar um *frame* deve pressionar num botão da interface (o “Create Frame” da [Figura 5.7](#)). O número de *frames* foi limitado a 6 por duas razões. A primeira, porque era necessário limitar de modo a não dar hipótese ao utilizador de criar *frames* infinitamente. E segundo, porque empiricamente cheguei à conclusão que 6 *frames* seriam suficientes para criar animações simples e complexas.

5.3.4 Interface

Foi criada uma interface ([Figura 5.7](#)) para facilitar o processo de criação que, além do botão de criação de *frames* mencionado na seção anterior, contém ainda as seguintes opções:

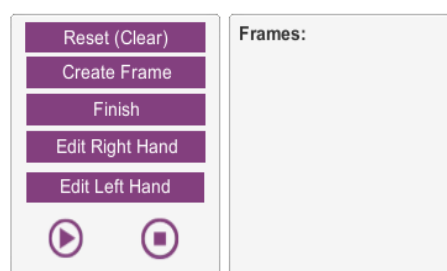


Figura 5.7: Painel Principal do Sistema 2.

1. **Reset:** Coloca o modelo na posição inicial e elimina todos os *frames* criados até ao momento.
2. **Create:** Cria um *frame* no ficheiro de animação.
3. **Finish:** Termina o processo de criação e volta ao menu principal ([Seção 7.1](#)) sendo que é criado o estado correspondente à nova animação no ficheiro que contém todos os estados ([Subseção 5.3.5](#)). Depois de concluída a animação, esta estará disponível tanto na interface do sistema 1 ([Subseção 4.1.3](#)), como no sistema de visualização de todos os gestos ([Seção 7.2](#)).

4. **Edit Right Hand e Edit Left Hand:** Altera, a qualquer momento, a configuração de uma das mãos à escolha.
5. **Play:** Toca a animação atual, ou seja, serve para ver o que já criamos até ao momento.
6. **Stop:** Para o que estiver a ser tocado, isto é, interrompe o Play.

Existem ainda outros três painéis na interface:

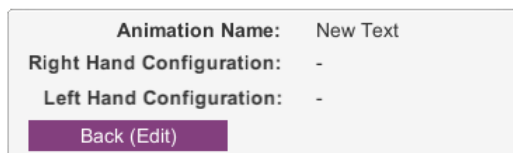


Figura 5.8: Painel 1



Figura 5.9: Painel 2



Figura 5.10: Painel 3

O painel da [Figura 5.8](#) contém informações gerais sobre o sistema e um botão para voltar ao ecrã anterior. Nesse menu existem 3 informações básicas: o nome da animação, o nome da configuração da mão esquerda e da mão direita. No segundo painel da [Figura 5.9](#), a primeira opção serve para voltar ao menu principal, sendo que, caso o utilizador tenha criado algum *frame*, será alertado para a perda desses valores e o segundo botão serve para mostrar informações ao utilizadores sobre o funcionamento do sistema do ecrã atual que se encontra, ou seja, parte do que já foi referido nas seções anteriores. Este painel está presente em todo o sistema. Por fim, o terceiro e último painel da [Figura 5.10](#), serve para alternar o eixo mencionado na [Subseção 5.3.2](#).

5.3.5 Conclusão da Animação

Esta seção serve para enfatizar o último passo que finaliza a criação da animação. Aparentemente, ao pressionar o botão “Finish” a animação é concluída, mas internamente ainda é feito mais um passo, a criação de um estado num repositório de estados. Estes estados na verdade correspondem a todas as animações criadas e servem como base de dados para o sistema de criação de estados apresentado [Subseção 4.1.2](#).

5.4 Plugin para Conversão de Animações

O Mecanim utiliza músculos, isto significa que a estrutura dos *bones* é interpretada de forma diferente. Em vez de movimentar cada ponto do *rig*, um a um, através dos referenciais X,Y ou Z, estes são movimentados utilizando os músculos. Basicamente, um músculo é um valor normalizado entre -1 e 1 que movimenta um ou mais *bones* num eixo, e servem para tornar mais realísticos os movimentos dos modelos.

No entanto, há uma desvantagem que deu origem à criação do plugin para o Unity que converte animações. A desvantagem é não se ter controlo dos movimentos, ou seja, imaginando que o ombro esquerdo do modelo se encontra no referencial X com valor 10, não é possível saber a que valor corresponde em termos de músculos. Por essa razão, e uma vez que o sistema 2 ([Capítulo 5](#)) é

baseado em movimentos usando o sistema de coordenadas X, Y e Z, não era possível manter os músculos.

Desta forma, foi criado um plugin (Figura 5.11) no Unity que, dado um modelo e a respetiva animação a converter no formato de músculos, transforma-a para o sistema de coordenadas normal para o modelo escolhido.

O plugin é útil, não para o resultado das animações do sistema 2 (Capítulo 5), mas sim para misturar essas animações às já existentes no PE2LGP original uma vez que ao serem importadas para o Unity ficariam em formato de músculos. O problema surge quando se misturam animações de diferentes tipos, que resultaram em problemas na orientação e posição do modelo.

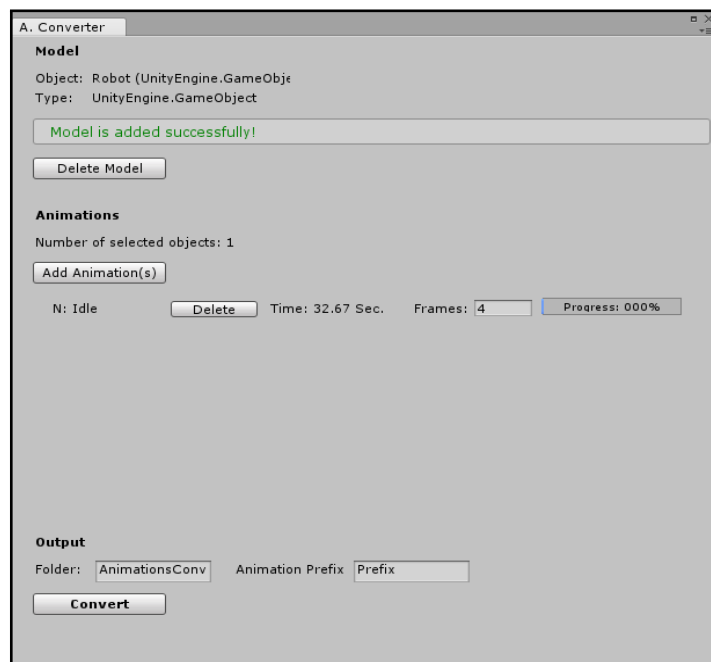


Figura 5.11: Interface do Plugin para Converter Animações.

Salientando que, nas versões mais recentes do Unity (pelo menos 5.4), já é possível de um modelo obter a *Human Pose*². Basicamente, a *Human Pose* é constituída por: *bodyPosition*, *bodyRotation* e *muscles*. Por outras palavras, isto significa que é possível obter os dados que antes não era possível, como por exemplo, a partir de um músculo, obter a sua posição ou rotação em termos de coordenadas X, Y ou Z.

Imaginando que temos uma animação chamada IDLE usando os músculos teria a representação da Figura 5.12, onde é possível verificar que músculos são utilizados na animação, em que para cada músculos existe apenas um único valor a variar entre -1 e 1.

²<https://docs.unity3d.com/ScriptReference/HumanPose.html>

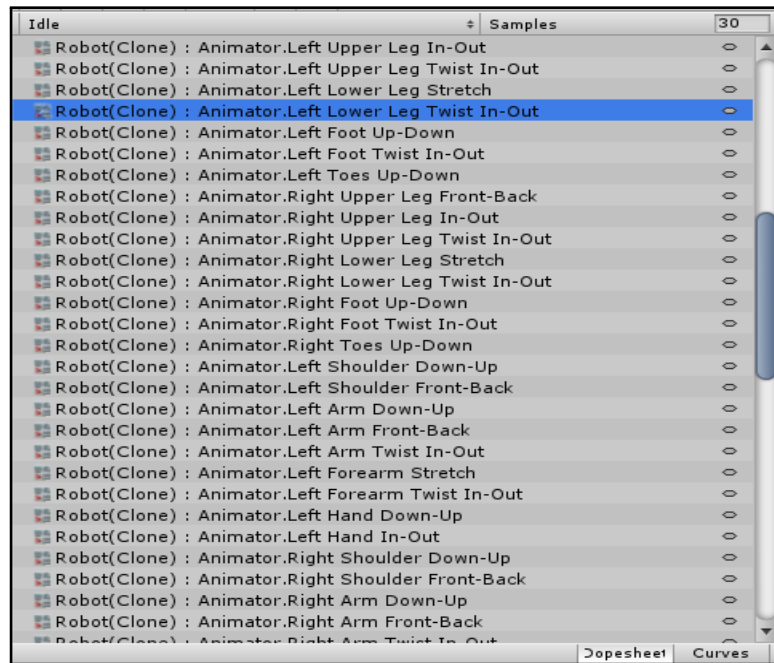


Figura 5.12: Animação Antes da Conversão.

Aquando transformada a animação utilizando o plugin, esta fica com a representação da [Figura 5.13](#), onde em vez de músculos para cada ponto, existe um ponto referencial normal, com as representações do sistema de coordenadas X, Y e Z.

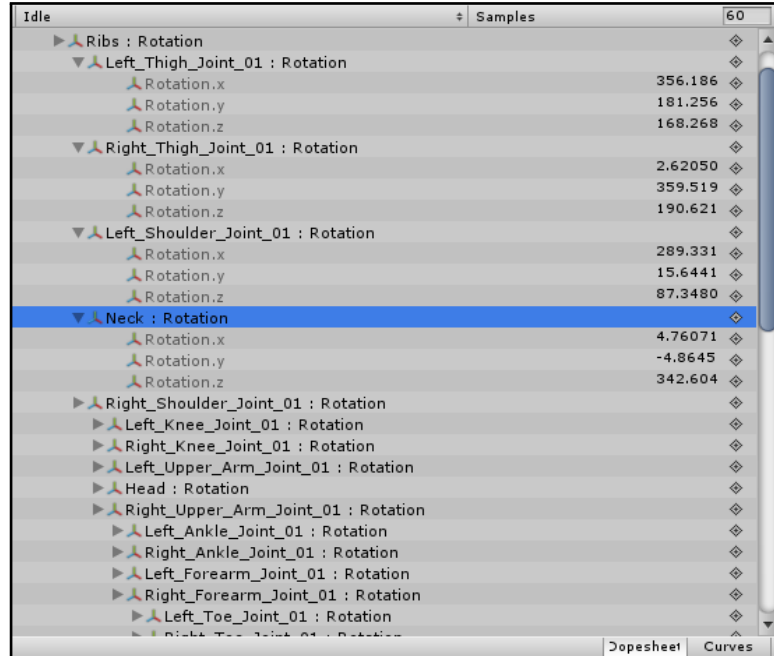


Figura 5.13: Animação Depois da Conversão.

6

Sistema 3 - Criação Automática de Gestos

Conteúdo

6.1	Introdução ao Sensor Kinect One	61
6.2	Mapeamento do Skeleton Tracking ao Modelo	62
6.3	Comandos de Voz	62
6.4	Melhorias e Extras	63
6.5	Interface	66

Neste capítulo apresenta-se em detalhe o sistema de criação de animações automático. Este, ao contrário do anterior ([Capítulo 5](#)), é automático, o que significa que requer pouca intervenção do utilizador para a criação da animação que corresponde a um gesto no PE2LGP. A ideia é utilizar o sensor Kinect para capturar os movimentos do utilizador, criando assim uma animação. No sistema anterior, estes movimentos teriam todos que ser feitos com a intervenção do utilizador através da movimentação manual do modelo. Serão também apresentadas as componentes que se podem encontrar na interface deste sistema.

6.1 Introdução ao Sensor Kinect One

Antes de mais, o Kinect One é o mais recente sensor de captura de movimentos criado pela Microsoft e lançado em 2014. Em relação à versão anterior, lançada em 2010, esta destaca-se sobretudo em: maior resolução, quer na camera de cores, quer na de profundidade, mais *joints* definidos no esqueleto, isto é, de 20 para 26, e mais esqueletos *tracked*.

A Microsoft em 2014, no mesmo ano do lançamento do sensor, lançou um plugin para o Unity¹ que contém uma demo que mostra algumas das funcionalidades do Kinect usando a *SDK*. Por outras palavras, este plugin veio facilitar o trabalho dos desenvolvedores na criação de aplicações. Uma das opções que o plugin tem é a hipótese de fazer *tracking* ao corpo, sendo cada parte do *skeleton* captada representada por cubos que depois são ligados por meio de retas e atualizados a cada *frame*, tal como se pode verificar na figura [Figura 6.1](#).

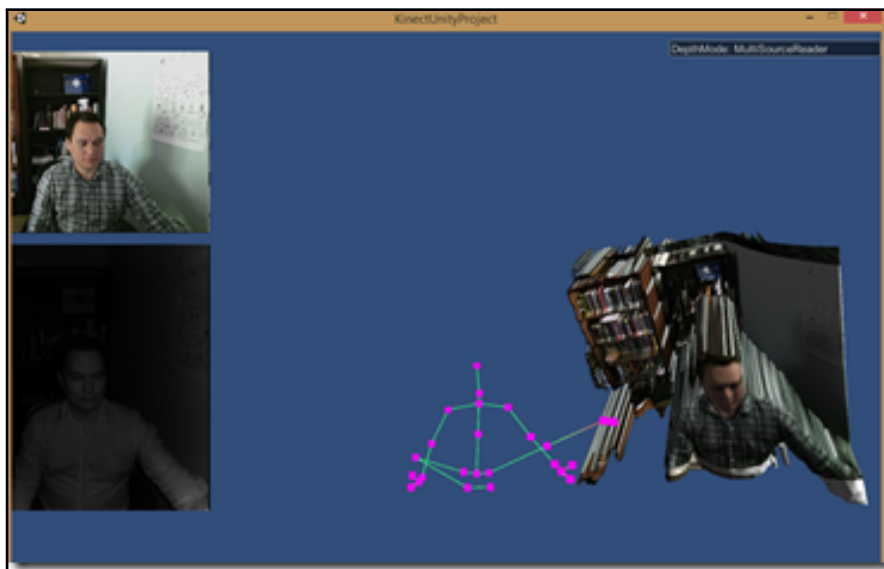


Figura 6.1: Exemplo do Tracking.

