



**PAULA MARGARIDA  
CONSTANTINO DIAS  
DE CARVALHO**

**INTERAÇÃO CONTROLADA POR GESTOS EM  
AMBIENTES 3D DISTRIBUÍDOS**



**PAULA MARGARIDA  
CONSTANTINO DIAS  
DE CARVALHO**

**INTERAÇÃO CONTROLADA POR GESTOS EM  
AMBIENTES 3D DISTRIBUÍDOS**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Comunicação Multimédia, realizada sob a orientação científica do Doutor Fernando Manuel dos Santos Ramos, Professor Catedrático do Departamento de Comunicação e Arte da Universidade de Aveiro e coorientação do Mestre Hélder Caixinha, Assistente Convidado do Departamento de Comunicação e Arte da Universidade de Aveiro.

Dedico este trabalho à minha família e amigos.

**o júri**

**presidente**

**Prof. Doutora Lúcia de Jesus Oliveira Loureiro da Silva**  
Professora Auxiliar com Agregação do Departamento de Comunicação e Arte da Universidade de Aveiro

**Prof. Doutor Carlos Sousa Pinto**  
Professor Auxiliar da Universidade do Minho

**Prof. Doutor Fernando Manuel dos Santos Ramos**  
Professor Catedrático do Departamento de Comunicação e Arte da Universidade de Aveiro

**Mestre Hélder José Marques Caixinha**  
Assistente Convidado do Departamento de Comunicação e Arte da Universidade de Aveiro

## **agradecimentos**

Agradeço aos meus orientador Professor Fernando Ramos e coorientador Mestre Hélder Caixinha pela orientação, disponibilidade e acompanhamento no decorrer desta investigação.

Agradeço à minha família, e em especial aos meus pais e irmã, pelo apoio incondicional, paciência e motivação.

Agradeço aos colaboradores da empresa Artshare pelo apoio no decorrer do projeto.

Agradeço aos meus colegas e amigos Samuel Almeida, Cátia Figueiredo e Mónica Aresta pelos incentivos e palavras encorajadoras.

Agradeço à Marta Pinto e Dalila Coelho pelo apoio, incentivo e disponibilidade.

Agradeço aos colegas da unidade de investigação CETAC.MEDIA pela boa disposição e apoio nos bons e menos bons momentos.

Agradeço ao Hugo Branquinho que esteve sempre presente.

Agradeço a todos os meus amigos que sempre me apoiaram, tiveram paciência, estiveram comigo e me deram sempre os melhores conselhos.

**palavras-chave**

interação, gestos, ambiente 3D, comunicação em rede, sensores de movimento

**resumo**

O ser humano naturalmente comunica através de gestos, expressões, palavras e movimentos que permitem descobrir o ambiente envolvente e manipular os objetos. A investigação e o desenvolvimento de novos dispositivos tecnológicos, como os sensores de movimento, permitem que atualmente se consiga adaptar cada vez mais os sistemas de forma a facilitar a interação entre o humano e o computador. As recentes pesquisas na área da interação humano-computador promoveram o desenvolvimento de interfaces naturais, que proporcionam formas mais intuitivas de interagir com aplicações multimédia interativas.

De forma a tornar a interação com o computador mais natural é possível detetar os movimentos do utilizador através de um sensor de movimento e controlar os avatares através dos gestos. As interfaces 3D permitem ao utilizador ter um sentido de presença no ambiente e, com recurso a um conjunto de tecnologias atualmente disponíveis, é possível desenvolver um sistema de controlo de um avatar num ambiente virtual 3D.

Do ponto de vista prático, esta investigação propõe um estudo de natureza exploratória, tendo como base a recolha bibliográfica nas áreas da interação humano-computador, interação natural, interfaces 3D, experiência do utilizador e usabilidade. O procedimento metodológico do estudo é investigação de desenvolvimento com a conceptualização e implementação de um protótipo de uma aplicação controlada por gestos num ambiente 3D colaborativo. A investigação através do desenvolvimento de um protótipo permitiu compreender as potencialidades e limitações das tecnologias que possibilitam a interação através de gestos num ambiente 3D multiutilizador e sistematizar um conjunto de recomendações para futuras aplicações deste tipo.

**keywords**

interaction, gestures, 3D environments, network communication, motion capture systems

**abstract**

The human being naturally communicates through gestures, expressions, words and movements that allow him to engage with the surrounding environment and manipulate objects. The research and development of new technological devices, such as motion capture sensors, make it so that nowadays, systems are increasingly adaptable, facilitating human-computer interaction. Recent studies in the human-computer interaction field have promoted natural user interface development, which generate more intuitive ways to approach interactive multimedia applications.

In such a way as to make interacting with a computer more natural, it's possible to detect user movements through a motion sensor, and control avatars with such gestures.

3D interfaces allow the user to feel immersed in the environment and, resorting to a set of currently available technologies, it is possible to develop an avatar control system in a 3D virtual environment.

From a practical standpoint, this research proposes an exploratory study, based on literature review on human-computer interaction, natural interaction, 3D interfaces, user experience and usability.

This study's methodological procedure is based on developmental research, along with the conceptualization and implementation of a gesture-controlled, collaborative 3D environment application's prototype.

Investigating through prototype development enabled the comprehension of potentialities and limitations of the technologies that allow multi-user 3D environment gesture interaction, as well as systematizing a set of recommendations for future applications of this type.



# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto	1
1.2	Problema de Investigação	2
1.3	Questão de investigação	4
1.4	Finalidade e Objetivos	4
1.5	Metodologia	6
<b>2</b>	<b>Enquadramento teórico</b>	<b>7</b>
2.1	Interação Humano-Computador	7
2.2	Interfaces Naturais	8
2.2.1	Definição de user interface	8
2.2.2	Evolução dos diferentes tipos de interface	9
2.2.3	Interfaces Naturais – Natural User Interfaces	12
2.3	Design de Interação	13
2.3.1	Interação em ambientes 3D	14
2.3.2	Paradigmas de interação em ambientes 3D	15
2.4	Dispositivos de captura de movimento	18
2.4.1	Câmaras de luz estruturada	19
2.4.2	Nintendo Wii remote	20
2.4.3	PlayStation Move Motion Controller	21
2.4.4	Microsoft Kinect para a Xbox	22
2.4.5	Asus Xtion PRO	28
2.4.6	Aspetos a acautelar no desenvolvimento de aplicações com dispositivos de captura de movimento	29
2.5	Tecnologias de reconhecimento 3D para o Kinect	30
2.5.1	Microsoft Kinect for Windows SDK	30
2.5.2	OpenNI (Open Natural Interaction)	31
2.5.3	OpenKinect	33
2.5.4	ZigFu	34
2.5.5	Kinetic Space	35
2.5.6	NI Mate	35
2.6	Interação e interfaces controladas por gestos	36
2.6.1	Gestos	36
2.6.2	Indicadores de caracterização dos gestos	37
2.6.3	Evolução das interfaces controladas por gestos	38

2.6.4	Problemas de Usabilidade nas interfaces controladas por gestos .....	43
2.6.5	Design de interação e interfaces com o sensor Kinect.....	45
<b>2.7</b>	<b>User experience .....</b>	<b>49</b>
<b>2.8</b>	<b>Exemplos de projetos de aplicações controladas por gestos.....</b>	<b>53</b>
2.8.1	Augmented Mirror .....	53
2.8.2	Project-Saara.....	54
2.8.3	Interactive Puppet.....	55
2.8.4	Follow This! .....	56
2.8.5	Trailerpark interactive visuals .....	57
2.8.6	Projeto SocialStage .....	58
2.8.7	Wi-Go.....	59
<b>2.9</b>	<b>Mundos Virtuais 3D.....</b>	<b>60</b>
2.9.1	Evolução dos mundos virtuais 3D .....	61
2.9.2	Principais características dos mundos virtuais.....	63
2.9.3	Desenvolvimento de ambientes virtuais 3D.....	64
<b>3</b>	<b>Metodologia .....</b>	<b>69</b>
3.1	Estrutura da investigação .....	70
3.2	Protótipo .....	70
3.3	Modelo de análise .....	71
<b>4</b>	<b>Implementação do protótipo .....</b>	<b>75</b>
4.1	Especificação dos requisitos .....	75
4.2	Escolha das Tecnologias .....	76
4.3	Pré-requisitos .....	78
4.4	Desenvolvimento prático do protótipo .....	80
4.4.1	Instalação da framework ZigFu no Unity3D.....	80
4.4.2	Interação Utilizador - Avatar no ambiente 3D, principais funções da framework ZigFu 83	
4.4.3	Comunicação em rede, criação de um servidor e de um cliente (Projeto com framework ZigFu).....	87
4.4.4	Framework ZigFu – Problemas no desenvolvimento .....	91
4.4.5	Framework OpenNI Wrapper.....	92
4.4.6	Desenvolvimento da Interface .....	100
<b>5</b>	<b>Conclusões .....</b>	<b>103</b>
5.1	Limitações do estudo .....	106
5.2	Proposta de desenvolvimentos futuros .....	107