Portanto, a utilização deste plugin para o Unity foi o ponto de partida na utilização do sensor.

¹Disponível em: go.microsoft.com/fwlink/?LinkID=513177

6.2 Mapeamento do Skeleton Tracking ao Modelo

Passando os preliminares da ambientação ao sensor e depois de explorado o Plugin da Microsoft para o Unity, o principal foco foi mapear os pontos do *skeleton* capturados pelo Kinect, em qualquer avatar, em especial o modelo do PE2LGP. O plugin do Kinect para Unity que utiliza a *SDK* já faz o *tracking* de 26 pontos do esqueleto do corpo humano. O Unity, por sua vez, dado um modelo, é possível obter os mesmos 26 pontos do rig. Portanto, em cada *frame* em que se consegue capturar os pontos, são aplicadas algumas transformações a esses pontos, como por exemplo, normalizar e colocar no mesmo sistema de coordenadas. Posteriormente, os pontos tratados são mapeados no avatar. Assim, é possível movimentar qualquer avatar utilizando o Kinect. O próximo passo seria com esses movimentos criar uma animação. Para esta tarefa foi possível contar com a contribuição de Rumen Filkov que disponibilizou o seu projeto² para fins de investigação. O seu blog, focado especialmente no Kinect, também foi importante³.

6.3 Comandos de Voz

Após o mapeamento dos pontos capturados pelo Kinect no modelo explicado na seção anterior, concluiu-se que seria necessário incluir comandos de voz, no mínimo um. A necessidade da introdução de comandos de voz surge em virtude de, se o utilizador usar o sistema sozinho para criar a animação, o movimento de ir até ao computador para terminar a animação faria parte da própria animação. Entre várias técnicas que poderia ter adotado, como por exemplo, reconhecimento de algum movimento indicativo de uma ação, optei pelos comandos de voz, considerando serem mais intuitivos e fáceis. Assim, o utilizador quando terminasse de criar a animação poderia completá-la através de um comando de voz sem ter de se deslocar até ao computador.

Mais uma vez, a implementação destes comandos foram feitos a partir das bibliotecas disponíveis na SDK. Basicamente, o microfone do Kinect ativa-se e fica à espera de receber *input*. Quando consegue captar, tenta interpretar e consequentemente classificar. A classificação tem um grau de confiança que varia entre 0 e 100 e dá-nos a quanto “certo” está o que dissemos.

Inicialmente foram implementados 4 comandos de voz:

1. **Recording:** Começar a gravar a animação.
2. **Stop:** Parar a gravação ou interromper a ação “Play”.
3. **Finish:** Concluir e voltar ao menu principal.
4. **Play:** Rever o que já teria sido gravado.

Sendo que, a aceitação da palavra estava inicialmente para um grau de confiança de pelo menos 0, ou seja, qualquer coisa seria interpretada como correta. Depois, e de forma empírica assumiu-se que só aceitava a classificação se o grau de confiança fosse acima dos 90 por cento, obtendo resultados melhores. No entanto, verificou-se que os comandos de voz nem sempre funcionam,

²<https://www.assetstore.unity3d.com/en/#!/content/18708>

³<https://rfilkov.com/>

dependem do ambiente, sobretudo do ruído presente. Por isso, decidi reduzir os comandos de voz a apenas o estritamente necessário, ou seja, o “Stop”.

Assim, o utilizador quando quiser começar a gravar, deve carregar na tecla “R” ou no botão “Start Recording” da interface (Figura 6.5), sendo que, a partir desse momento aparecerá um contador decrescente (Subseção 6.4.4) que consiste no tempo de desconto (definido empiricamente 5 segundos) até a pessoa se colocar na posição *TPose* e visível perante o Kinect (Subseção 6.4.5).

O comando de voz pode ser ativado ou desativado a qualquer momento recorrendo à interface (Figura 6.5). Além disso, a funcionalidade de “Stop”, ativada pela voz, pode também ser executada através de um botão na interface ou pressionando a tecla “S”.

6.4 Melhorias e Extras

Depois de estar concluída uma primeira versão deste sistema, foram feitas algumas melhorias e adicionadas algumas funcionalidade, sendo que esta seção tem como objetivo apresentar essas alterações.

As modificações surgiram em virtude da necessidade de tornar o sistema com a maior qualidade possível e também pelo *feedback* recebido através do evento LXMLS⁴ realizado no final do mês de Julho, onde várias pessoas tiveram oportunidade de ver e utilizar o sistema.

6.4.1 Limitações no Movimento Capturado

Inicialmente não foram impostas nenhuma restrição quanto ao movimento. Achei que era uma experiência mais dinâmica para o utilizador ver a captura de todos os pontos, embora na altura da criação só estivessem a ser guardados os valores de alguns dos pontos.

No entanto, e devido ao *feedback* que obtive no evento LXMLS, tive de mudar esta abordagem por dois motivos. O primeiro é que não preciso de capturar todos os pontos do esqueleto, apenas os 8 pontos já referidos na Subseção 5.3.1. Segundo, porque o fato de capturar todos os pontos iria contribuir para o aumento da probabilidade do sistema não funcionar, isto porque o sistema requer que o utilizador esteja numa posição em que todos os pontos pudessem ser capturados pelo sensor. Desta forma, ao limitar a captura para apenas 8 pontos, o utilizador tem mais liberdade no seu movimento, ou seja, basta que o Kinect consiga capturar da cintura para cima, podendo o utilizado estar, por exemplo, sentado.

6.4.2 Múltiplas Câmaras para Feedback

Mais uma vez, a pensar em questões de usabilidade, decidi integrar na interface múltiplas câmaras tal como se pode verificar indicado através dos números 1 e 2 presentes na Figura 6.2.

No lado esquerdo temos a câmara RGB do sensor, e no lado direito temos a mesma câmara RGB mas com 2 particularidades: a ilustração dos pontos capturados ligados por meio de retas que são

⁴O LXMLS é um evento anual realizado no Instituto Superior Técnico, denominado de “escola de verão” (*summer school*) que cobre vários tópicos no âmbito da aprendizagem automática (*Machine Learning*). Mais detalhes em: <http://lxmls.it.pt/2016/>



Figura 6.2: Referência às Duas Câmaras.

atualizados a cada *frame*, e a eliminação dos pixels que não estão perto desses mesmos pontos dando um efeito *Chroma Key*. O *Chroma key* é uma técnica de efeito visual que consiste em colocar uma imagem sobre outra, anulando o background de uma delas através do anulamento de uma cor padrão, como por exemplo o verde ou o azul. Por outras palavras, esta técnica substitui o background de modo a isolar personagens ou objetos de interesse.

6.4.3 Câmara Principal Associada ao Modelo

Inicialmente, a câmara que está apontada ao modelo era independente e apanhava todo o corpo. As consequências negativas desta abordagem é que o utilizador ao movimentar-se para a frente ou para trás, o modelo faria o mesmo movimento e, portanto, ficaria ou mais distante ou mais perto. Na maioria dos casos ficava sempre distante, sendo que o espaço do ecrã não era totalmente ocupado e ficava difícil perceber o detalhe ao nível das mãos. Por essa razão, decidi mudar ligeiramente a metodologia da câmara, colocando-a como “filho” (*child*) do modelo, ficando assim associada ao mesmo. Quer isto dizer, que a câmara não é mais independentemente do modelo e move-se de acordo com o mesmo. Por outras palavras, o utilizador verá o modelo a ocupar todo o espaço do ecrã sempre. Além disso, e de acordo com a modificação descrita na [Subseção 6.4.1](#), a câmara apanha somente o modelo da cintura para cima, visto que o resto não é importante e mantém-se sempre estático.

6.4.4 Contador Decremental na Criação da Animação

Existe um contador decremental com um valor definido empiricamente de 5 segundos ([Figura 6.3](#)). O objectivo deste contador é o início da criação da animação não começar imediatamente. Isto é útil especialmente quando o sistema é utilizado por uma única pessoa, sendo que a pessoa carrega na tecla “R” para começar a gravar e tem 5 segundos para se colocar na posição correta. Caso o utilizador não queira usar a interface, poderá usar os comandos de voz ([Seção 6.3](#)).



Figura 6.3: Exemplo do Contador Decremental.

6.4.5 TPose

A *TPose* surgiu também em virtude do feedback do evento LXLMS. Basicamente, na primeira versão do sistema, não era pedido nada ao utilizador além de se ter de colocar numa posição suficiente para que o Kinect pudesse capturar. Contudo, aconteceu, por grande parte das vezes, o Kinect capturar mais do que uma pessoa por estarem próximas e começar a misturar os pontos de captura do modelo. Por isso, o modelo ficaria muitas vezes defeituoso, perturbando assim a criação das animações ou o sistema de um modo geral. Portanto, foi criada a *TPose*, em que é requerido ao utilizador, antes de começar a gravar, que se coloque nessa posição, de braços abertos e esticados (Figura 6.4). Desta forma, o Kinect captura apenas essa pessoa durante todo o processo, sem que outras consigam intervir.

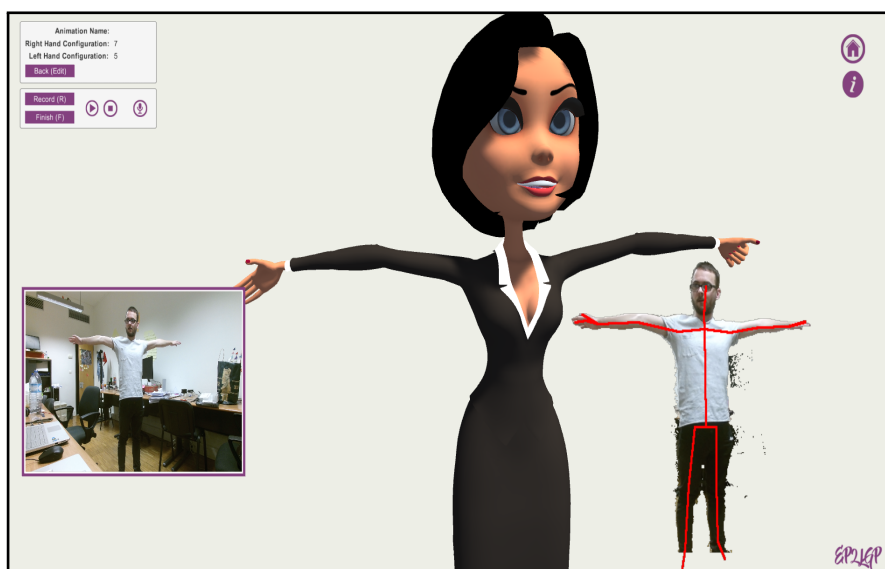


Figura 6.4: Exemplo da TPose.

6.4.6 Nome da Animação

Sem detalhar, este sistema de criação, dispõe como primeiro ecrã para a criação do ficheiro e da escolha do nome da animação, um menu análogo ao descrito na [Seção 5.1](#) para o Sistema Manual.

6.4.7 Seleção da Configuração das Mãos

O Kinect, por omissão, não captura a mão ao nível do detalhe dos dedos. Na melhor das hipóteses, captura o ponto central da mão e o polegar, dependendo da posição da mão em relação ao Kinect. Desta forma, e sendo a LGP maioritariamente executada através das mãos, era necessário arranjar uma solução embora pudesse ser redutora, no sentido de limitar o sistema. Deste modo, o utilizador dispõe de 2 ecrãs, idênticos aos que foram descritos na [Seção 5.2](#) para o Sistema Manual. Assim, existe a hipótese de escolher com qual das configurações das mãos quer criar a animação. Por outras palavras, a captura das mãos bem como outros pontos ([Subseção 6.4.1](#)), foram desativos, ou seja, permanecem estáticos durante a criação da animação. Assim sendo, esta é uma limitação deste sistema.

6.5 Interface

Nesta seção, tal como nos capítulos anteriores, pretendo apresentar a interface e todas as suas funcionalidades.

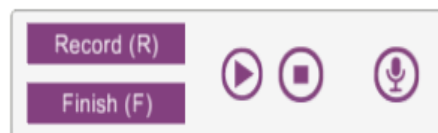


Figura 6.5: Interface para Controlar o Kinect.

1. **Record:** Inicia a gravação, começando pela aparição do contador decremental.
2. **Finsh:** Completa a gravação e voltar ao menu principal.
3. **Play:** Toca a gravação atual, ou seja, serve para ver o que já criamos até ao momento.
4. **Stop:** Para o que estiver a ser tocado.
5. **Microfone:** Ativa ou desativa os comandos de voz.

Além da interface, o utilizador pode aceder às mesmas ações (do ponto 1 até 4), pressionando nas teclas correspondentes às iniciais das ações: R para Record, F para Finish, P para Play e S para Stop.

7

Unificação dos Sistemas

Conteúdo

7.1	Menu Principal	68
7.2	Visualizador das Animações	68
7.3	Usabilidade e Design	69
7.4	Conclusão	70

Neste capítulo é explicada a origem de unificar todos os sistemas e as modificações que posteriormente advieram. Em primeiro lugar, e como já foi referido anteriormente, o projeto foi pensando na criação de dois sistemas em plataformas diferentes que se interligam. O primeiro sistema é focado na mudança da arquitetura que introduz o motor de jogo Unity. O segundo, foca-se na utilização do sensor Kinect para a criação de animações. Porém, uma vez que o segundo sistema foi passado para o Unity, pensou-se em unificar ambos os sistemas e criar uma única aplicação, juntando também a interface explicada no [Capítulo 4](#).

Inicialmente, a aplicação teria 3 sistemas: a interface ([Capítulo 4](#)), o sistema de criação manual ([Capítulo 5](#)) e o sistema de criação automático ([Capítulo 6](#)). Para complementar, ainda foi adicionado um sistema que permite visualizar todas as animações disponíveis no sistema ([Seção 7.2](#)).