<b>Bibliografia</b> .....	<b>109</b>
<b>Anexos</b> .....	<b>115</b>

## Índice de Figuras

Figura 1 – Linha de Comandos do sistema operativo Windows.....	10
Figura 2 – Exemplo da primeira interface gráfica ( <i>Xerox 8010 Star Information System</i> ). ....	11
Figura 3 – Contextualização da evolução do paradigma da interação natural.....	12
Figura 4 – Sistema com câmara de luz estruturada. ....	19
Figura 5 - O controlador <i>Wii remote</i> .....	20
Figura 6 - Detalhe do sensor.....	21
Figura 7 - <i>PlayStation Move Motion Controller</i> . ....	21
Figura 8 - <i>PlayStation Eye</i> . ....	22
Figura 9 – Estrutura interna detalhada do sensor Microsoft Kinect.....	24
Figura 10 – Estrutura externa do Kinect para a Xbox. ....	25
Figura 11 – Utilização do controlo por gestos pelo Primeiro Ministro Polaco. ....	26
Figura 12 – Interfaces controladas por gestos em situações de simulação. ....	27
Figura 13 – Interfaces controladas por gestos nos jogos. ....	27
Figura 14 - <i>Asus Xtion Pro</i> . ....	28
Figura 15 - <i>Asus Xtion Pro Live</i> . ....	28
Figura 16 - Resumo da arquitetura do funcionamento do <i>Kinect</i> no <i>Windows</i> (Microsoft, 2011b).....	30
Figura 17 – Logótipo <i>OpenNI</i> (“OpenNI,” n.d.).....	32
Figura 18 – Estrutura da <i>framework OpenNI</i> (“Open NI User Guide,” n.d.). ....	32
Figura 19 – Logótipo do <i>OpenKinect</i> (OpenKinect, 2011). ....	33
Figura 20 – Elograph.....	39
Figura 21 – <i>Accutouch</i> .....	40
Figura 22 – VIDEOPLACE.....	41
Figura 23 - Hewlett-Packard 150.....	42
Figura 24 – Esquema do funcionamento da <i>Digital Desk</i> . ....	42
Figura 25 – Direção, progressão e duração dos gestos. ....	46
Figura 26 – Gestos com uma mão e gestos com duas mãos.....	46
Figura 27 – Postura do utilizador e limites físicos. ....	47
Figura 28 – Duas interpretações diferentes do gesto “ <i>Wave</i> ”.....	48
Figura 29 – Ajuda na execução dos gestos através de imagens ou animações.....	48
Figura 30 - Gestos sugerido para sair da aplicação.....	52
Figura 31 – Projeto <i>Augmented Mirror</i> . ....	53
Figura 32 - Menu de utilitários da aplicação – disponível através de comandos por voz.....	55
Figura 33 - Indicação da posição para calibração e reconhecimento do utilizador.....	55
Figura 34 - Instalação multimédia interativa – <i>Interactive Puppet</i> . ....	56

Figura 35 - Jogo interativo controlado por gestos Follow This. ....	57
Figura 36 - <i>Trailerpark Interactive Visuals</i> . ....	57
Figura 37 - <i>Projeto SocialStage</i> (Hardeman & Licherdell, 2011). ....	58
Figura 38 - Wi-Go. ....	59
Figura 39 - Interface do <i>Starcraft</i> , um MUD. ....	62
Figura 40 - Exemplo do TinyMUD desenvolvido para o Apple II. ....	63
Figura 41 - <i>Second Life</i> (Book, 2004). ....	64
Figura 42 - openSimulator. ....	65
Figura 43 - Exemplo de modelo desenvolvido no Blender3D / Logótipo do Blender3D. ....	66
Figura 44 - <i>Unity3D</i> Logótipo. ....	66
Figura 45 - Exemplos de aplicações desenvolvidas com o <i>Unity3D</i> . ....	67
Figura 46 - Exemplo de posicionamento incorreto do sensor. ....	78
Figura 47 - Posicionamento correto do sensor. ....	79
Figura 48 - Instalação do Pacote de <i>Drivers</i> . ....	81
Figura 49 - Site <i>ZigFu</i> , <i>download</i> pacote para o <i>Unity3D</i> . ....	81
Figura 50 - <i>Unity3D</i> , criação de um novo projeto. ....	82
Figura 51 - Menu importação novo <i>package</i> . ....	82
Figura 52 - Visão geral do projeto no <i>Unity3D</i> ( <i>framework ZigFu</i> ). ....	83
Figura 53 - Visão geral interação utilizador-avator ( <i>framework ZigFu</i> ). ....	84
Figura 54 - Propriedades do Avatar 3D. ....	85
Figura 55 - Características do objeto "Sensor" ( <i>ZigFu</i> ). ....	86
Figura 56 - Declaração das variáveis de rede. ....	88
Figura 57 - Função OnGUI (primeira parte). ....	89
Figura 58 - Função OnGUI (segunda parte). ....	89
Figura 59 - Interface da rede (criar ou conectar a um servidor). ....	90
Figura 60 - Desconectar do servidor. ....	90
Figura 61 - Função OnConnectedToServer. ....	90
Figura 62 - Joints do esqueleto do avatar. ....	94
Figura 63 - <i>Joint "head"</i> do esqueleto do avatar com a componente <i>Network View</i> . ....	95
Figura 64 - Instanciar o avatar. ....	96
Figura 65 - Variável avatar0. ....	96
Figura 66 - Instanciar três avatares. ....	97
Figura 67 - Instanciar o cubo e esfera. ....	98
Figura 68 - Movimentos do cubo e da esfera através do controlo por teclas. ....	99
Figura 69 - Menu Aplicação. ....	100
Figura 70 - Opções do quadrado de representação da deteção de utilizadores. ....	101
Figura 71 - Seleção do elemento. ....	102



## Lista de Acrónimos

		<b>Breve descrição</b>
<b>API</b>	<i>Application Programming Interface</i>	É um conjunto de funções disponibilizado por um <i>software</i> para a utilização das suas funcionalidades por outras aplicações sem necessidade de entrar em detalhes de programação.
<b>CLI</b>	<i>Command-line user Interface</i>	Tipo de interface que funciona com a introdução de comandos ou instruções sob a forma de texto.
<b>GUI</b>	<i>Graphical user interface</i>	Tipo de interface que permite a interação com elementos digitais através de componentes gráficos como ícones, menus e outros indicadores visuais presentes no monitor de visualização.
<b>IHC</b>	Interação Humano-computador	Disciplina que estuda e investiga a interação entre as pessoas e os computadores.
<b>IP</b>	<i>Internet Protocol</i>	Protocolo de comunicação para o encaminhamento de pacotes de dados na internet.
<b>IxD</b>	<i>Interaction Design</i>	Tem como principal objetivo definir e facilitar a interação entre os utilizadores através de um produto ou um serviço.
<b>MMORPG</b>	<i>Massively multiplayer online role-playing game</i>	Género de jogo que permite muitos jogadores jogarem num mundo virtual dinâmico simultaneamente na Web.
<b>NUI</b>	<i>Natural User Interface</i>	Tipo de interface que é desenvolvida de forma a que a sua utilização seja o mais natural e intuitiva possível para o utilizador.
<b>OSC</b>	<i>Open Sound Control</i>	Formato de mensagens de comunicação entre computadores e outros dispositivos multimédia.
<b>QREN</b>	Quadro de Referência Estratégico Nacional	Enquadramento legal para a aplicação da política comunitária de coesão económica e social em Portugal no período 2007-2013.

<b>SDK</b>	<i>Software Development Kit</i>	Ferramenta de desenvolvimento de <i>software</i> distribuída para a criação de aplicações para determinada aplicação, sistema operativo, consola de jogos, etc.
<b>UI</b>	<i>User Interface</i>	Dispositivo através do qual decorre a interação entre o utilizador e a aplicação. A interface pode ser suportada em diferentes equipamentos, como computador, telemóvel ou iPad.
<b>USB</b>	<i>Universal Serial Bus</i>	Padrão de conexão que define os cabos, tipos de conectores e protocolos de comunicação entre computadores e dispositivos periféricos.
<b>UX</b>	<i>User Experience</i>	É a forma como uma pessoa experiencia a utilização de um produto, sistema ou serviço.





# 1 Introdução

## 1.1 Contexto

A presente dissertação descreve o trabalho de investigação teórica e prática desenvolvida no contexto do Mestrado em Comunicação Multimédia<sup>1</sup>, ramo Multimédia Interativo, relativo à conceção e desenvolvimento de um protótipo de uma aplicação multimédia interativa 3D colaborativa controlada por gestos.

Esta dissertação insere-se no projeto Engenho e Arte<sup>2</sup>, na componente deste projeto relacionada com a investigação e desenvolvimento de novos paradigmas de interação em ambientes virtuais 3D.

Engenho e Arte (SkilledArt) é um projeto promovido pela empresa ARTSHARE<sup>3</sup> juntamente com a Universidade do Porto e a Universidade de Aveiro na qualidade de copromotores. É um projeto financiado pelo Quadro de Referência Estratégico Nacional<sup>4</sup> (QREN) e tem como principal objetivo a investigação e desenvolvimento de interfaces e estratégias para a geração e manutenção de comunidades de conhecimento transdisciplinar no âmbito da tecnologia, ciência e arte. O seu objeto central é a comunicação entre espaços de interação físicos e virtuais integrados, no contexto da aprendizagem mediada por computador – *e-learning*. Para a mediação entre os espaços físicos e virtuais o projeto tinha como objetivo o desenvolvimento de interfaces de interação homem-máquina, no sentido da interação telemática homem-homem e homem-avataar.

O projeto Engenho e Arte contempla várias fases de desenvolvimento e implementação, onde se inclui investigação nas áreas da *Computer Vision*<sup>5</sup> e Bio-Electromagnetismo. A área de *Computer Vision* aborda sistemas multimédia interativos multiutilizador permitindo a criação de aplicações e ambientes de interação controlados por imagem. *Computer Vision* permite ainda deteção de movimento, reconhecimento de padrões e o despoletar de eventos específicos. A área de investigação do Bio-Electromagnetismo investiga e desenvolve técnicas para a transmissão telemática de sensações. Os resultados dos vários testes e experiências desenvolvidos no projeto Engenho e Arte serão registados num manual de boas práticas, estando a divulgação, promoção e

---

<sup>1</sup> <http://www.ua.pt/ca/PageCourse.aspx?id=101&b=1> Página de apresentação do Mestrado

<sup>2</sup> <http://skilledart.eu/> Página de apresentação do projeto Engenho e Arte

<sup>3</sup> <http://artshare.pt/> Página da empresa ARTSHARE

<sup>4</sup> <http://www.qren.pt/np4/home> Página do QREN

<sup>5</sup> **Computer vision** - Visão por computador - área de estudo da computação que estuda os sistemas de visão artificial implementadas por recurso a *hardware* e/ou *software*.

difusão do conhecimento desenvolvido nesta área a ser efectuada através da rede internacional de investigação *The Planetary Collegium*<sup>6</sup>, o parceiro estratégico internacional deste projeto.

O projeto Engenho e Arte tem como um dos seus objetivos o estudo de novos paradigmas de interação entre espaços físicos e virtuais. A presente dissertação tem por objetivo dar um contributo para a investigação sobre interfaces 3D e a interação controlada por gestos no contexto das interfaces de utilizador (interação homem-avatar) em ambiente multiutilizador (interação avatar-avatar). O desenvolvimento de um protótipo teve como objetivo permitir compreender o potencial e as limitações das tecnologias necessárias para responder às especificidades de uma aplicação desta natureza, e os resultados obtidos serão incluídos no manual de boas práticas do projeto Engenho e Arte.

Esta dissertação está organizada em sete capítulos. O presente capítulo, Introdução, pretende apresentar a contextualização e o problema de investigação, os objetivos do projeto e também a metodologia adotada para este estudo. O segundo capítulo, Enquadramento Teórico, introduz os conceitos fundamentais sobre as principais áreas envolvidas nesta investigação à luz do conhecimento atual. A Metodologia orientadora desta investigação é apresentada no terceiro capítulo. O quarto capítulo apresenta e discute a Implementação do Protótipo, incluindo a arquitetura e os componentes de *hardware* e *software* necessários, funcionalidades implementadas, e principais problemas enfrentados. No quinto capítulo são apresentadas as Conclusões deste trabalho, sendo expostos os principais resultados da investigação, as limitações do estudo e algumas perspectivas de trabalho futuro.

## **1.2 Problema de Investigação**

*"Gestures are much more richer than traditional input methodologies"*

(Valli, 2008)

Nesta dissertação pretendeu-se estudar o problema do controlo de um avatar<sup>7</sup> através da interação gestual num ambiente 3D virtual em rede. O objetivo essencial foi compreender qual o potencial e as limitações de algumas tecnologias atualmente disponíveis para esse efeito, nomeadamente a interface *Microsoft Kinect*<sup>8</sup> e o ambiente virtual *Unity3D*<sup>9</sup>.

---

<sup>6</sup> <http://www.planetary-collegium.net/>

<sup>7</sup> **Avatar** – representação gráfica do utilizador em forma tridimensional ou bidimensional. Tipicamente esta representação permite uma melhor exploração do ambiente virtual 3D.

<sup>8</sup> **Microsoft Kinect** – Sensor de movimento da consola de jogos *Microsoft Xbox 360*. O *Kinect* é um dispositivo que permite aos utilizadores controlarem a Xbox sem nenhum controlo ou comando adicional.

A interação com o computador através da linha de comandos (*Command Line interface*) representou a etapa inicial da interação entre humanos e os computadores. Este tipo de comunicação exigia ao utilizador que conhecesse um conjunto de comandos e instruções para interagir com o sistema. Tratava-se de uma interação muito exigente do ponto de vista do esforço cognitivo pedido ao utilizador. Mais tarde houve uma evolução e passou-se para o paradigma das interfaces gráficas (*Graphical User Interface*), que foi uma tentativa bem-sucedida de aproximar e facilitar a comunicação entre os utilizadores e o computador. A base deste paradigma são as metáforas gráficas. Mais recentemente assistiu-se a uma evolução para as interfaces naturais, ou seja, para um paradigma de interação natural em que o utilizador pode utilizar gestos ou sons. O ser humano naturalmente comunica através de gestos, expressões, movimentos e descobre o mundo olhando em volta e manipulando os objetos (Valli, 2008). É para este paradigma que se pretende que a interação com os computadores evolua, assumindo uma forma mais natural para o utilizador e com uma curva de aprendizagem menor do que a que se verifica com os anteriores paradigmas.

Historicamente, a linguagem de comunicação entre pessoas e computadores foi essencialmente determinada por constrangimentos tecnológicos e os utilizadores tiveram de se adaptar à linguagem dos computadores. O que se verifica atualmente é a tentativa de construir máquinas e sistemas que "falem" como os humanos.

Os mais recentes avanços nas técnicas de *Computer Vision* e o desenvolvimento de sensores de movimento, como o *Microsoft Kinect*, permitem recolher imagem em profundidade e detetar uma forma humana, possibilitando, assim, a interação com o computador sem necessidade de um dispositivo intermédio como o rato, o teclado ou outro tipo de dispositivo.

A necessidade de tornar a manipulação dos objetos virtuais mais intuitiva e ao mesmo tempo mais natural para o utilizador, foi uma das principais motivações deste projeto. Pretendeu-se tirar partido das mais recentes evoluções tecnológicas ao nível da interação para conceber e desenvolver um protótipo que permitisse compreender os problemas da manipulação em tempo real de um avatar num ambiente 3D distribuído. Este protótipo tinha por objetivo disponibilizar um ambiente onde vários utilizadores poderiam controlar o seu avatar interagindo com esse ambiente e com os restantes avatares.

O problema desta investigação centrou-se essencialmente no estudo das vantagens e limitações associadas às tecnologias de suporte da interação natural do utilizador, numa interface controlada

---

<sup>9</sup> **Unity3D** – Motor de jogo desenvolvido pela empresa *Unity Technologies*

por gestos num ambiente 3D distribuído. Pretendeu-se estudar e compreender de que forma a interação natural, desprovida de objetos acessórios como o rato e o teclado, influencia a experiência do utilizador numa aplicação multimédia interativa.