7.1 Menu Principal

O menu principal que contém os quatro painéis ([Figura 7.1](#)), é o primeiro contato com o utilizador e tem como objetivo apresentar todos os sistemas disponíveis. Cada um é representado através de um painel que contém uma imagem ilustrativa e uma breve descrição.



Figura 7.1: Painéis do Menu Principal.

O utilizador poderá escolher qualquer um dos sistemas e voltar ao menu principal a qualquer momento.

7.2 Visualizador das Animações

O visualizador das animações corresponde ao último painel apresentado na figura [Figura 7.1](#) e tem como objetivo dar a hipótese ao utilizador de poder visualizar todas as animações que contém o modelo, ou seja, ver todos os gestos do PE2LGP. Para cada gesto é possível:

1. Ver repetidamente até ordem contrária.
2. Parar.
3. Apagar.

No caso da ação apagar, esta tem duas características. A primeira, é que só é possível em animações que não sejam padrão, isto é, que forem criadas através da nova arquitetura do PE2LGP e não da anterior. Por outras palavras, só é possível apagar animações que forem criadas, ou pelo sistema 1, ou pelo sistema 2. A segunda, é que o utilizador ao fazer uso desta ação, ser-lhe-á apresentado um menu de confirmação, evitando assim, cometer erros.

Deste modo, a figura [Figura 7.2](#) representa o visualizador e as respetivas ações para cada gesto.

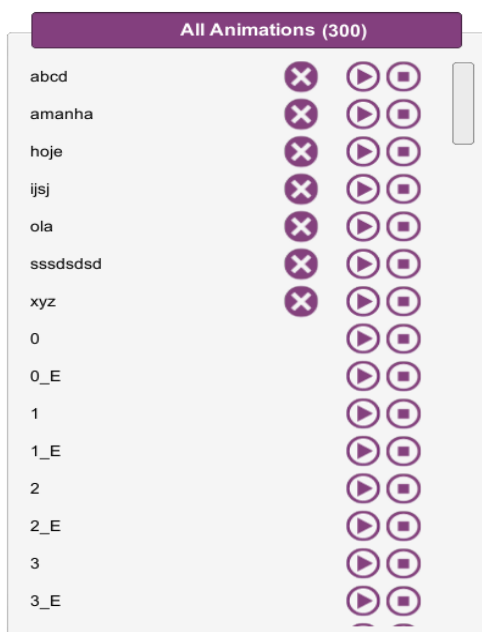


Figura 7.2: Interface Principal da Lista de Animações.

Além das três principais ações, o utilizador poderá ainda, tal como no primeiro sistema criado em virtude da modificação à arquitetura apresentado no [Capítulo 4](#), interagir com o modelo para:

1. Movimentar o modelo em torno do seu próprio eixo X.
2. Fazer zoom-in ou zoom-out ao modelo.

7.3 Usabilidade e Design

Na unificação do sistema foram focados dois pontos principais: a usabilidade e o design. Assim, tentou-se sempre transparecer a imagem de que o sistema seria apenas um e não 3 separados que, por acaso, teriam sido ligados. Por isso, era importante, criar algum tipo de consistência entre os diferentes ecrãs presentes.

No diz que respeito ao painéis do menu principal, decidiu-se juntar não só uma breve descrição do que representa o sistema, mas também uma imagem ilustrativa.

As cores foram escolhidas de forma empírica, resultando assim em 4 principais: roxo, branco, preto, e bege. O objetivo era que as cores claras em conjunto com as escuras, pudessem não só fazer contraste, mas também que não atrapalhem o modelo e as suas animações. No caso de cores escuras, pude verificar que por vezes era difícil distinguir os vários elementos do modelo.

Além do logótipo e das cores definidas, foi preciso manter a consistência noutros aspectos, como por exemplo: o tamanho e o tipo de fonte, mensagens de erro, ícones e as suas respetivas posições, assim como é o caso do ícone para voltar ao menu principal que se encontra no canto superior direito de todos os ecrãs.

7.4 Conclusão

Esta unificação permitiu juntar e manter consistentes os 3 sistemas apresentados nos capítulos anteriores e que são provenientes dos objetivos definidos inicialmente. Por um lado, temos a interface, e por outro, dois sistemas de criação de gestos. Para complementar a experiência do utilizador, ainda é possível visualizar todos os gestos. Assim sendo, pode-se dizer que o PE2LGP contém 4 sistemas.

8

Avaliação

Conteúdo

8.1	Métodos	72
8.2	Participantes	74
8.3	Métricas	74
8.4	Resultados	76
8.5	Conclusão	83

Este capítulo tem como objetivo apresentar as metodologias, os resultados e as conclusões da avaliação aos 3 sistemas descritos nos capítulos anteriores.

O propósito da avaliação foi:

1. *Testar a usabilidade dos 3 sistemas.*
2. *Avaliar a qualidade dos gestos produzidos pelos sistemas 2 e 3.*
3. *Comparar os sistemas 2 e 3.*

*A avaliação focou-se principalmente nos dois sistemas de criação de gestos: o sistema 2 ([Capítulo 5](#)) e o sistema 3 ([Capítulo 6](#)). **Estes dois sistemas tiveram dois momentos de avaliação.***

Num primeiro momento, pretendeu-se perceber como é que pessoas de diferentes áreas se adaptavam na utilização dos sistemas. Depois, num segundo momento, foi necessário perceber se os gestos produzidos pelos sistemas tinham qualidade, isto é, se eram adequados para serem utilizados no contexto da [LGP](#).

8.1 Métodos

Os métodos utilizados consistiram em pedir a pessoas para realizarem um conjunto de tarefas e, posteriormente, para preencherem um questionário.

O procedimento da avaliação foi semelhante nos 3 sistemas. Em primeiro lugar, foi feita uma breve apresentação de todo o PE2LGP e, depois, entregue, por cada sistema, 2 conjuntos de folhas: o primeiro com uma explicação mais detalhada do sistema em questão, e o outro com as tarefas a realizar, bem como um questionário¹. Para que ambos os sistemas de criação pudessem ser comparados, as tarefas, bem como os questionários, foram idênticos. Relativamente ao segundo momento da avaliação dos sistemas de criação de gestos, foi utilizado apenas o questionário.

8.1.1 Sistema 1

Relativamente ao primeiro sistema foi pedido aos avaliadores que realizassem dois conjuntos de tarefas. O primeiro para testarem a componente linguística, onde era pedido para introduzirem diferentes *inputs* através da interface. O segundo era relativo à usabilidade, onde era pedido para testarem as várias funcionalidades do sistema, como por exemplo, aumentar e diminuir a velocidade do gesto.

8.1.2 Sistemas 2 e 3 - Primeiro Momento

Dado que a avaliação envolvendo os três sistemas e a criação de um gesto, demoraria entre 30 a 45 minutos, decidiu-se que cada utilizador criaria um gesto em cada sistema (resultando em 10 no final da avaliação).

¹O formato de documento utilizado para todos os questionários poderá ser encontrado nas seções dos anexos [Subseção A.1.1](#), [Subseção A.1.2](#) e [Subseção A.1.3](#)).

Os 5 gestos definidos para a criação foram retirados do Dicionário de Língua Gestual Portuguesa de Ana Baltazar² e também da página Spreadthesign³. Nesta escolha surgiram as primeiras dificuldades. O segundo sistema, o automático que utiliza o sensor Kinect, não pode alternar a configuração das mãos durante a criação do gesto, pelo que, estariam excluídos muitos gestos. Além disso, tendo em conta o dicionário de LGP e a página Spreadthesign, existe uma inconsistência na forma como são produzidos os gestos, pelo que, foi um desafio obter gestos que se realizassem da mesma forma. Depois, foi ainda preciso garantir que era possível criar os gestos através de ambos os sistemas.

Ultrapassados os problemas iniciais, foram definidos para criação os seguintes gestos:

1. Baixo:

Dicionário: página 93

Spreadthesign: <https://media.spreadthesign.com/video/mp4/6/304870.mp4>

2. Hoje:

Dicionário: página 434

Spreadthesign: <https://media.spreadthesign.com/video/mp4/6/4201.mp4>

3. Mal:

Dicionário: página 599

Spreadthesign: <https://media.spreadthesign.com/video/mp4/6/5381.mp4>

4. Noite:

Dicionário: página 700

Spreadthesign: <https://media.spreadthesign.com/video/mp4/6/16558.mp4>

5. Pagar:

Dicionário: página 759

Spreadthesign: <https://media.spreadthesign.com/video/mp4/6/16655.mp4>

Cada utilizador teria de criar um dos cinco gestos, tendo como auxílio, não só o dicionário para consulta, para verificar como era executado o gesto, como também uma página HTML reunindo vídeos a ilustrar como executar os gestos.

8.1.3 Sistemas 2 e 3 - Segundo Momento

Depois de criados os 5 gestos por cada um dos sistemas, pessoas ligadas à LGP avaliariam a sua qualidade, utilizando para isso um questionário composto por várias perguntas e vídeos dos gestos.

²<https://www.portoeditora.pt/produtos/ficha/dicionario-de-lingua-gestual-portuguesa/3501376>

³Spreadthesign é um projeto piloto do Swedish International Programme Office of Education and Training, suportado pela União Europeia. Este projecto nasceu com o fundamento de ensinar a comunicar com surdos através da sua linguagem, a linguagem gestual. No site, pode aprender a traduzir esta língua em vários idiomas, tem vários vídeos que complementam de forma elucidativa este magnífico idioma. <https://www.spreadthesign.com/pt/>

O utilizador teria de identificar, através do vídeo, qual era o gesto que estava a ser executado e, caso conseguisse, teria, de acordo com uma escala de 1 a 5, avaliar a qualidade do gesto no contexto da LGP.

8.2 Participantes

No que diz respeito ao primeiro momento de avaliação, foram utilizadas 6 pessoas: 3 estudantes da área da Engenharia Informática, 1 um estudante ligado à LGP e 2 funcionários do IST não ligados à LGP nem à informática. O aluno envolvido na LGP é fluente em Libras e um potencial utilizador do PE2LGP. São todos da região de Lisboa e com idades compreendidas entre os 22 e os 33 anos.

O questionário correspondente ao segundo momento da avaliação dos sistemas 2 e 3, foi avaliado por um grupo de 6 pessoas ligadas à LGP. Os avaliadores são das regiões de Lisboa, Amadora e Oeiras, têm entre 22 a 52 anos e exercem as seguintes profissões: psicóloga, assistente social, professora, apoio de lavandaria, técnica e audiologista.

8.3 Métricas

As métricas usadas em todos os sistemas basearam-se em perguntas dos questionários, com o principal foco na usabilidade. As perguntas dos questionários tinham uma escala de 1 até 5, ou então resposta binária de “Sim” ou “Não”, sendo que, em caso negativo, o utilizador teria de a justificar.

8.3.1 Sistema 1

O questionário utilizado no sistema 1 é composto por 10 perguntas e corresponde à Tabela 8.1⁴:

Pergunta		Escala
4	De um modo geral como avalia o sistema?	1 - Difícil até 5 - Fácil;
5	Qual é a dificuldade na utilização de todo o sistema?	
8	De um modo geral como avalia o sistema?	1 - Mau até 5 - Bom;
1	Conseguiu perceber a transcrição de todos os gestos?	Sim; Não;
2	O tamanho da transcrição é adequado?	
3	A velocidade dos gestos é adequada?	
6	Adicionava mais opções?	
7	Removia alguma opção?	
9	Acha o sistema útil para aprender Língua Gestual Portuguesa?	
10	Tem alguma ideia para melhorar o sistema?	

Tabela 8.1: Questionário Utilizado na Avaliação do Sistema 1.

⁴A tabela foi ordenadas pela escala para uma melhor análise aos resultados.

O âmbito das perguntas focou-se sobretudo em perceber como teria sido a experiência do utilizador com o sistema, a sua opinião em alguns aspetos, como por exemplo, se removia opções ou se achava o sistema útil no contexto da LGP e, por fim, se tinha conseguido realizar as tarefas requeridas.

8.3.2 Sistemas 2 e 3 - Primeiro Momento

Quanto aos sistemas 2 e 3, o questionário contém 9 perguntas e corresponde à [Tabela 8.2⁵](#):

Pergunta		Escala
1	De um modo geral como avalia o sistema?	1 - Mau até 5 - Bom;
3	Qual é a dificuldade na utilização de todo o sistema?	1 - Difícil até 5 - Fácil;
5	Qual é a dificuldade na utilização de todo o sistema?	
2	Conseguiu criar todos os gestos?	Sim; Não;
4	Percebeu como funcionava todo o sistema?	
6	Acha as opções disponibilizadas adequadas?	
7	Adicionava mais opções?	
8	Acha o sistema útil para a Língua Gestual Portuguesa?	
9	Tem alguma ideia para melhorar o sistema?	

Tabela 8.2: Questionário Utilizado no Primeiro Momento da Avaliação dos Sistemas 2 e 3.

Tal como no sistema anterior, a escolha das perguntas surge em forma de tentar perceber como foi a interação do utilizador com os sistemas relativamente à usabilidade.

Além do questionário, foi utilizada uma métrica complementar: o tempo de criação de cada gesto.

8.3.3 Sistemas 2 e 3 - Segundo Momento

Cada avaliador tinha como tarefa identificar e classificar cada um dos 5 gestos: baixo, hoje, mal, noite e pagar. O questionário continha 3 perguntas por gesto e corresponde à [Tabela 8.3](#):

Pergunta		Escala
1	Penso que o gesto é:	Resposta aberta;
2	Tendo em conta a palavra que identificou, quão preciso considera o gesto executado pela Catarina?	0 - Não Reconheço, 1 - Mau até 5 - Perfeito;
3	Sugestões e/ou críticas (Opcional):	Resposta aberta;

Tabela 8.3: Questionário Utilizado no Segundo Momento da Avaliação dos Sistemas 2 e 3.

⁵A tabela foi ordenada pela escala para uma melhor análise aos resultados.

8.4 Resultados

Os resultados⁶ foram obtidos através das respostas dos questionários, de notas tiradas durante a avaliação e, no caso do primeiro momento dos sistemas 2 e 3, também da anotação do tempo de criação de cada gesto criado.

8.4.1 Sistema 1

Através dos resultados do questionário (Tabela 8.4, Figura 8.1 e Figura 8.2), podemos concluir uma consistência nas respostas entre diferentes utilizadores. O sistema é de fácil utilização e todas as opções adequadas. No entanto, todos os utilizadores sugerem a introdução de uma ou outra opção, nomeadamente um botão “Play”, sugerem ainda que o cursor esteja sempre na caixa de texto da interface. As alterações poderão melhorar a experiência do utilizador. Além disso, todos acharam o sistema útil no contexto da LGP.

		Pessoas						Total	Escala
		1	2	3	4	5	6		
Perguntas	4	4	5	5	5	5	5	4,8	1 - Difícil até 5 - Fácil;
	5	5	4	5	5	5	5	4,8	
	8	5	2	5	5	5	5	4,5	1 - Mau até 5 - Bom;
	1	1	1	1	1	1	1	1,0	0 - Não e 1 - Sim;
	2	1	1	1	1	1	1	1,0	
	3	1	1	1	1	1	1	1,0	
	6	1	0	1	1	1	0	0,7	
	7	0	0	0	0	0	0	0,0	
	9	1	1	1	1	1	1	1,0	
	10	0	0	1	0	1	0	0,3	

Tabela 8.4: Resultados do Questionário Utilizado na Avaliação do Sistema 1.

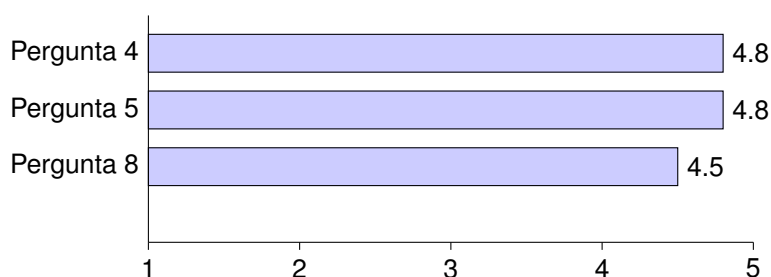


Figura 8.1: Resultados das Perguntas 4, 5 e 8, do Questionário do Sistema 1.

⁶Os resultados de todos os questionários podem ser encontrados nas seções dos anexos: Subseção A.1.3, Subseção A.1.4, Subseção A.1.5 e Subseção A.1.6

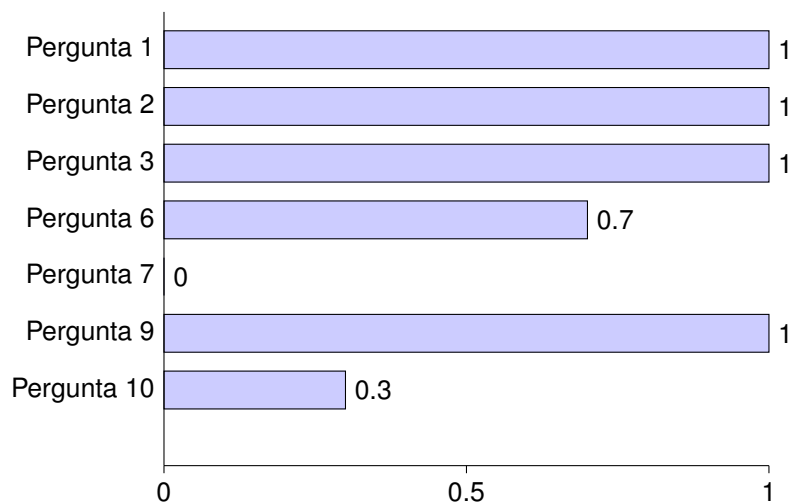


Figura 8.2: Resultados das Perguntas 1, 2, 3, 6, 7, 9 e 10, do Questionário do Sistema 1.

8.4.2 Sistema 2 - Primeiro Momento

O principal foco foi perceber a adaptação do utilizador ao sistema. De um modo geral, analisando os resultados ([Tabela 8.5](#), [Figura 8.3](#), [Figura 8.4](#) e [Figura 8.5](#)), a adaptação foi positiva, melhor do que o esperado. Utilizadores ligados a áreas diferentes da informática, como é o caso do aluno ligado à [LGP](#) e dos dois trabalhadores, facilmente se adaptaram. No entanto, em alguns casos, ainda persistiu alguma dificuldade em fazer a rotação desejada do modelo. As dificuldades podem ser ultrapassadas ao longo do tempo de utilização, embora com esta avaliação não se pudesse comprovar.

A criação de um gesto durou em média 11.8 minutos. Os valores por gesto podem ser consultados através da [Figura 8.5](#).

Foi unânime que as opções são adequadas: o utilizador percebeu o funcionamento do sistema, foi capaz de realizar todas as tarefas e acha o sistema útil no contexto da [LGP](#).

		Pessoas						Total	Escala
		1	2	3	4	5	6		
Perguntas	4	5	4	5	5	5	5	4,8	1 - Mau até 5 - Bom;
	5	5	3	4	3	5	4	4,0	1 - Difícil até 5 - Fácil;
	8	5	4	5	3	5	5	4,5	
	1	1	1	1	1	1	1	1,0	0 - Não e 1 - Sim;
	2	1	1	1	1	1	1	1,0	
	3	1	1	1	1	1	1	1,0	
	6	1	0	0	1	1	0	0,5	
	7	1	1	1	1	1	1	1,0	
	9	1	0	0	0	1	0	0,3	

Tabela 8.5: Resultados do Questionário Utilizado na Avaliação do Sistema 2.

Tal como o sistema anterior, os utilizadores sugerem a adição de mais funcionalidades, nomeadamente: edição de frames, rotação completa da mão incluindo os dedos em vez do limite ser o pulso, escolha da configuração das mãos por imagem e não só por texto, rotação de vários pontos do *rig* em simultâneo criando um efeito de “arrastamento”, limitar melhor o ângulo de rotação dos 8 pontos definidos, entre outros.

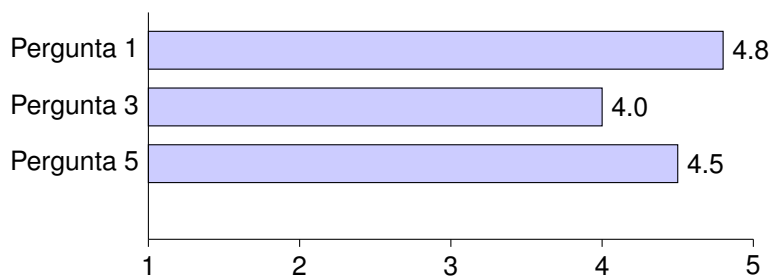


Figura 8.3: Resultados das Perguntas 1, 3 e 5, do Questionário do Sistema 2.

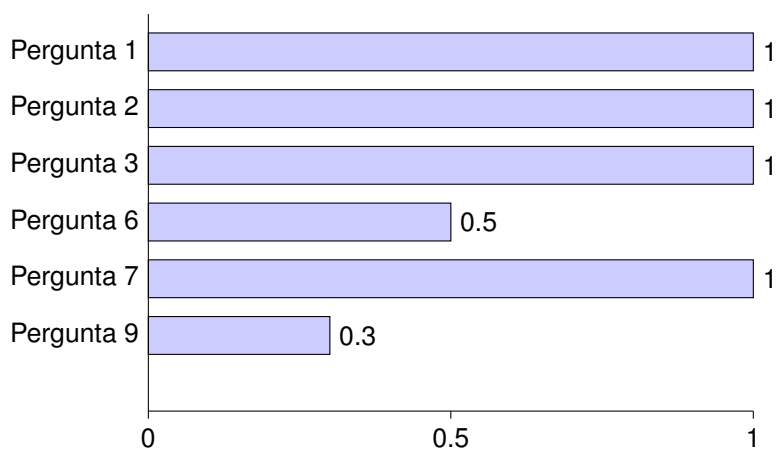


Figura 8.4: Resultados das Perguntas 1, 2, 3, 6, 7 e 9, do Questionário do Sistema 2.

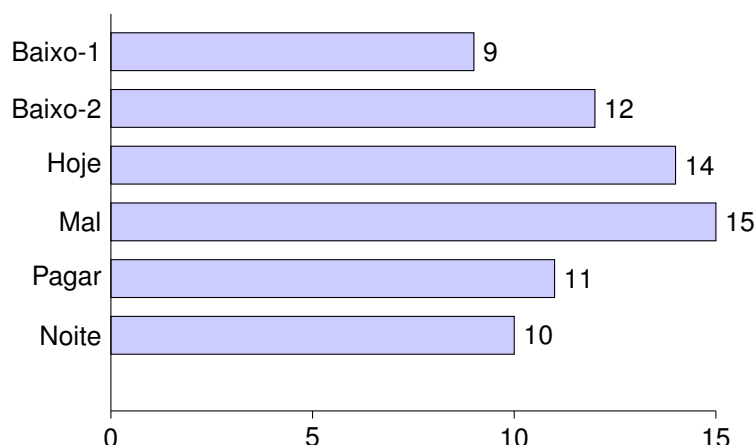


Figura 8.5: Tempo de Criação, em Minutos, de Um gesto por Pessoa.

Observação: O gesto baixo encontra-se repetido na [Figura 8.5](#) porque foi criado por duas pessoas.

8.4.3 Sistema 3 - Primeiro Momento

Considerando os resultados ([Tabela 8.6](#), [Figura 8.6](#), [Figura 8.7](#) e [Figura 8.8](#)), o sistema 3 revelou-se mais instável do que o esperado, no sentido em que alguns utilizadores, ou não conseguiram concluir a tarefa ou tiveram dificuldades em fazê-lo, isto porque foi avaliado num ambiente diferente daquele em que foi produzido. Em alguns casos, o ruído no momento da avaliação foi constante, dificultando o reconhecimento de voz por parte do sensor Kinect. Houve mesmo um utilizador que não conseguiu concluir a tarefa, devido a não conseguir parar a gravação através dos comandos de voz. Além disso, este não foi o único obstáculo: muitas vezes, sobretudo com pessoas fora da área da informática, a compreensão do funcionamento do reconhecimento de voz não foi adequado, sendo que a pausa entre comandos não era a suficiente para que fosse feito o reconhecimento. Desta forma, praticamente todos os utilizadores tiveram problemas a parar a gravação do gesto ou até mesmo a concluí-la, sobretudo quando havia algum ruído no ambiente. Por vezes, a paragem da gravação tinha de ser feita recorrendo aos botões na interface. Contudo, os resultados não evidenciam esses problemas, devido às perguntas dos questionários não terem em conta problemas na realização da tarefa, mas sim, se foi ou não concluída.

À parte deste problema e das limitações do próprio sistema, como por exemplo, não estar a ser feita a captura das mãos e a rotação de alguns pontos do corpo, o sistema revelou-se fácil de utilizar.

A criação de um gesto durou em média 2.4 minutos. Os valores por gesto podem ser consultados através da [Figura 8.8](#).

Houve apenas uma única sugestão para adicionar novas funcionalidades e refere-se quase que à mistura de ambos os sistemas, isto é, criar o gesto com o Kinect e, em algum momento, refinar a captura manualmente em pontos específicos, utilizando um mecanismo semelhante ao sistema 2.

No entanto, este sistema, ao contrário do anterior, não está adequado para o uso no contexto da [LGP](#), devido não só às limitações já referidas anteriormente, como aos problemas encontrados na

avaliação.

		Pessoas						Total	Escala
		1	2	3	4	5	6		
Perguntas	4	5	3	5	5	5	N.R.	4,6	1 - Mau até 5 - Bom;
	5	4	5	5	4	4	N.R.	4,4	1 - Difícil até 5 - Fácil;
	8	5	5	5	4	5	N.R.	4,8	
	1	1	0	1	1	1	N.R.	0,8	0 - Não e 1 - Sim;
	2	1	1	1	1	1	N.R.	1,0	
	3	1	1	1	1	1	N.R.	1,0	
	6	0	0	0	0	0	N.R.	0,0	
	7	1	1	1	1	1	N.R.	1,0	
	9	0	1	0	1	0	N.R.	0,4	

Tabela 8.6: Resultados do Questionário Utilizado na Avaliação do Sistema 2. N.R - Não Respondeu (um dos seis utilizadores, não teve oportunidade de avaliar este sistema).

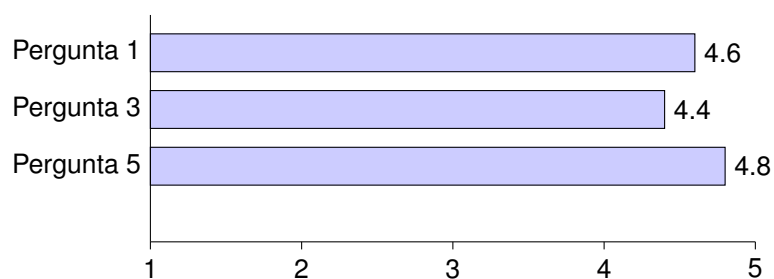


Figura 8.6: Resultados das Perguntas 1, 3 e 5, do Questionário do Sistema 3.

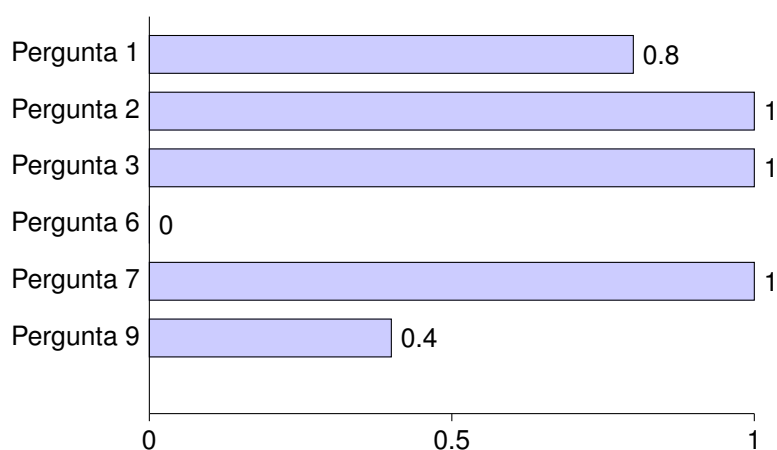


Figura 8.7: Resultados das Perguntas 1, 2, 3, 6, 7 e 9, do Questionário do Sistema 3.