### **1.3 Questão de investigação**

A questão de investigação pretende, de forma resumida, apresentar o que se procura saber, investigar e compreender melhor no estudo. Esta questão é um elemento essencial e servirá como fio condutor da pesquisa. Como referem os autores Quivy & Campenhoudt (1995), a pergunta deve ser precisa e concisa para orientar a investigação sem desvios da problemática, deve também ser exequível e pertinente. Tendo em conta o contexto e a problemática da investigação, a pergunta que orientou esta dissertação é a seguinte:

Qual o potencial e as limitações das atuais tecnologias de suporte à **interação controlada por gestos** na promoção de novas **formas de interação** em **ambientes 3D** distribuídos?

### **1.4 Finalidade e Objetivos**

A interação em ambientes 3D muitas vezes é efetuada através da utilização de dispositivos de entrada como o rato e o teclado. O principal objetivo deste trabalho é perceber em que medida algumas tecnologias atualmente disponíveis permitem que a interação controlada por gestos, a interação natural, seja utilizada como forma de promover um novo paradigma de interação em ambientes 3D distribuídos.

Para tal, no âmbito desta investigação propõe-se conceber e prototipar um ambiente 3D distribuído que permita um controlo, através de gestos, de movimentos do avatar característicos de um humanóide. A aplicação deverá estar desenvolvida de forma que permita ser um ambiente de colaboração, proporcionando assim uma interação entre vários utilizadores separados fisicamente, mas juntos, em tempo-real, no ambiente 3D. Através de um referencial teórico sólido e de um contexto de uso específico e delimitado pretende-se construir um conjunto de estudos que permita responder à pergunta de investigação proposta. A investigação contempla componentes de investigação de técnicas de *Computer Vision*, de captura de movimentos/gestos e de modelação e interação em ambientes 3D.

Seguidamente apresentam-se os objetivos gerais e específicos delineados para esta investigação.

**Compreender os conceitos de Interação humano-computador, interação e interação gestual:**

- Identificar os momentos chave da história da interação humano-computador e sua influência no âmbito dos produtos multimédia interativos;
- Compreender o conceito de interação e identificar os paradigmas de interação;
- Descrever o conceito de interação gestual e principais características;
- Identificar as características essenciais de uma aplicação controlada por gestos num ambiente 3D distribuído.

**Perceber de que forma a interação natural influencia a experiência de utilização (UX) de uma aplicação multimédia 3D distribuída:**

- Definir o conceito de *User experience* (UX);
- Identificar os principais problemas de usabilidade associados à interação controlada por gestos num ambiente 3D distribuído;
- Identificar as principais características de usabilidade adaptadas a uma aplicação 3D distribuída.

**Identificar os principais sensores de movimento e tecnologias de reconhecimento 3D:**

- Identificar os principais sensores de movimento atualmente disponíveis e as suas características;
- Descrever as tecnologias mais relevantes de reconhecimento 3D;
- Perceber as relações entre as inovações tecnológicas ao nível dos sensores de movimento e das tecnologias de reconhecimento 3D na promoção e desenvolvimento de aplicações baseadas no paradigma de interação natural.

**Desenvolver um protótipo funcional que permita a interação gestual com um avatar no ambiente 3D distribuído:**

- Identificar os requisitos funcionais do protótipo da aplicação;
- Pesquisar os requisitos ao nível de *software* necessário para o desenvolvimento da aplicação de acordo com os requisitos funcionais especificados;
- Desenvolver o módulo de rede que permita a comunicação em ambiente 3D distribuído;
- Desenvolver a componente de interação utilizador-avatar através da interação gestual;
- Elaborar o módulo de interação avatar-avatar no ambiente 3D distribuído;
- Construir um ambiente que permita a interação entre os vários utilizadores no ambiente virtual em tempo real.

## **1.5 Metodologia**

A natureza da presente investigação é exploratória pois pretende-se perceber um fenómeno relativamente recente e do qual não se conhece o suficiente.

O procedimento metodológico da investigação é investigação de desenvolvimento ou investigação tecnológica (R. Richey, Klein, & Nelson, 2004). Esta abordagem metodológica começa por identificar e analisar um problema e tem por finalidade conceptualizar e desenvolver uma proposta de resolução desse problema. Este processo é composto por uma investigação de natureza teórica mas também o desenvolvimento de objetos de estudo práticos e envolve a necessidade de verificação e validação através do desenvolvimento de dispositivos experimentais.

Do ponto de vista teórico, esta investigação contempla a revisão bibliográfica dos conceitos teóricos associados à investigação, mas também o contacto com outros investigadores na área. Numa perspetiva prática esta investigação centra-se no processo de conceção e implementação de um protótipo funcional que permita testar e validar os processos de desenvolvimento.

A investigação relacionada com o desenvolvimento do protótipo irá permitir perceber como se comportam algumas das tecnologias atualmente disponíveis que permitem a interação controlada por gestos num ambiente 3D distribuído. O estudo deste protótipo pretende também averiguar as potencialidades e limitações das atuais tecnologias de suporte à interação controlada por gestos num ambiente 3D distribuído.

## 2 Enquadramento teórico

Neste capítulo é apresentada uma revisão bibliográfica dos principais conceitos envolvidos na investigação. A revisão da literatura permite identificar e analisar documentos que se relacionam com a temática da investigação e assim contribuir para uma melhor compreensão do contexto e dos pilares teóricos deste estudo. Pretende-se, ainda, apresentar o estado da arte do contexto deste estudo, bem como identificar projetos de investigação em curso nas áreas dos sistemas de visão por computador, interfaces controladas por gestos e mais especificamente o controlo de interfaces em ambientes 3D através de gestos.

### 2.1 Interação Humano-Computador

A disciplina da interação humano-computador (IHC) centra-se no estudo, planeamento e desenho de interações entre os utilizadores e os computadores. A interação entre os utilizadores (humanos) e o computador processa-se através de uma interface (*user interface*) e é possível devido ao *hardware* e *software* desenvolvidos para o efeito. Um dos principais propósitos da interação humano-computador é promover e melhorar as interações entre os utilizadores e os computadores tornando os computadores mais fáceis de utilizar, adaptando assim os sistemas ao utilizador e às suas necessidades (Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, 1996).

*“Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them.”* (Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, 1996).

A interação humano-computador está ligada a muitas disciplinas, como as ciências da comunicação e da informação, ciências da computação e engenharia, ergonomia, sociologia, design gráfico entre outras.

*“HCI involves the design, implementation, and evaluation of interactive systems in the context of the user’s task and work”* (Dix, Finlay, Abowd, & Beale, 2003).

O termo interação humano-computador começou a ser mais divulgado na década de 80, mas foi com a Segunda Guerra Mundial que o interesse pelo estudo da interação entre os humanos e as máquinas começou. Nesta fase as preocupações estavam centradas nas questões físicas do trabalhador e no aumento da sua *performance* no trabalho (Dix et al., 2003). Até aos anos 70 do

século passado, as pessoas que mais interagiam com computadores eram especialistas e profissionais ou pessoas com grande interesse na tecnologia como um *hobby*<sup>10</sup> (Carroll, 2011). Com o aparecimento e a evolução dos computadores, os investigadores começaram a centrar a sua atenção na interação entre os humanos e os computadores, não sendo só uma preocupação ao nível físico, mas também ao nível psicológico, bem como de outros aspetos teóricos da interação entre o humano e o computador. O aparecimento do computador pessoal nos anos 80 fez com que a utilização do computador deixasse de ser exclusiva dos investigadores da área da computação, passando todas as pessoas a ser potenciais utilizadores do computador (Pavlovic, Sharma, & Huang, 1997). Um aumento do número de utilizadores, juntamente com a necessidade de adaptar o sistema ao utilizador, levou vários investigadores a pensar nas pessoas e nos computadores para além das questões meramente funcionais e começar a investigar áreas relacionadas com a usabilidade (Carroll, 2011).

Inicialmente os principais objetivos da interação humano-computador eram o foco na produtividade das aplicações de uso pessoal (como editores de texto, por exemplo). Mas depois a disciplina começou a abranger outras áreas como a visualização, os sistemas de informação ou os sistemas colaborativos. Atualmente já não é apenas uma disciplina específica da ciência da computação, mas uma disciplina que estuda não só o comportamento individual do utilizador mas também os comportamentos sociais e organizacionais (Carroll, 2011).