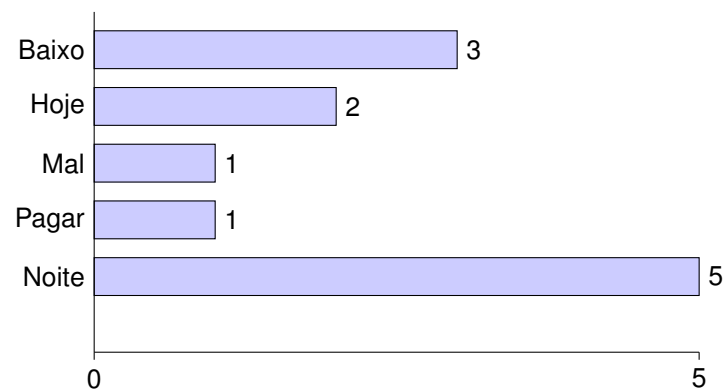


Figura 8.8: Tempo de Criação, em Minutos, de Um gesto por Pessoa.

8.4.4 Sistema 2 - Segundo Momento

Os 10 gestos, 5 em cada sistema, foram criados por mim e enviados aos avaliadores através de um questionário online. Os resultados (Figura 8.9 e Figura 8.14) revelam que poucas pessoas acertaram os gestos, apenas 3 gestos foram descobertos uma vez. Algumas pessoas não responderam e outras responderam erradamente, pelo que a classificação dos gestos não é precisa. As classificações oscilam entre todos os valores da escala. Assim sendo, pode-se concluir que os gestos não estão a ser realizados de modo adequado no contexto da LGP.

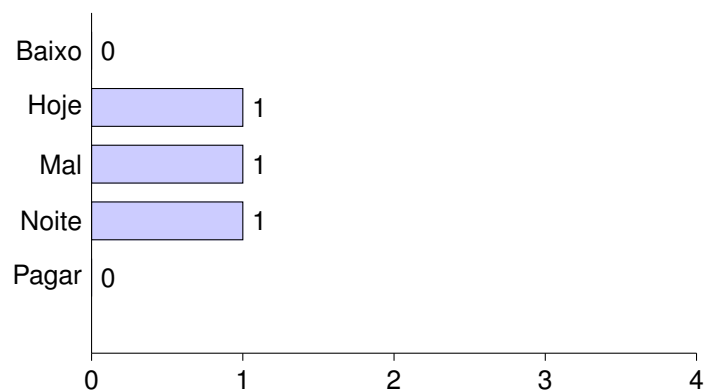


Figura 8.9: Número de Pessoas que Adivinharam os Gestos do Questionário do Sistema 2.

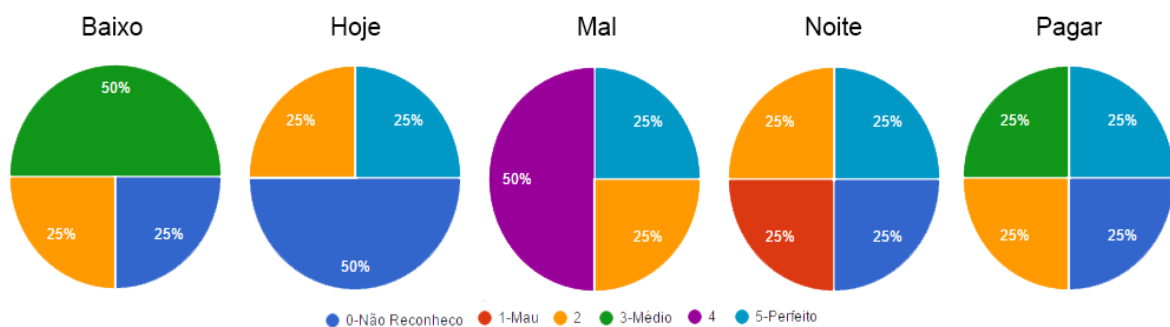


Figura 8.10: Resultados da Classificação dos Gestos do Sistema 2.

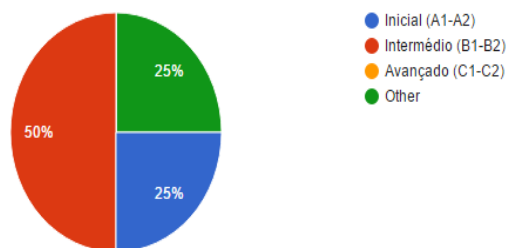


Figura 8.11: Resultados do Nível de LGP dos Avaliadores do Sistema 2.

8.4.5 Sistema 3 - Segundo Momento

Os resultados do segundo momento do sistema 3 (Figura 8.12 e Figura 8.13) são semelhantes ao do sistema 2, isto é, poucas pessoas acertaram os gestos e a classificação oscilou entre os vários valores da escala. No entanto, quanto à classificação, verificou-se que foi pior, sendo o valor mais registado o zero (não reconhecimento). Tal como foi a avaliação anterior, conclui-se que os gestos não estão a ser realizado de modo adequado no contexto da LGP.

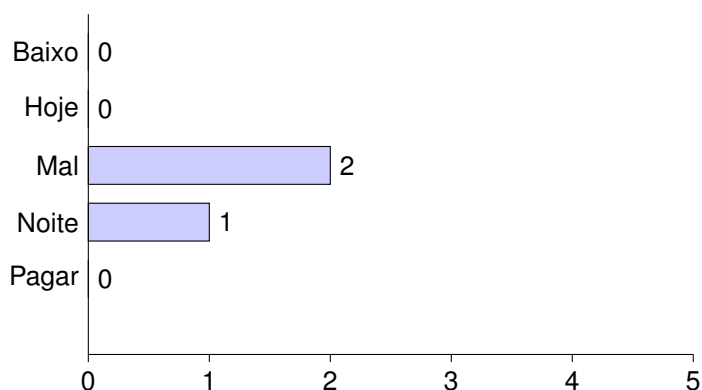


Figura 8.12: Número de Pessoas que Adivinharam os Gestos do Questionário do Sistema 3.

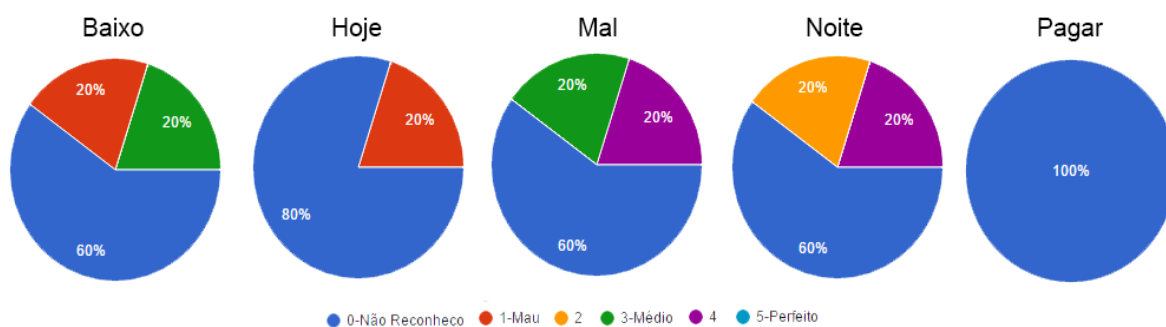


Figura 8.13: Resultados da Classificação dos Gestos do Sistema 3.

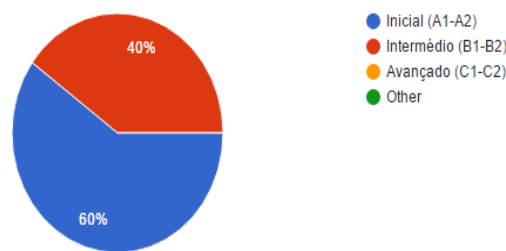


Figura 8.14: Resultados do Nível de LGP dos Avaliadores do Sistema 3.

8.4.6 Comparação entre os Sistemas 2 e 3

Há dois fatores importantes que distinguem os sistemas. Por um lado, o sistema 3, no tempo de criação é superior ao primeiro, uma vez que dura menos de um terço do que o sistema 2. Por outro lado, a dificuldade em criar um gesto poderá depender da sua complexidade, no entanto, relativamente àqueles que foram testados, o sistema 2 revelou-se melhor. No que diz respeito à flexibilidade, o sistema 3 é inferior, não só por estar limitado pela captura do Kinect, como também não ser possível alternar a configuração das mãos durante a gravação.

No contexto da LGP não se conseguem produzir gestos de qualidade, ainda assim, o sistema 2 foi superior na classificação que obteve através da avaliação.

8.5 Conclusão

A evolução feita ao PE2LGP revelou-se, de um modo geral, positiva. Todos os sistemas são úteis no contexto da LGP e de fácil utilização. Contudo, cada uma delas, ainda tem de evoluir para que possa ser transformado num produto final e ser usado num contexto profissional.

No sistema 1, o principal foco é melhorar a componente linguística. No sistema 2, é preciso dar uma maior liberdade ao utilizador do controlo com o modelo, bem como adicionar um conjunto de opções sugeridas nos questionários. No sistema 3, é necessário melhorar os comandos de voz, assim como adicionar a captura das mãos ao nível do detalhe das mãos e também uma melhor deteção das rotações dos pontos capturados pelo Kinect.

Quanto à qualidade dos gestos no contexto da LGP, os resultados além de negativos foram inconclusivos. Os avaliadores tinham de preencher um questionário onde era requerido que identificassem 5 gestos através de vídeos e posteriormente classificassem-nos. O grupo testado nesta parte da avaliação era composto por 6 pessoas, com diferentes níveis de LGP e de diferentes regiões. Existem vários fatores que poderão ter levado a esses resultados, como por exemplo, o nível de LGP ser baixo. Assim, terão de ser aprofundados esses fatores através de um outro tipo de avaliação e um grupo de focus maior.

9

Conclusão

Conteúdo

9.1 Dificuldades	85
9.2 Trabalho Futuro	86

Os objetivos deste trabalho focaram-se em duas principais vertentes: tornar o sistema mais flexível e agilizar o modo de criação de novos gestos.

Em primeiro lugar, destaca-se a introdução de um motor de jogo, o Unity. Esta modificação surge em virtude da necessidade de eliminar a dependência que existia ao Blender, isto porque qualquer modificação requeria sempre a intervenção no Blender. Deste modo, o projeto foi exportado para o Unity, dando assim uma maior flexibilidade e quebrando a dependência ao Blender. Além disso, foi criada uma REST API para fazer a comunicação com a componente linguística que também ficou independente do Blender. Essa componente sofreu algumas alterações, sendo que agora é possível ter controlo de cada uma das fases do processamento do texto. Uma vez que é independente, poderá ainda ser desenvolvida à parte do PE2LGP ou ligada a outro projeto.

Depois, numa tentativa de agilizar o modo de criação de novos dados, foram criados dois sistemas para criar gestos, no Unity. O primeiro sistema, manual, denominado sistema 2, permite ao utilizador criar novos gestos, interagindo diretamente com o modelo, usando para isso uma interface. O segundo, automático, denominado sistema 3, também permite a criação de gestos, mas utilizando o sensor Kinect One, automatizando assim todo o processo.

O sistema manual tem a vantagem ter vários níveis de liberdade na criação dos gestos, sobretudo ao nível das rotações. É possível fazer praticamente todos os movimentos e rotações utilizados nos gestos da LGP. No sistema automático, os movimentos, em particular as rotações dos braços, estão dependentes da captura do sensor Kinect, que na maior parte das vezes tem um grau elevado de erro, não na captura da posição, mas na rotação. Por outro lado, no sistema automático, a vantagem é o tempo de criação, que é muito reduzido em comparação com o sistema manual.

Inicialmente, estes sistemas, em particular o automático, foram pensados para serem independentes da mudança da arquitetura e do Unity. No entanto, de modo a maximizar a experiência do utilizador e a qualidade do projeto, foram criados e juntos no Unity. Assim sendo, existem 3 sistemas no Unity: a interface criada em virtude da importação do PE2LGP para o Unity, o sistema manual e o sistema automático. Além destes 3, ainda foi criado um outro para complementar, o “Visualizador de Animações” que, como o nome sugere, é uma interface para ver e/ou eliminar os gestos existentes.

Os objetivos definidos inicialmente foram todos cumpridos com sucesso. Mas, sendo este projeto, em particular os dois sistemas de criação de gestos, protótipos, requerem trabalho futuro, pelo que não produzem o melhor resultado possível nem são um produto final.

9.1 Dificuldades

As maiores dificuldades foram sobretudo em ultrapassar problemas encontrados durante a criação dos dois sistemas de criação de animações, com ênfase na criação da própria animação.

O Unity é conhecido como um motor de jogo, onde geralmente não se fazem animações nem modelações, estas são feitas recorrendo a outros programas, tais como o Autodesk 3DS Max ou o Blender. No entanto, o Unity contém uma componente de criação de animações, mas que a maioria dos desenvolvedores, ou desconhece, ou simplesmente não utiliza, recorrendo aos programas exter-

nos. Deste modo, a documentação e a ajuda na resolução de problemas neste contexto é reduzida, pelo que a destaco como tendo sido o maior desafio neste trabalho.