*"It expanded from desktop office applications to include games, e-learning, e-commerce, military systems, and process control. It expanded from early graphical user interfaces to include myriad interaction techniques and devices, multi-modal interactions, and host of emerging ubiquitous, handheld and context-aware interactions."* (Carroll, 2011).

## **2.2 Interfaces Naturais**

### **2.2.1 Definição de user interface**

*"Each time someone uses an application, or any digital product, he carries on a conversation with the machine. (...) The user interface mediates that conversation, helping users achieve whatever ends they had in mind. "* (Tidwell, 2011).

---

<sup>10</sup> Passatempo ou atividade de entretenimento.

No contexto da interação Humano-computador (IHC) a *user interface* (UI) – interface do utilizador - refere-se aos elementos gráficos, textuais ou auditivos apresentados ao utilizador e que este pode controlar através dos diferentes dispositivos de entrada, como por exemplo o rato ou o teclado. Com a evolução da tecnologia, e o aparecimento de mais dispositivos e de mais funcionalidades, surgiu a necessidade de desenvolver as questões relacionadas com a apresentação dos dados e organização dos elementos gráficos. Tipicamente a criação dos elementos gráficos como ícones, botões, tipografia ou menus é da responsabilidade do designer de interfaces (*Interface Designer*) (Galitz, 2007).

*User interface design* é uma subárea de investigação no âmbito da disciplina de interação humano-computador. A *user interface* é uma componente das aplicações que correm num computador e consiste em *software* que permite ao utilizador fornecer indicações que controlam ou influenciam a execução da aplicação e receber informações da aplicação, executando ações de visualização, audição, toque (*touch*) e manipulação. A *user interface* é composta por dois componentes essenciais, o *input* e o *output*. O *input* consiste na forma como o utilizador comunica as suas indicações ao computador. Rato, teclado, voz (instruções por voz) ou as mãos (para dispositivos que permitem o toque) são exemplos de componentes de entrada de dados. O *output* consiste na forma como o sistema apresenta os resultados da interação do utilizador com o sistema, através de dispositivos de *output* como o monitor vídeo ou as colunas de som.

Uma boa interface resulta da combinação ideal de vários fatores entre o *input* e o *output* que satisfaça as necessidades do utilizador da forma mais eficiente possível, e deve permitir ao utilizador focar-se na informação e nas tarefas a realizar em vez dos mecanismos de apresentação da informação (Tidwell, 2011).

A questão da interface é relevante, pois do ponto de vista dos utilizadores, a interface é o que está imediatamente visível ao utilizador e representa muitas vezes a totalidade do sistema que o utilizador tem acesso. Todos os outros elementos constituintes do sistema (*hardware*, componentes de *software*) são normalmente invisíveis para o utilizador. Assim, um dos principais objetivos do design de interface é fazer com que a utilização do sistema seja simples, produtivo e divertido (Galitz, 2007).

### **2.2.2 Evolução dos diferentes tipos de interface**

Acompanhando a evolução dos computadores, os designers têm desenvolvido as interfaces sempre na tentativa de melhorar a comunicação entre o utilizador e o computador. Sistemas complexos com estruturas pouco flexíveis e difíceis de adaptar para o utilizador exigiram novas soluções que possibilitassem a utilização dos sistemas de forma mais fácil e universal. As

interfaces estão em constante evolução, com o objetivo de ser mais um elemento facilitador do processo de comunicação entre o humano e o computador (Galitz, 2007).

As interfaces passaram por várias fases, começando pela linha de comandos (*command line interface*). Este tipo de interface permite a inserção de instruções (comandos) para o computador através de texto. A linha de comandos, tal como a utilizada no sistema operativo MS-DOS (Figura 1), está presente desde os anos 60 do século XX na interação humano-computador (Liu, 2011). Permite o acesso a todas as funcionalidades do computador, mas apresenta alguma dificuldade porque é necessário conhecer o repertório de comandos incluindo as suas diferentes variantes. São comandos difíceis de memorizar, muitas vezes com designações complexas e pouco explícitas da ação que desempenham, frequentemente com múltiplos parâmetros, o que aumenta a probabilidade de confusão e erros na sua execução (Dix et al., 2003).

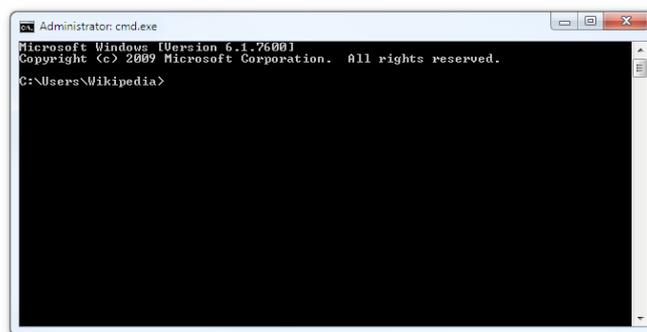


Figura 1 – Linha de Comandos do sistema operativo Windows inspirada na do sistema operativo MS-DOS<sup>11</sup>.

A evolução das tecnologias permitiu o aumento e a diversificação das tarefas que se podiam fazer num computador e, conseqüentemente, o aumento do número de utilizadores. O aumento da quantidade e tipo de tarefas pode estar relacionado com a evolução das interfaces para uma nova etapa, as Interfaces gráficas (*Graphical user interface*) (Wigdor & Wixon, 2011).

A primeira interface gráfica foi inserida num sistema operativo em 1981 e continha ícones gráficos, janelas e menus. Tratou-se do sistema *Xerox 8010 Star Information System*, de cuja interface se pode ver um exemplo na Figura 2.

---

<sup>11</sup> Retirado de [http://upload.wikimedia.org/wikipedia/en/8/89/Command\\_prompt\\_on\\_windows\\_vista.png](http://upload.wikimedia.org/wikipedia/en/8/89/Command_prompt_on_windows_vista.png)

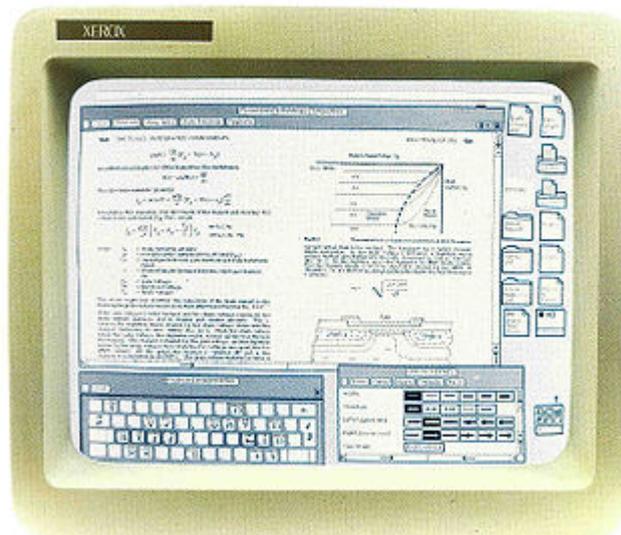


Figura 2 – Exemplo da primeira interface gráfica (Xerox 8010 *Star Information System*)<sup>12</sup>.

A ideia principal subjacente à evolução das interfaces é a de desenvolver sistemas mais fáceis de utilizar, que sejam mais intuitivos e naturais para quem interage. Neste sentido assiste-se atualmente a um desenvolvimento de interfaces naturais (*Natural User interface*), as quais potenciam uma diminuição da curva de aprendizagem por parte dos utilizadores (Villaroman, Rowe, Ph, & Swan, 2011).

*“NUIs promise to reduce the barriers to computing still further, while simultaneously increasing the power of the user, and enabling computing to access still further niches of use.”* (Wigdor & Wixon, 2011).

A nova abordagem das interfaces naturais não deve ser considerada como uma substituição dos paradigmas anteriores, mas deve ser encontrado o espaço (contextos de uso, novas aplicações) onde se verifique que este novo paradigma pode ajudar a melhorar a eficiência na comunicação. A Figura 3 sintetiza os principais patamares da evolução das interfaces até ao momento.

<sup>12</sup>Retirado de [http://upload.wikimedia.org/wikipedia/en/f/f2/Xerox\\_8010\\_compound\\_document.jpg](http://upload.wikimedia.org/wikipedia/en/f/f2/Xerox_8010_compound_document.jpg)

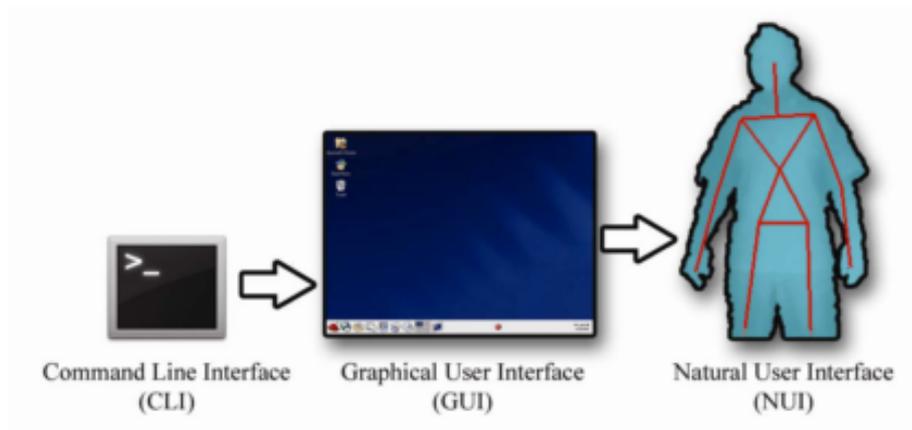


Figura 3 – Contextualização da evolução do paradigma da interação natural<sup>13</sup>.

A comunicação entre humanos não envolve dispositivos, pois o ser humano é capaz de sentir o ambiente envolvente (através dos sentidos como a visão, a audição ou o tacto). Assim se pretende também que a interação com os sistemas funcione, que imite estas capacidades do ser humano através da utilização de dispositivos como câmaras e microfones (Von Hardenberg & Bérard, 2001).

### 2.2.3 Interfaces Naturais – Natural User Interfaces

Dan Saffer (Saffer, 2009) refere alguns aspetos essenciais a ter em conta no desenvolvimento de um sistema interativo, especificamente de uma interface controlada por gestos, características que se podem generalizar para o contexto mais geral das interfaces naturais:

- **Discoverable** – Como indicar ao utilizador que um ecrã é *touchscreen*? Como indicar ao utilizador que determinado ambiente é interativo? É aqui que surge uma das características importantes das interfaces controladas por gestos: as *affordances*<sup>14</sup>. *Affordance* trata-se de uma propriedade de um objeto que indica ao utilizador de que forma este pode interagir com o sistema ou com funcionalidades do sistema.
- **Trustworthy** – Antes do utilizador interagir com o sistema o sistema deve transmitir confiança e segurança. A aplicação deverá também respeitar a privacidade do utilizador.

<sup>13</sup> Imagem retirada do artigo intitulado “A Study on Natural Interaction for Human Body Motion Using Depth Image Data” (Sidik et al., 2011).

<sup>14</sup> **Affordances** – uma das várias características dos objetos ou ambientes que dá indicações imediatas de como se deve interagir com uma ou várias funcionalidades do objeto em questão. Potencial de um objeto para que ele seja manipulado da forma como foi pensado para funcionar.

- **Responsive** – A necessidade do sistema responder às interações do utilizador, ou seja dar *feedback*. Sem uma resposta às interações do utilizador, ele vai continuar a repetir determinada ação até que alguma coisa aconteça e o sistema responda.
- **Appropriate** – As aplicações controladas por gestos deverão ser adequadas aos contextos e culturas. Determinados gestos aceites numa cultura podem ter uma interpretação completamente diferente noutra cultura.
- **Meaningful** – Os gestos deverão ter significado para o utilizador.
- **Smart** – Pretende-se que os sistemas façam o que é humanamente mais complicado, processamento rápido, memória infalível, reconhecimento de padrões complexos. Os sistemas devem ser inteligentes e deverão fazer aquilo que o utilizador precisa.
- **Clever** – O sistema deve fazer uma deteção automática das necessidades do utilizador, e responder a essas necessidades de uma forma agradável ao utilizador.
- **Playful** – Através de uma interface agradável e divertida, o utilizador terá mais interesse em explorar a aplicação e as funcionalidades. A possibilidade do utilizador errar na utilização da aplicação deve ser reduzida.
- **Pleasurable** – As interfaces e funcionalidades das aplicações controladas por gestos deverão ser esteticamente agradáveis. O aspeto visual interessante poderá permitir ao utilizador uma utilização mais agradável.
- **Good** – Ter respeito pelo utilizador e pensar nas limitações que este tem. Evitar ridicularizar a interação do utilizador (especialmente em frente a outras pessoas) ou dificultar a performance do utilizador ao colocar gestos demasiado complexos ou impossíveis de realizar.

Estas são recomendações teóricas e práticas a ter em conta quando se desenvolve uma interface natural. São conselhos importantes e que podem fazer uma boa interface natural.

### 2.3 Design de Interação

A disciplina de design de interação pode ser considerada uma extensão da interação humano-computador, pois deriva da necessidade de identificar problemas associados à IHC, mas também desenvolver soluções que facilitem a interação entre os utilizadores através de um produto ou de um serviço.

O design de interação é definido por Dan Saffer como “*the art of facilitating interactions between humans through products and services*” (Saffer, 2007). Essa designação foi utilizada pela primeira vez pelo CEO da empresa IDEO, Bill Moggridge, que sentindo que o tipo de produtos que estava a desenvolver não eram exatamente design de produto nem ciência computacional, decidiu designar essa nova forma de design como Design de interação (Saffer, 2007).

Na sua obra de 2007, *Designing interactions*, Bill Moggridge apresenta a definição de Gillian Crampton “*interaction design is shaping our life with interactive technologies – computers, telecommunications mobile phones, and so on. If I were to sum up interaction design in a sentence, I would say that it’s about shaping our every day life through digital artifacts – for work, for play, and for entertainment*” (Moggridge, 2007). Sharp, Rogers e Preece (2007) referem ainda que “*By interaction design, we mean designing interactive products to support people in their everyday and working lives. In particular, it is about creating user experiences that enhance and extend the way people work, communicate and interact.*”.

O design de interação pode permitir resolver problemas de comunicação entre os utilizadores e o sistema, e preocupa-se com a facilitação da comunicação entre humanos, ou seja, a comunicação entre utilizadores através dos dispositivos e serviços (Saffer, 2007).

O design de interação relaciona-se com o *feedback* do sistema, com a navegação nos sistemas interativos, com a consistência da aplicação, ou seja, uma mesma instrução tem de produzir o mesmo efeito noutra parte da aplicação, caso contrário torna-se ambígua. O design de interação apresenta um conjunto de características que ajuda no desenvolvimento de aplicações multimédia interativas (Moggridge, 2007).

### **2.3.1 Interação em ambientes 3D**

Uma interface 3D envolve a interação humano-computador na qual as ações dos utilizadores ocorrem num ambiente virtual tridimensional com dispositivos de entrada 3D ou 2D (mas posteriormente mapeados para 3 dimensões). Os utilizadores controlam essas interfaces através de dispositivos de entrada e de paradigmas de interação que possibilitam a manipulação de elementos 3D gerados por computador (LaViola & Keefe, 2011). Estes paradigmas de interação são métodos que permitem ao utilizador cumprir com os objetivos pretendidos dentro da aplicação e podem ser implementados independentemente dos dispositivos de captura de movimento (Bowman, Kruijff, LaViola, & Poupyrev, 2005).

A interação em ambientes 3D permite aos utilizadores terem uma sensação de presença nesse ambiente virtual. Este tipo de interação é mais direta do que a que acontece em ambiente 2D e pode permitir diminuir a distância cognitiva, pois os utilizadores realizam tarefas mais naturais, permitindo reduzir o tempo de resposta entre a ação do utilizador e o *feedback* do sistema (Bowman et al., 2005).

Existem quatro paradigmas distintos de interação nas aplicações 3D (Bowman et al., 2005): navegação, seleção, manipulação e controlo do sistema. Existem ainda outras formas que podem ser consideradas, mas apenas são utilizadas em aplicações com contextos muito específicos.

A forma como o utilizador pode interagir com os elementos 3D é uma componente importante no âmbito desta investigação, pelo que em seguida se fará uma breve introdução dos quatro principais paradigmas de interação em ambientes 3D.

### **2.3.2 Paradigmas de interação em ambientes 3D**

Tal como referido anteriormente, ao nível da interação em ambientes 3D existem essencialmente quatro paradigmas de interação: navegação, seleção, manipulação e controlo do sistema. Cada um destes apresenta características específicas que a seguir se apresentam.