9.2 Trabalho Futuro

Tal como foi dito anteriormente, tanto o PE2LGP original, são protótipos, e como tal requerem trabalho futuro. Desta forma, destaco os seguintes objetivos:

1. Criar de um sistema semelhante ao sistema 2, manual, mas dedicado à criação e edição de configurações de mãos.
2. Guardar num servidor as animações criadas pelo sistema manual e automático. É necessário um mecanismo para enviar e receber pedidos do servidor, algo semelhante ao que foi feito com a *REST API* da componente linguística.
3. Elaborar a componente linguística. É necessário um estudo aprofundado da língua, para que depois se possam extrair todas as informações e aplicá-las no PE2LGP.
4. Exportar para web e dispositivos móveis.
5. Implementar e, posteriormente, melhorar a captura das mãos através do Kinect.
6. Melhorar as rotações dos pontos capturados pelo Kinect no sistema automático.
7. Integrar o sensor Leap Motion no PE2LGP e utilizá-lo em conjunto com o Kinect no âmbito do sistema automático.
8. Utilizar o sistema manual como base para a criação de um outro que converte texto em animações. O texto é semelhante às descrições presentes no Dicionário de Língua Gestual. Por exemplo, “mão dominante em configuração 9 que toca duas vezes junto ao pescoço do mesmo lado” seria automaticamente convertido para uma animação.
9. Implementar a edição de animações, frame por frame. O utilizador pode criar uma animação no sistema automático e depois ajustá-la manualmente através do sistema manual. Além disso, esta edição deverá ser incluída durante a criação dos frames no sistema manual.
10. Integrar expressões faciais, utilizando, por exemplo, a ajuda do sensor Kinect e dos recursos mencionados no trabalho relacionado.
11. Adicionar mais níveis de rotação no sistema 2. Por exemplo, ao movimentar o pulso, poderá também mover-se o cotovelo e o ombro.
12. Exportar os gestos de qualquer palavra ou frase para vídeo.

Salientando que já foram realizadas experiências com o sensor Leap Motion, inclusive com a última atualização, a Orion. A captura do sensor por omissão utilizando as bibliotecas disponibilizadas pelos desenvolvedores sofreu várias melhorias, mas dedicadas à realidade virtual, o que requer

um uso diferente (no que diz respeito à disposição) do suposto neste projeto. Assim sendo, a captura é razoável, mas não suficiente para abranger gestos na [LGP](#).

Também foi testado o uso do Kinect e do Leap Motion em simultâneo, com ênfase na disposição dos sensores. Uma delas é o utilizador poder estar sentado e os dispositivos à sua frente, para que seja feita a captura por ambos os sensores.

O sugerido é que sejam captadas as mãos com ambos os sensores, utilizando as bibliotecas por omissão e também através de algoritmos (como por exemplo vários dos mencionados no trabalho relacionado), e posteriormente aplicar o resultado ao sistema automático, o que requer um mapeamento semelhante ao já criado, mas dedicado às mãos.

Bibliografia

- [Jordi Adell and Linda Johanna Castañeda Quintero (2012)]. Tecnologías emergentes, pedagogías emergentes.
- [Susana Afonso, Eckhard Bick, Renato Haber, and Diana Santos (2002)]. Floresta sintática: A trebank for portuguese. URL <http://www.linguateca.pt/Diana/download/AfonsoetalLREC2002.pdf>.
- [Inês Almeida, Luísa Coheur, and Sara Candeias (2015)]. Coupling natural language processing and animation synthesis in portuguese sign language translation. In *Proceedings of the Fourth Workshop on Vision and Language*, pages 94–103, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. URL <http://aclweb.org/anthology/W15-2815>.
- [Inês Almeida, Luísa Coheur, and Sara Candeias (2015)]. From european portuguese to portuguese sign language. In *6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, page 140, 2015b.
- [Inês Almeida (2014)]. Exploring Challenges in Avatar-based Translation from European Portuguese to Portuguese Sign Language. URL <https://fenix.tecnico.ulisboa.pt/downloadFile/844820067123782/shorttv.pdf>.
- [Maria Augusta Amaral, Amândio Coutinho, Maria Raquel Delgado Martins, and Robert Johnson (1994)]. *Para uma gramática da língua gestual portuguesa*.
- [Vassilis Athitsos and Stan Sclaroff (2003)]. Estimating 3D hand pose from a cluttered image. *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2003 Proceedings*, 2:II–432–9, 2003. ISSN 10636919. doi: 10.1109/CVPR.2003.1211500.
- [Davide Barberis, Nicola Garazzino, Paolo Prinetto, and Gabriele Tiotto (2011)]. Improving accessibility for deaf people. *Assets (Xiii)*, page 253, 2011. doi: 10.1145/2049536.2049593. URL <http://dl.acm.org/citation.cfm?doid=2049536.2049593>.
- [José Bento, Ana Paula Claudio, and Paulo Urbano (2014)]. Avatars on Portuguese sign language. *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, 2014. ISSN 21660735. doi: 10.1109/CISTI.2014.6876959. URL <http://www.scopus.com/inward/record.url?eid=2-s2.0-84906692159&partnerID=tZ0tx3y1>.

- [Wisarut Bholsithi, Nonlapas Wongwaen, and Chanjira Sinthanayothin (2014)].** 3d avatar developments in real time and accuracy assessments. In *2014 International Computer Science and Engineering Conference*. Institute of Electrical & Electronics Engineers (IEEE), jul 2014. doi: 10.1109/icsec.2014.6978174. URL <http://dx.doi.org/10.1109/ICSEC.2014.6978174>.
- [Antoinette Bruciati and Colette Rossignol (2015)].** Using mocap & chromakey special effects for increasing instructor presence in an online course. *Learning*, 11:45am, 2015.
- [Jacky CP Chan, Howard Leung, Jeff KT Tang, and Taku Komura (2011)].** A virtual reality dance training system using motion capture technology. *IEEE Transactions on Learning Technologies*, 4 (2):187–195, 2011.
- [Sun-Wook Choi, Woong-Ji Kim, and Chong Ho Lee (2013)].** Interactive display robot: projector robot with natural user interface. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pages 109–110. IEEE Press, 2013.
- [Pedro Costa, Hooshar Zolfagharnasab, João P Monteiro, Jaime S Cardoso, and Hélder P Oliveira (2014)].** 3d reconstruction of body parts using rgb-d sensors: challenges from a biomedical perspective. In *Proceedings of the 5th International Conference on 3D Body Scanning Technologies*. Citeseer, 2014.
- [Maria Fernanda da Silva Bettencourt (2015)].** A ordem das palavras na língua gestual portuguesa: Breve estudo comparativo com o português e outras línguas gestuais. Faculdade de Letras da Universidade do Porto.
- [Samarjit Das, Laura Trutoiu, Akihiko Murai, Dunbar Alcindor, Michael Oh, Fernando De la Torre, and Jessica Hodgins (2011)].** Quantitative measurement of motor symptoms in parkinson's disease: A study with full-body motion capture data. In *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6789–6792. IEEE, 2011.
- [Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly (2007)].** Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding*, 108 (1-2):52–73, 2007. ISSN 10773142. doi: 10.1016/j.cviu.2006.10.012.
- [Wenjuan Gong, Zhichao Wang, and Weishan Zhang (2015)].** Creating personal 3d avatar from a single depth sensor. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1193–1198. IEEE, 2015.
- [Lorna Herda, Pascal Fua, Ralf Plänkers, Ronan Boulic, and Daniel Thalmann (2001)].** Using skeleton-based tracking to increase the reliability of optical motion capture. *Human movement science*, 20(3):313–341, 2001.

- [Paramat Ittisam and Nunnapad Toaditthep (2010)].** 3D Animation Editor and Display Sign Language System case study: Thai Sign Language. *2010 3rd International Conference on Computer Science and Information Technology*, 4:633–637, 2010. doi: 10.1109/ICCSIT.2010.5564559. URL <http://www.scopus.com/inward/record.url?eid=2-s2.0-77958604747&partnerID=tZ0tx3y1>.
- [Mohamed Jemni, Sameer Semreen, Achraf Othman, Zouhour Tmar, and Nadia Aouiti (2013)].** Toward the creation of an Arab Gloss for arabic Sign Language annotation. *Fourth International Conference on Information and Communication Technology and Accessibility (ICTA)*, pages 1–5, 2013. doi: 10.1109/ICTA.2013.6815292. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6815292>.
- [Lynette A. Jones and Susan J. Lederman (2006)].** *Human Hand Function*. Oxford University Press, 2006. ISBN 0195173155. URL <http://www.amazon.com/Human-Hand-Function-Lynette-Jones/dp/0195173155%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0195173155>.
- [Jestin Joy and Kannan Balakrishnan (2014)].** A prototype Malayalam to Sign Language Automatic Translator. URL <http://arxiv.org/abs/1412.7415>.
- [J Kennedy and R Eberhart (1995)].** Particle swarm optimization. *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 4:1942–1948 vol.4, 1995. ISSN 19353812. doi: 10.1109/ICNN.1995.488968.
- [Cem Keskin, Furkan Kiraç, Yunus Emre Kara, and Lale Akarun (2012)].** Hand pose estimation and hand shape classification using multi-layered randomized decision forests. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7577 LNCS(PART 6):852–863, 2012. ISSN 03029743. doi: 10.1007/978-3-642-33783-3{_}61.
- [Michael Kipp, Alexis Heloir, and Quan Nguyen (2011)].** Sign language avatars: Animation and comprehensibility. In *Intelligent Virtual Agents*, pages 113–126. Springer Science Business Media, 2011. doi: 10.1007/978-3-642-23974-8_13. URL http://dx.doi.org/10.1007/978-3-642-23974-8_13.
- [Despoina Kotsidou, Constantinos Angelis, Stamatis Dragoumanos, and Athanasios Kakaountas (2015)].** Computer assisted gesture recognition for the greek sign language / fingerspelling. In *Proceedings of the 19th Panhellenic Conference on Informatics*. Association for Computing Machinery (ACM), 2015. doi: 10.1145/2801948.2802034. URL <http://dx.doi.org/10.1145/2801948.2802034>.
- [Philip Krejov, Andrew Gilbert, and Richard Bowden (2015)].** Combining Discriminative and Model Based Approaches for Hand Pose Estimation. doi: 10.1109/FG.2015.7163141.

- [Luís Leite and Veronica Orvalho (2013)].** Inter-acting: Understanding interaction with performance-driven puppets using low-cost optical motion capture device. *Journal International Journal of Advanced Computer Science*, 3(2):65–69, 2013.
- [Scott K. Liddell (2003)].** *Grammar, Gesture, and Meaning in American Sign Language*. Cambridge University Press, 2003. ISBN 0521016509. URL <http://www.amazon.com/Grammar-Gesture-Meaning-American-Language/dp/0521016509%3FSubscriptionId%3D0JYN1NVW651KCA56C102%26tag%3Dtechkie-20%26linkCode%3Dxm2%26camp%3D2025%26creative%3D165953%26creativeASIN%3D0521016509>.
- [JL Liu, HC Chuan, and PC Kuan (2014)].** Assessment of range of shoulder motion using kinect. *Gerontechnology*, 13(2):249, 2014.
- [Jean-Luc Lugin, David Zilch, Daniel Roth, Gary Bente, and Marc Erich Latoschik (2016)].** Facebo: Real-time face and body tracking for faithful avatar synthesis. In *2016 IEEE Virtual Reality (VR)*, pages 225–226. IEEE, 2016.
- [David McNeill and Susan D. Duncan (2005)].** Grammar, Gesture, and Meaning in American Sign Language (review). *Sign Language Studies*, 5(4):506–523, 2005. ISSN 1533-6263. doi: 10.1353/sls.2005.0016.
- [Alberto Menache (2000)].** *Understanding motion capture for computer animation and video games*. Morgan kaufmann, 2000.
- [Iason Oikonomidis, Nikolaos Kyriazis, and Antonis a Argyros (2011)].** Efficient Model-based 3D Tracking of Hand Articulations using Kinect. *22nd British Machine Vision Conference*, pages 1–11, 2011. doi: 10.5244/C.25.101.
- [Silvia Cristina Martini Campinas Ossada, Sandra Aparecida Ribeiro Rodrigues (2015)].** Uma análise de softwares para inclusão de deficientes auditivos na educação à distância. pages 30–37. Faculdade de Tecnologia de Campinas e Roccato, Universidade de Mogi das Cruzes, 2015.
- [Ayberk Özgür, Stéphane Bonardi, Massimo Vespignani, Rico Möckel, and Auke J Ijspeert (2014)].** Natural user interface for roombots. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, pages 12–17. IEEE, 2014.
- [Juan Alejandro Peña Palacio and Santiago Londoño Arévalo (2014)].** Remote control flying rc helicopter through natural user interface. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)*, pages 1–6. IEEE, 2014.
- [Alejandro Lopez Rincon, Hiroshi Yamasaki, and Shingo Shimoda (2016)].** Design of a video game for rehabilitation using motion capture, emg analysis and virtual reality. In *2016 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 198–204. IEEE, 2016.

[Stefano Scheggi, Leonardo Meli, Claudio Pacchierotti, and Domenico Prattichizzo (2015)].

Touch the virtual reality. Association for Computing Machinery (ACM), 2015. doi: 10.1145/2787626.2792651. URL <http://dx.doi.org/10.1145/2787626.2792651>.

[Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel (2015)]

Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, robust, and flexible real-time hand tracking. URL <http://research.microsoft.com/apps/pubs/default.aspx?id=238453>.

[Merkourios Simos and Nikolaos Nikolaidis (2016)].

Greek sign language alphabet recognition using the leap motion device. In *Proceedings of the 9th Hellenic Conference on Artificial Intelligence - SETN*. Association for Computing Machinery (ACM), 2016. doi: 10.1145/2903220.2903249. URL <http://dx.doi.org/10.1145/2903220.2903249>.

[Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt (2013)].

Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. *2013 IEEE International Conference on Computer Vision*, pages 2456–2463, 2013. ISSN 1550-5499. doi: 10.1109/ICCV.2013.305. URL [http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6751416\\$delimiter"026E30F\\$nhhttp://www.cv-foundation.org/openaccess/content_{_}iccv_{_}2013/papers/Sridhar_{_}Interactive_{_}Markerless_{_}Articulated_{_}2013_{_}ICCV_{_}paper.pdf](http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6751416$delimiter).

[Andrea Tagliasacchi, Matthias Schröder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly (2015)].

Robust articulated-icp for real-time hand tracking. *Symposium on Geometry Processing (Computer Graphics Forum)*, 2015.

[Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan (2012)].

Scanning 3d full human bodies using kinects. *IEEE transactions on visualization and computer graphics*, 18(4):643–650, 2012.

[Shuangyuan Wu, Shihong Xia, Zhaoqi Wang, and Chunpeng Li (2009)].

Efficient motion data indexing and retrieval with local similarity measure of motion strings. *The Visual Computer*, 25(5-7): 499–508, 2009.

[Qiang Zhang and Xiaocui Guo (2013)].

3d human motion retrieval based on human hierarchical index structure. *Biology of Sport*, 30:145–151, 2013.



Anexos

A.1 Questionários

Nesta seção serão apresentados todos os questionários utilizados na avaliação dos 3 sistemas descritos no [Capítulo 4](#), no [Capítulo 5](#) e no [Capítulo 6](#). Os questionários incluem não só os modelos, como também todos os resultados obtidos preenchidos pelos próprios utilizadores.

Nota: Caso esteja a visualizar a versão impressa deste trabalho, recomenda-se, para uma melhor percepção das imagens, o uso da versão digital em formato PDF.

A.1.1 Modelo do Questionário do Sistema 1

A [Figura A.11](#), a [Figura A.9](#), a [Figura A.28](#) e a [Figura A.25](#), representam o modelo utilizado no questionário de avaliação do primeiro sistema.

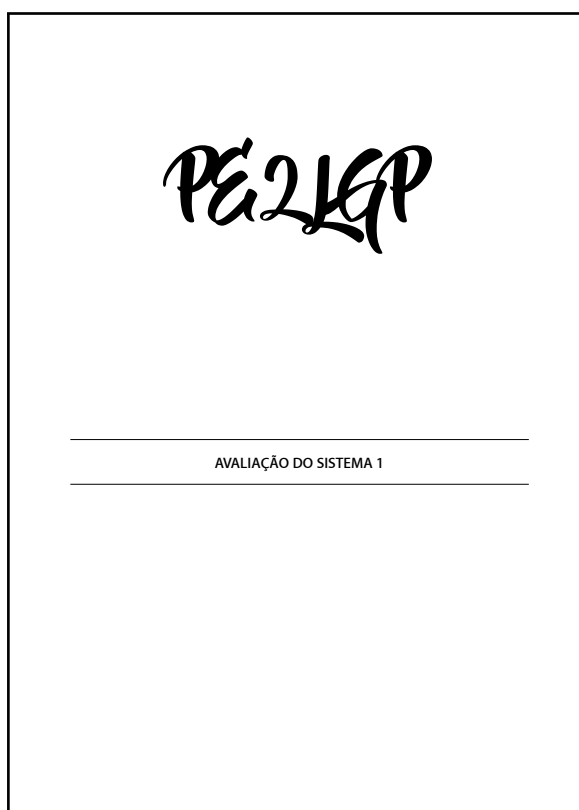


Figura A.1: Modelo do Questionário 1 (Capa).

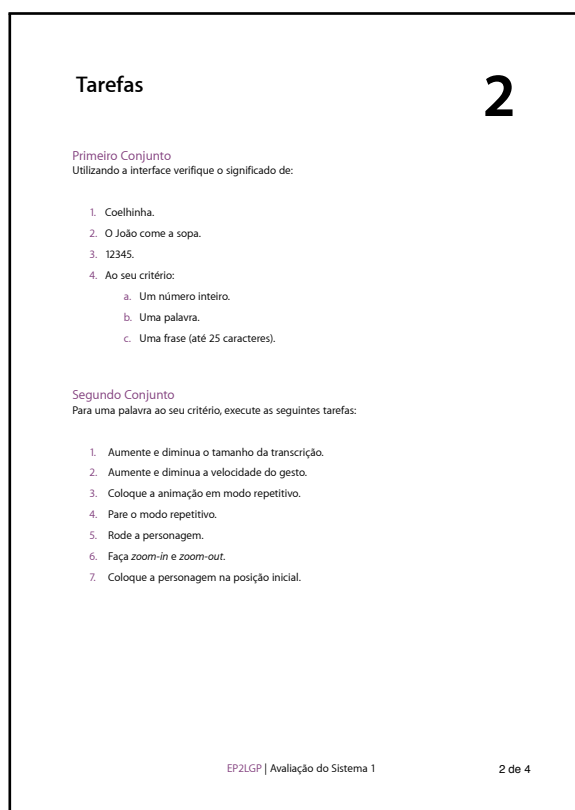


Figura A.2: Modelo do Questionário 1 (2 de 4).

Salientado que o modelo da capa da [Figura A.11](#), é utilizada em todos os questionários. Assim, esta figura não será repetida nas próximas seções.

Questionário

3

1. Conseguiu perceber a transcrição de todos os gestos?

☐ Sim.

☐ Não. Porquê? _____

2. O tamanho da transcrição é adequado?

☐ Sim.

☐ Não, aumentava.

☐ Não, diminuía.

3. A velocidade dos gestos é adequada?

☐ Sim.

☐ Não. Porquê? _____

4. Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

6. Adicionava mais opções?

☐ Sim. _____

☐ Não.

7. Removia alguma opção?

☐ Sim. _____

☐ Não.

8. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1

☐ 2

☐ 3

☐ 4

☐ 5

EP2LGPT | Avaliação do Sistema 1

3 de 4

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?

☐ Sim.

☐ Não.

10. Tem alguma ideia para melhorar o sistema?

☐ Sim. _____

☐ Não.

Dados Pessoais:

1. Primeiro e último nome:

2. Idade:

3. Sexo:

4. Ocupação:

Figura A.3: Modelo do Questionário 1 (3 de 4).

Figura A.4: Modelo do Questionário 1 (4 de 4).

A.1.2 Modelo do Questionário 1 dos Sistemas 2 e 3

A [Figura A.5](#), a [Figura A.6](#) e a [Figura A.7](#), representam o modelo utilizado no questionário das avaliações dos sistemas 2 e 3.

Tarefas

2

O objetivo das tarefas é, a partir do sistema descrito anteriormente, criar 5 gestos. Os gestos podem ser consultados no Dicionário de Língua Gestual ou então através da página: <https://www.spreadthesign.com/pt/>.

Criação de cinco gestos.
Crie os seguintes gestos:

1. Baixo
Dicionário: página 93
SpreadTheSign: <https://media.spreadthesign.com/video/mp4/6/304870.mp4>.
2. Hoje
Dicionário: página 434
SpreadTheSign: <https://media.spreadthesign.com/video/mp4/6/4201.mp4>.
3. Mal
Dicionário: página 599
SpreadTheSign: <https://media.spreadthesign.com/video/mp4/6/5381.mp4>.
4. Noite
Dicionário: página 700
SpreadTheSign: <https://media.spreadthesign.com/video/mp4/6/16558.mp4>.
5. Pagar
Dicionário: página 759
SpreadTheSign: <https://media.spreadthesign.com/video/mp4/6/16655.mp4>.

Nota: Deve usar o formato **NOME_DO_GESTO + INDICADOR**, por exemplo, Dia1, sendo que o valor do **INDICADOR** ser-lhe-á dado.

PE2LGP | Avaliação do Sistema 2

2 de 4

Questionário

3

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
2. Conseguiu criar todos os gestos?
☐ Sim. ☐ Não. Porquê? _____
3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
4. Percebeu como funcionava todo o sistema?
☐ Sim. ☐ Não. Porquê? _____
5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
6. Acha as opções disponibilizadas adequadas?
☐ Sim. ☐ Não. Porquê? _____
7. Adicionava mais opções?
☐ Sim. _____ ☐ Não.
8. Acha o sistema útil para a Língua Gestual Portuguesa?
☐ Sim. ☐ Não.

PE2LGP | Avaliação do Sistema 2

3 de 4

Figura A.5: Modelo do Questionário 2 e 3 (2 de 4).

Figura A.6: Modelo do Questionário 2 e 3 (3 de 4).

9. Tem alguma ideia para melhorar o sistema?
☐ Sim. _____ ☐ Não.

Dados Pessoais:

1. Primeiro e último nome:

2. Idade:

3. Sexo:

4. Ocupação:

PE2LGP | Avaliação do Sistema 2

4 de 4


Figura A.7: Modelo do Questionário 2 e 3 (4 de 4).

A.1.3 Modelo e Resultados do Questionário 2 dos Sistemas 2 e 3

02/10/2016 PEZLGP-QUIZ-SYSTEM02 - Google Forms

PEZLGP

O PEZLGP é um protótipo desenvolvido no L2F/INESC-ID, que tem como objectivo traduzir, em tempo real, frases de Português Europeu em Língua Gestual Portuguesa, através de um avatar (a Catarina). No quadro de uma tese de mestrado, pretende-se avaliar quão perceptíveis e precisos são cinco gestos deste avatar. Desde já obrigado pelo seu feedback.



Para mais informações, utilize os seguintes contactos:
ruben.d.santos@tecnico.ulisboa.pt ou luisa.coheur@tecnico.ulisboa.pt

https://docs.google.com/forms/d/1Gv-G3pXOvv4k1rQuOdMj--lMkLuisGZr8BgYy04Ba_ored#responses 1/35

Figura A.8: Modelo do Questionário 4 (Capa).

02/10/2016 PEZLGP-QUIZ-SYSTEM02 - Google Forms

PEZLGP

Dados Pessoais

Primeiro e último nome:
Ana mineiro

Idade:
50

Ocupação:
professora

Região onde vive:
Lisboa

Qual é o seu nível de Língua Gestual? *

☐ Inicial (A1-A2)

☒ Intermediário (B1-B2)

☐ Avançado (C1-C2)

☐ Other:

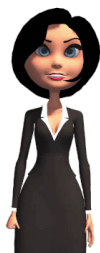
https://docs.google.com/forms/d/1Gv-G3pXOvv4k1rQuOdMj--lMkLuisGZr8BgYy04Ba_ored#responses 2/35

Figura A.9: Seção 1 de 3 do Questionário 4.

02/10/2016 PEZLGP-QUIZ-SYSTEM02 - Google Forms

Sugestões e/ou críticas (Opcional):

3



Penso que o gesto é:

Tendo em conta a palavra que identificou, quão preciso considera o gesto executado pela Catarina? *

0-Não Reconheço

Sugestões e/ou críticas (Opcional):

https://docs.google.com/forms/d/1Gv-G3pXOvv4k1rQuOdMj--lMkLuisGZr8BgYy04Ba_ored#responses 5/35

Figura A.10: Seção 2 de 3 do Questionário 4.

02/10/2016 PEZLGP-QUIZ-SYSTEM02 - Google Forms

PEZLGP

Dados Pessoais

Primeiro e último nome:
Ana

Idade:
Silva

Ocupação:
Psicóloga

Região onde vive:
Lisboa

Qual é o seu nível de Língua Gestual? *

☐ Inicial (A1-A2)

☒ Intermediário (B1-B2)

☐ Avançado (C1-C2)

☐ Other:

https://docs.google.com/forms/d/1Gv-G3pXOvv4k1rQuOdMj--lMkLuisGZr8BgYy04Ba_ored#responses 9/35

Figura A.11: Seção 3 de 3 do Questionário 4.

Os restantes questionários foram omitidos devido à sua extensão e repetição.

A.1.4 Resultados dos Questionários do Sistema 1

Questionário

3

1. Conseguiu perceber a transcrição de todos os gestos?

☒ Sim ☐ Não. Porque? _____

2. O tamanho da transcrição é adequado?

☒ Sim ☐ Não, aumentava ☐ Não, diminuía

3. A velocidade dos gestos é adequada?

☒ Sim ☐ Não. Porque? _____

4. Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

6. Adicionava mais opções?

☒ Sim. Mantive a opção na caixa de texto para ser possível editar logo. ☐ Não.

7. Removia alguma opção?

☐ Sim. _____ ☒ Não.

8. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?

☒ Sim ☐ Não.

10. Tem alguma ideia para melhorar o sistema?

☐ Sim. _____ ☒ Não.

Dados Pessoais:

1. Primeiro e último nome:

Rui Santos

2. Idade:

22

3. Sexo:

Masculino

4. Ocupação:

Estudante

EP2LGP | Avaliação do Sistema 13 de 4

Figura A.12: Resultado do Questionário do Sistema 1 - Pessoa 1 de 6.

Questionário

3

1. Conseguiu perceber a transcrição de todos os gestos?

☒ Sim ☐ Não. Porque? _____

2. O tamanho da transcrição é adequado?

☒ Sim ☐ Não, aumentava ☐ Não, diminuía

3. A velocidade dos gestos é adequada?

☒ Sim ☐ Não. Porque? _____

4. Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

6. Adicionava mais opções?

☐ Sim. _____ ☒ Não.

7. Removia alguma opção?

☐ Sim. _____ ☒ Não.

8. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?

☒ Sim ☐ Não.

10. Tem alguma ideia para melhorar o sistema?

☐ Sim. _____ ☒ Não.

Dados Pessoais:

1. Primeiro e último nome:

Soni Dias

2. Idade:

22

3. Sexo:

M

4. Ocupação:

Estudante

EP2LGP | Avaliação do Sistema 13 de 4

Figura A.13: Resultado do Questionário do Sistema 1 - Pessoa 2 de 6.

Questionário

3

- Conseguiu perceber a transcrição de todos os gestos?
☒ Sim. ☐ Não. Porque? _____
- O tamanho da transcrição é adequado?
☒ Sim. ☐ Não, aumentava. ☐ Não, diminuía.
- A velocidade dos gestos é adequada?
☒ Sim. ☐ Não. Porque? _____
- Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil
☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
- Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil
☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
- Adicionava mais opções?
☐ Sim. _____ ☒ Não.
- Removia alguma opção?
☐ Sim. _____ ☒ Não.
- De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?
☒ Sim. ☐ Não.

10. Tem alguma ideia para melhorar o sistema?
☐ Sim. _____ ☒ Não.

Dados Pessoais:

- Primeiro e último nome:
Vânia Rendonça
- Idade:
25
- Sexo:
F
- Ocupação:
Estudante, investigadora

EP2LGP | Avaliação do Sistema 1

3 de 4

Figura A.14: Resultado do Questionário do Sistema 1 - Pessoa 3 de 6.