#### **2.3.2.1 Navegação**

A navegação é o paradigma mais frequente na interação em ambientes 3D. Corresponde à deslocação no ambiente virtual e subdivide-se em dois componentes: *Travel* e *Wayfinding*.

O computador precisa de informar o utilizador relativamente à sua localização e movimento no ambiente 3D. O *Wayfinding* permite desenvolver e planear rotas que permitam chegar ao destino pretendido.

O *Travel* diz respeito à movimentação de um determinado ponto até outro da localização do ambiente 3D, ou seja, este é o componente que permite ao utilizador movimentar-se e explorar o ambiente 3D.

O *Travel* subdivide-se em quatro categorias:

- ***Gaze-Directed Steering***: este tipo de movimentação/deslocação no ambiente virtual assume a direção do olhar da câmara (normalmente o centro de visualização da janela do monitor do computador do utilizador) e tipicamente a movimentação realizada *frame a frame*. *"To implement gaze-directed steering, typically a callback function is set up that executes before each frame is rendered. Within this callback, first obtain the head tracker information (usually in the form of a 4x4 matrix). This matrix gives you a transformation between the base tracker coordinate system and the head tracker coordinate system. By also considering the transformation between the world coordinate system and the base tracker coordinates (if any), you can get the total composite transformation. Now, consider the vector (0, 0, -1) in head tracker space (the negative z-axis, which usually points out the front of the tracker). This vector, expressed in world coordinates, is the direction you want to move."* (LaViola & Keefe, 2011).

- **Pointing:** a direção do movimento é efetuada através do controlo das mãos do utilizador. Ao nível da implementação prática a técnica é a mesma da *gaze-directed steering* (LaViola & Keefe, 2011).
- **Map-based Travel:** Nesta técnica, o utilizador está representado graficamente num mapa e o utilizador movimenta a representação do mapa para a localização desejada. O sistema anima a mudança de localização do utilizador de um local para o outro. Ao nível da implementação prática desta técnica é necessário ter em atenção dois aspetos: a escala do mapa e qual a relação entre cada ponto no mapa e a correspondente posição no ambiente virtual (necessidade de saber as coordenadas no mapa e as correspondentes no ambiente virtual) (LaViola & Keefe, 2011).
- **Grabbing the Air:** Esta técnica permite, literalmente, agarrar o ar, e pode ser comparada ao utilizador a puxar-se numa corda para se mover. Esta técnica permite uma navegação em qualquer direção e é utilizada em situações como nadar, voar, subir uma montanha ou um edifício.

### 2.3.2.2 Seleção

O paradigma seleção é o que permite aceder e selecionar um objeto no ambiente virtual 3D. A seleção do objeto normalmente é efetuada através de um botão (teclado ou rato), gestos ou comandos por voz. Um dos aspetos fundamentais da seleção é dar o *feedback* ao utilizador de que o objeto está na mira de ser selecionado. Ter uma lista de objetos que podem ser selecionados aumenta a eficiência da aplicação, pois o sistema não necessita de estar a testar todos os objetos para verificar se podem ser selecionados (LaViola & Keefe, 2011).

Existem quatro técnicas de implementação do paradigma de seleção:

- **Virtual hand:** Técnica de seleção mais frequente, funciona através da intersecção da mão virtual com o objeto virtual.
- **Ray-casting:** Esta técnica, tem como metáfora o laser, ou seja existe um laser invisível que sai da direção do utilizador até ao infinito, podendo ser selecionado o primeiro objeto que o laser interseta.
- **Occlusion:** Nesta técnica o objeto é selecionado através da aproximação da mão virtual (não existe toque), o que permite a seleção de objetos distantes.
- **Arm extension:** Esta técnica permite selecionar objetos distantes através da desintegração do mapeamento da mão real para a mão virtual; permite assim que o braço da mão virtual possa ser aumentado para além de tamanhos humanamente possíveis e possa, assim, alcançar objetos distantes.

### 2.3.2.3 Manipulação

A manipulação está diretamente ligada à seleção dos objetos, pois um objeto tem necessariamente de ser selecionado para poder ser manipulado. Quando um objeto está a ser manipulado é necessário desativar a seleção, para evitar sobreposição e erros de manipulação (LaViola & Keefe, 2011). Existem várias técnicas que permitem a implementação do paradigma da manipulação:

- **HOMER:** A técnica *Hand-Centered Object Manipulation Extending Ray-Casting (HOMER)* utiliza a técnica de seleção *Ray-casting* sendo deslocada a mão virtual até ao objeto para a manipulação. A utilização desta técnica implica uma discordância entre a posição da mão real e a da mão virtual.
- **Scaled-World Grab:** Esta técnica é frequentemente utilizada com a técnica de seleção *Occlusion*. Após a seleção do objeto pretendido o utilizador (avatar) é aumentado (*scaled up*) de forma a dar a ideia de que a mão está de facto a tocar no objeto. No caso de o utilizador não se movimentar não se vai aperceber deste efeito de escala; mas se se mover vai aperceber-se de que existem diferenças na escala dos elementos.
- **World-in-Miniature:** Esta técnica cria uma versão miniatura do mundo virtual e permite a manipulação direta dos elementos virtuais através da utilização da técnica de seleção *virtual hand*.

### 2.3.2.4 Controlo do sistema

Ao paradigma do controlo do sistema estão associadas tarefas que envolvem a emissão de comandos para a aplicação, personalização da aplicação ou ativação de algumas funcionalidades. Algumas das técnicas que possibilitam o controlo do sistema incluem menus, comandos por voz ou gestos. A colocação dos menus influencia o acesso por parte do utilizador, pelo que é relevante ter em conta o posicionamento para que os menus sejam acessíveis.

*"Gestural interaction can be a very powerful system control technique."* (LaViola & Keefe, 2011).

Existem alguns problemas associados à interação através de gestos nos ambientes 3D; como a memorização dos gestos pode ser difícil, utilizadores inexperientes poderão sentir dificuldades. Por forma a diminuir os erros e as dificuldades por parte dos utilizadores, pode ser necessário um reforço do *feedback* ao utilizador. *"In order to make gestural interaction easier to use for a less advanced user, strong feedback, like visual cues after initiation of a command, might be needed."* (LaViola & Keefe, 2011).

Os comandos por voz permitem desenvolver várias tarefas, como inicialização do sistema ou seleção de objetos. "*The use of voice input as a system control technique can be very powerful: it is hands-free and natural.*" (LaViola & Keefe, 2011). A utilização de interação gestual pode assumir várias tarefas, como inicializar o sistema, ou permitir desencadear outros eventos (associação de determinado gesto a um evento, como, por exemplo, sair da aplicação).

## **2.4 Dispositivos de captura de movimento**

A evolução da tecnologia nos últimos anos permitiu o desenvolvimento dos sensores de movimento (*Motion Capture Systems*), nomeadamente para aplicações na área de jogos. O aparecimento dos dispositivos de controlo de movimento (*motion controllers*) possibilitou e facilitou a sua utilização, não só nos jogos mas também na construção de aplicações noutras áreas (Katz, Cook, & Smart, 2011). Sensores são dispositivos que convertem um sinal físico num sinal elétrico passível de ser manipulado no computador, ou seja, o sensor deteta modificações no ambiente (Wilson, 2008). As alterações no ambiente podem ser de várias naturezas, por isso existem vários tipos de sensores, sendo os mais comuns, no âmbito das interfaces controladas por gestos, sensores de proximidade, luz, pressão, som, movimento, orientação e *tilt* (inclinação, ângulo) (Saffer, 2009). A investigação da interação humano-computador na área dos sensores pretende perceber como estes podem promover e melhorar a interação, na elaboração de novos dispositivos de entrada, ou no desenvolvimento de aplicações com comportamentos mais próximos do dos humanos.

As mais recentes inovações dos três maiores fabricantes na indústria dos jogos (*Microsoft*, *Nintendo* e *Sony*) focam-se no desenvolvimento de equipamentos *user-friendly*, ou seja, o desenvolvimento de produtos que valorizam a usabilidade, acessibilidade e a experiência do utilizador. O potencial destes dispositivos está a ser explorado muito para além da área dos videojogos (Sung, 2011). Dispositivos como o *Nintendo Wii Remote*, o *Microsoft Kinect* ou o *Playstation move* permitem efetuar um reconhecimento do movimento do utilizador com um nível de detalhe elevado (LaViola & Keefe, 2011). Os vários sensores de movimento disponíveis no mercado apresentam características distintas e providenciam diferentes experiências de utilização. A análise detalhada das características de cada um desses dispositivos de captura de movimento, as suas vantagens e limitações, é indispensável para se compreender o potencial de utilização de cada dispositivo.