Questionário

3

- Conseguiu perceber a transcrição de todos os gestos?
☒ Sim. ☐ Não. Porque? _____
- O tamanho da transcrição é adequado?
☒ Sim. ☐ Não, aumentava. ☐ Não, diminuía.
- A velocidade dos gestos é adequada?
☒ Sim. ☐ Não. Porque? _____
- Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5
- Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil
☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
- Adicionava mais opções?
☒ Sim. Aproximação do texto ao avatar ☐ Não.
- Removia alguma opção?
☐ Sim. _____ ☒ Não.
- De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?
☒ Sim. ☐ Não.

10. Tem alguma ideia para melhorar o sistema?
☐ Sim. _____ ☒ Não.

Dados Pessoais:

- Primeiro e último nome:
Paulo Silva
- Idade:
19
- Sexo:
Masculino
- Ocupação:
Segurança

EP2LGP | Avaliação do Sistema 1

3 de 4

Figura A.15: Resultado do Questionário do Sistema 1 - Pessoa 4 de 6.

Questionário

3

1. Conseguiu perceber a transcrição de todos os gestos?

☒ Sim. ☐ Não. Porque? _____

2. O tamanho da transcrição é adequado?

☒ Sim. ☐ Não, aumentava. ☐ Não, diminuía.

3. A velocidade dos gestos é adequada?

☒ Sim. ☐ Não. Porque? A Escolha

4. Qual a dificuldade das tarefas executadas? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

6. Adicionava mais opções?

☒ Sim. touch screen ☐ Não.

7. Removia alguma opção?

☐ Sim. _____ ☒ Não.

8. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?

☒ Sim. ☐ Não. Necessário Ampliar Dic.

10. Tem alguma ideia para melhorar o sistema?

☒ Sim. Adequar sintático ☐ Não. SA p/ Linguista

Dados Pessoais:

1. Primeiro e último nome: _____

2. Idade: _____

3. Sexo: _____

4. Ocupação: _____

EP2LGP | Avaliação do Sistema 1

3 de 4

Figura A.16: Resultado do Questionário do Sistema 1 - Pessoa 5 de 6.

Questionário

3

1. Conseguiu perceber a transcrição de todos os gestos?

☒ Sim. ☐ Não. Porque? _____

2. O tamanho da transcrição é adequado?

☒ Sim. ☐ Não, aumentava. ☐ Não, diminuía.

3. A velocidade dos gestos é adequada?

☒ Sim. ☐ Não. Porque? _____

4. Qual a dificuldade das tarefas executadas? 1 - Difícil, 5 - Fácil

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

5. A interface é de fácil utilização? 1 - Difícil, 5 - Fácil

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

6. Adicionava mais opções?

☒ Sim. Um botão para iniciar a animação ☐ Não.

7. Removia alguma opção?

☐ Sim. _____ ☒ Não.

8. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

9. Acha o sistema útil para aprender Língua Gestual Portuguesa?

☒ Sim. ☐ Não.

10. Tem alguma ideia para melhorar o sistema?

☒ Sim. Fazer aparecer o button stop ao momento da animação estar a decorrer, e desaparecer quando a animação acaba. ☐ Não.

Dados Pessoais:

1. Primeiro e último nome: Volodymyr Polytsky

2. Idade: 25

3. Sexo: Masculino

4. Ocupação: Web Developer

EP2LGP | Avaliação do Sistema 1

3 de 4

Figura A.17: Resultado do Questionário do Sistema 1 - Pessoa 6 de 6.

A.1.5 Resultados dos Questionários do Sistema 2

Questionário

3

9. Tem alguma ideia para melhorar o sistema?

☐ Sim ☒ Não

Dados Pessoais:

1. Primeiro e último nome: Thi. Santos

2. Idade: 22

3. Sexo: Masculino

4. Ocupação: Estudante

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim ☐ Não. Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim ☐ Não. Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim ☐ Não. Porquê? _____

7. Adicionava mais opções?

☐ Sim ☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim ☐ Não.

PEZLGP | Avaliação do Sistema 23 de 4PEZLGP | Avaliação do Sistema 24 de 4

Figura A.18: Resultado do Questionário do Sistema 2 - Pessoa 1 de 6.

Questionário

3

9. Tem alguma ideia para melhorar o sistema?

☐ Sim ☒ Não

Dados Pessoais:

1. Primeiro e último nome: Javi Dias

2. Idade: 22

3. Sexo: M

4. Ocupação: Estudante

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

2. Conseguiu criar todos os gestos?

☒ Sim ☐ Não. Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim ☐ Não. Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim ☐ Não. Porquê? _____

7. Adicionava mais opções?

☐ Sim ☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim ☐ Não.

PEZLGP | Avaliação do Sistema 23 de 4PEZLGP | Avaliação do Sistema 24 de 4

Figura A.19: Resultado do Questionário do Sistema 2 - Pessoa 2 de 6.

Questionário

3

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim ☐ Não. Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim ☐ Não. Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim ☐ Não. Porquê? _____

7. Adicionava mais opções?

☐ Sim _____ ☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim ☐ Não.

9. Tem alguma ideia para melhorar o sistema?

☐ Sim _____ ☒ Não.

Dados Pessoais:

1. Primeiro e último nome: _____

2. Idade: _____

3. Sexo: _____

4. Ocupação: _____

PEZLGP | Avaliação do Sistema 2

3 de 4

PEZLGP | Avaliação do Sistema 2

4 de 4

Figura A.20: Resultado do Questionário do Sistema 2 - Pessoa 3 de 6.

Questionário

3

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim ☐ Não. Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim ☐ Não. Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim ☐ Não. Porquê? _____

7. Adicionava mais opções?

☐ Sim _____ ☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim ☐ Não.

9. Tem alguma ideia para melhorar o sistema?

☐ Sim _____ ☒ Não.

Dados Pessoais:

1. Primeiro e último nome: Paula Silva

2. Idade: 19

3. Sexo: Feminina

4. Ocupação: Letragrãfica

PEZLGP | Avaliação do Sistema 2

3 de 4

PEZLGP | Avaliação do Sistema 2

4 de 4

Figura A.21: Resultado do Questionário do Sistema 2 - Pessoa 4 de 6.

Questionário **3**

9. Tem alguma ideia para melhorar o sistema? ☒ Sim ☐ Não

Dados Pessoais:

1. Primeiro e último nome: T.A. / Nê Nunes

2. Idade: 33

3. Sexo: Mps cl. no

4. Ocupação: Desbrando

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?
☒ Sim ☐ Não Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Percebeu como funcionava todo o sistema?
☒ Sim ☐ Não Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

6. Acha as opções disponibilizadas adequadas?
☒ Sim ☐ Não Porquê? _____

7. Adicionava mais opções?
☒ Sim ☐ Não Porquê? _____

8. Acha o sistema útil para a Língua Gestual Portuguesa?
☒ Sim ☐ Não

PEZLGP | Avaliação do Sistema 2 3 de 4

Desaturar o (X) do pun 6.
No escolha das CM.
Uma imagem com todas as CMs da LGP e escolha a correta.
Visão/visualização do todo /
Tela de escolha: deixar igua fig. 5
Fica melhor
opção clicar e arrastar as junções
algas opções de pontuação (seg. Bastião analógico)

Figura A.22: Resultado do Questionário do Sistema 2 - Pessoa 5 de 6.

Questionário **3**

9. Tem alguma ideia para melhorar o sistema? ☒ Sim ☐ Não

Dados Pessoais:

1. Primeiro e último nome: Vladyslava Balytskyi

2. Idade: 25

3. Sexo: Masculino

4. Ocupação: Web Developer

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?
☒ Sim ☐ Não Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Percebeu como funcionava todo o sistema?
☒ Sim ☐ Não Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Mau, 5 - Bom
☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

6. Acha as opções disponibilizadas adequadas?
☒ Sim ☐ Não Porquê? _____

7. Adicionava mais opções?
☒ Sim ☐ Não Porquê? eliminar alguns indicadores

8. Acha o sistema útil para a Língua Gestual Portuguesa?
☒ Sim ☐ Não

PEZLGP | Avaliação do Sistema 2 3 de 4

Identificar o número de vezes que se pode clicar no máximo.
Botão STOP deveria estar visível enquanto a animação não decorrer.

Figura A.23: Resultado do Questionário do Sistema 2 - Pessoa 6 de 6.

A.1.6 Resultados dos Questionários do Sistema 3

Questionário		3	
<p>1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5</p> <p>2. Conseguiu criar todos os gestos?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não. Porquê? _____</p> <p>3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil</p> <p><input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>4. Percebeu como funcionava todo o sistema?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não. Porquê? _____</p> <p>5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>6. Acha as opções disponibilizadas adequadas?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não. Porquê? _____</p> <p>7. Adicionava mais opções?</p> <p><input type="radio"/> Sim. _____ <input checked="" type="radio"/> Não.</p> <p>8. Acha o sistema útil para a Língua Gestual Portuguesa?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não.</p>		<p>9. Tem alguma ideia para melhorar o sistema?</p> <p><input type="radio"/> Sim. _____ <input checked="" type="radio"/> Não.</p> <p>Dados Pessoais:</p> <p>1. Primeiro e último nome: <u>Felix Santos</u></p> <p>2. Idade: <u>22</u></p> <p>3. Sexo: <u>Masculino</u></p> <p>4. Ocupação: <u>Estudante</u></p>	
EP2LGP Avaliação do Sistema 2		3 de 4	
EP2LGP Avaliação do Sistema 2		4 de 4	

Figura A.24: Resultado do Questionário do Sistema 3 - Pessoa 1 de 5.

Questionário		3	
<p>1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>2. Conseguiu criar todos os gestos?</p> <p><input type="radio"/> Sim. <input checked="" type="radio"/> Não. Porquê? <u>Não conseguiu criar o gesto</u></p> <p>3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>4. Percebeu como funcionava todo o sistema?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não. Porquê? _____</p> <p>5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>6. Acha as opções disponibilizadas adequadas?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não. Porquê? _____</p> <p>7. Adicionava mais opções?</p> <p><input type="radio"/> Sim. _____ <input checked="" type="radio"/> Não.</p> <p>8. Acha o sistema útil para a Língua Gestual Portuguesa?</p> <p><input checked="" type="radio"/> Sim. <input type="radio"/> Não.</p>		<p>9. Tem alguma ideia para melhorar o sistema?</p> <p><input checked="" type="radio"/> Sim. <u>Uma ou mais melhorias</u> <input type="radio"/> Não.</p> <p>Dados Pessoais:</p> <p>1. Primeiro e último nome: <u>Jon Dias</u></p> <p>2. Idade: <u>22</u></p> <p>3. Sexo: <u>M</u></p> <p>4. Ocupação: <u>Estudante</u></p>	
EP2LGP Avaliação do Sistema 2		3 de 4	
EP2LGP Avaliação do Sistema 2		4 de 4	

Figura A.25: Resultado do Questionário do Sistema 3 - Pessoa 2 de 5.

Questionário

3

9. Tem alguma ideia para melhorar o sistema?

☐ Sim
☒ Não

Dados Pessoais:

1. Primeiro e último nome:

Paulo Silva

2. Idade:

19

3. Sexo:

Masculino

4. Ocupação:

Legislação

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1
☐ 2
☐ 3
☐ 4
☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim
☐ Não. Porquê?

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☒ 1
☐ 2
☐ 3
☐ 4
☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim
☐ Não. Porquê?

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1
☐ 2
☐ 3
☐ 4
☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim
☐ Não. Porquê?

7. Adicionava mais opções?

☐ Sim
☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim
☐ Não.

EP2LGP | Avaliação do Sistema 2

3 de 4

EP2LGP | Avaliação do Sistema 3

4 de 4

Figura A.26: Resultado do Questionário do Sistema 3 - Pessoa 3 de 5.

Questionário

3

9. Tem alguma ideia para melhorar o sistema?

☐ Sim
☒ Não.

Dados Pessoais:

1. Primeiro e último nome:

2. Idade:

3. Sexo:

4. Ocupação:

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1
☐ 2
☐ 3
☐ 4
☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim
☐ Não. Porquê?

3. Qual a dificuldade da criação dos gestos? 1 - Fácil, 5 - Difícil

☐ 1
☒ 2
☐ 3
☐ 4
☐ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim
☐ Não. Porquê?

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Fácil, 5 - Difícil

☒ 1
☐ 2
☐ 3
☐ 4
☐ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim
☐ Não. Porquê?

7. Adicionava mais opções?

☐ Sim
☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim
☐ Não.

EP2LGP | Avaliação do Sistema 2

3 de 4

EP2LGP | Avaliação do Sistema 2

4 de 4

Figura A.27: Resultado do Questionário do Sistema 3 - Pessoa 4 de 5.

Questionário

3

1. De um modo geral como avalia o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Conseguiu criar todos os gestos?

☒ Sim. ☐ Não. Porquê? _____

3. Qual a dificuldade da criação dos gestos? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Percebeu como funcionava todo o sistema?

☒ Sim. ☐ Não. Porquê? _____

5. Qual é a dificuldade na utilização de todo o sistema? 1 - Mau, 5 - Bom

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

6. Acha as opções disponibilizadas adequadas?

☒ Sim. ☐ Não. Porquê? _____

7. Adicionava mais opções?

☐ Sim. _____ ☒ Não.

8. Acha o sistema útil para a Língua Gestual Portuguesa?

☒ Sim. ☐ Não.

9. Tem alguma ideia para melhorar o sistema?

☒ Sim. Todos os conteúdos por ☐ Não.

All comments could be sent
Duration in possible content can be set. (e.g. finish, finish,
play, stop, ...)

Dados Pessoais:

1. Primeiro e último nome: Vladimir Bulsky

2. Idade: 25

3. Sexo: Masculino

4. Ocupação: Web Developer

EP2LGP | Avaliação do Sistema 2

3 de 4

EP2LGP | Avaliação do Sistema 2

4 de 4

Figura A.28: Resultado do Questionário do Sistema 3 - Pessoa 5 de 5.