As subsecções seguintes reúnem informação relevante sobre os dispositivos de captura de movimento comercializados pelos fabricantes identificados, sobre a tecnologia que permite a criação de imagens em 3D bem sobre como estes dispositivos podem ser utilizados na construção de interfaces 3D.

### 2.4.1 Câmaras de luz estruturada

As câmaras de luz estruturada (*Structured light camera systems*) estão equipadas com tecnologia que permite a criação de uma imagem 3D, através da projeção de um sinal, que pode ser uma frequência de luz codificada ou padrões. Com isto procede-se à observação de como o sinal atinge e é refletido pela superfície ou de como o padrão emitido é deformado, determinando-se assim a distância a que a superfície se encontra do ponto de origem do sinal. A partir desses dados é possível construir uma imagem a três dimensões. A Figura 4 mostra um exemplo de um sistema com utilização de uma câmara de luz estruturada (Keane, Hall, & Perry, 2012).

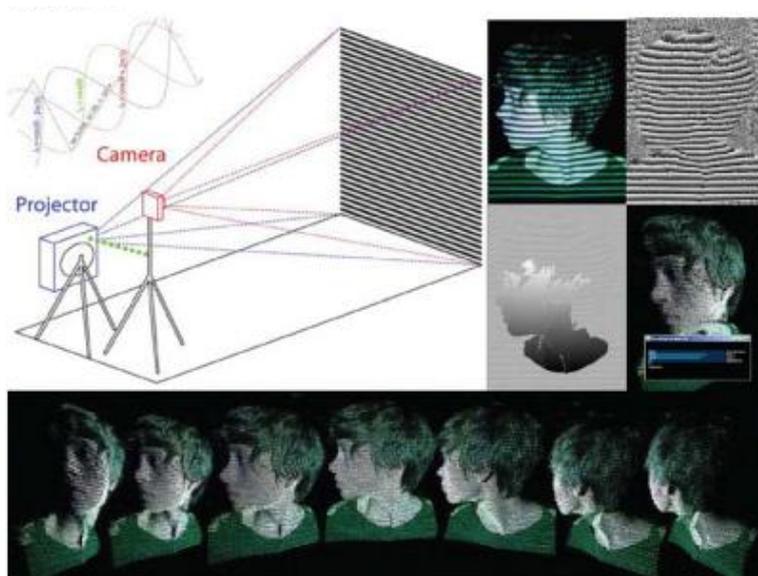


Figura 4 – Sistema com câmara de luz estruturada<sup>15</sup>.

O sensor *Microsoft Kinect* ou o *Asus Xtion* são exemplo de sensores que recorrem a este tipo de processo, através da emissão de luz infravermelha. Estes sistemas, com recurso a luz infravermelha que não estando no espectro de luz visível, asseguram a inexistência de interferências no decorrer do processo de deteção e captura da imagem.

A este tipo de sistemas estão associadas algumas vantagens, como é o caso do preço dos dispositivos ser relativamente baixo e acessível a um grande número de utilizadores.

<sup>15</sup> Imagem retirada do livro “*Meet the Kinect*” (Keane et al., 2012)

## 2.4.2 Nintendo Wii remote

O *Wii remote*<sup>16</sup> é um dispositivo de interação electrónico sem fios da consola de jogos *Wii* da *Nintendo* (Schreiber, von Wilamowitz-Moellendorff, & Bruder, 2009).

O comando (Figura 5) foi apresentado publicamente pela primeira vez no evento *Tokyo Game Show* em 2005. Recebeu desde aí atenções devido às suas características contrastantes com os outros controladores de jogos existentes (LaViola & Keefe, 2011). As características que mais se destacam no *Wii remote* é a capacidade de deteção de movimento e a forma intuitiva como a mão do utilizador o pode agarrar e manipular.



Figura 5 - O controlador *Wii remote*<sup>17</sup>.

Através da interface *bluetooth*<sup>18</sup> é possível ler e processar os dados transmitidos no computador. Estas características fazem deste sensor uma opção interessante no âmbito das interfaces humano-computador (Schreiber et al., 2009).

O comando *Nintendo wii remote* é um comando que permite captar os movimentos do jogador quando este o move. Ao possibilitar a captura dos movimentos permite ao utilizador interagir e manipular os elementos no ecrã. Apresenta três acelerómetros que efetuam uma recolha de dados nos eixos X,Y e Z, tal como ilustrado na Figura 6. O comando tem um conjunto de 9 botões, tipicamente associados aos comandos das consolas de jogos, e inclui também uma câmara de infravermelhos, um sistema de vibração e uma pequena coluna de som (LaViola & Keefe, 2011).

<sup>16</sup> <http://www.nintendo.com/wii/what-is-wii.jsessionid=DC72C2CDE04A33C0397B00F21A4AE89B#/controls>

<sup>17</sup> Imagem retirada de <http://www.nintendo.com/images/wii/whatis/controller-color-white.png>

<sup>18</sup> **Bluetooth** - é um padrão global de comunicação sem fios que permite a transmissão de dados entre dispositivos.

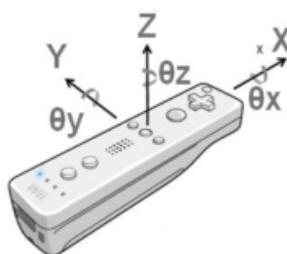


Figura 6 - Detalhe do sensor<sup>19</sup>.

O outro sensor, que comunica com o comando, está ligado diretamente à consola Wii e é constituído por um conjunto de LED infravermelhos que são captados pelo comando *wii remote* permitindo saber a posição X e Y do comando.

### 2.4.3 PlayStation Move Motion Controller

O *PlayStation Move Motion Controller* (Figura 7) é o sensor de movimento da consola de jogos *PlayStation 3* e foi anunciado no ano de 2009 no evento mundial de jogos *Electronic Entertainment Expo*<sup>20</sup> e começou a ser comercializado no Japão em Outubro de 2010.



Figura 7 - *PlayStation Move Motion Controller*<sup>21</sup>.

O sensor de movimento *PlayStation Move*<sup>22</sup> usa uma câmara que faz a deteção e o reconhecimento do utilizador. A bola presente na extremidade do sensor emite todas as cores presentes no sistema RGB<sup>23</sup> através do sistema de LED. A cor funciona como referência para a câmara *PlayStation Eye* (Figura 8), pois a cor da bola é destaca-se dos elementos de fundo do cenário, evitando assim problemas de reconhecimento e interferência de cores externas. Dentro do sensor de movimento estão alocados dois sensores, o primeiro é um acelerómetro que calcula

<sup>19</sup> Imagem retirada do artigo “*New interaction concepts by using the Wii remote*” (Schreiber, von Wilamowitz-Moellendorff, & Bruder, 2009)

<sup>20</sup> <http://www.e3expo.com/> - Evento anual dedicado aos videojogos.

<sup>21</sup> Imagem retirada de <http://webassetsj.scea.com/pscomauth/groups/public/documents/webasset/ps3-goldeneye-bundle-2-large.png>

<sup>22</sup> <http://pt.playstation.com/psmove/>

<sup>23</sup> **RGB** (*Red, Green, Blue*) – Abreviatura do sistema de cores aditivo composto por vermelho (*Red*), verde (*Green*) e azul (*Blue*).

no eixo do X, Y e Z e o segundo um magnetómetro que recolhe os dados respetivos ao campo magnético da Terra. O magnetómetro também calcula a orientação do sensor com o *PlayStation Eye*<sup>24</sup> (LaViola & Keefe, 2011).

Para o funcionamento do controlador de jogo *PlayStation Move* é necessário o *PlayStation Eye*, que captura os movimentos da bola colorida presente no comando. A comunicação entre o sensor de movimento e a consola *PlayStation* de jogos é efetuada via *bluetooth*.



Figura 8 - *PlayStation Eye*<sup>25</sup>.

#### 2.4.4 Microsoft Kinect para a Xbox

O *Microsoft Kinect* é um sensor de movimento desenvolvido para a consola de jogos *Xbox 360* e o seu principal objetivo anunciado é proporcionar uma nova e completa experiência de jogo aos seus utilizadores. Sem controlos adicionais, basicamente os utilizadores podem jogar e controlar o jogo através do seu próprio corpo. Para além de ser possível jogar sem comandos adicionais, existe também a possibilidade de vários utilizadores se encontrarem em jogo simultaneamente. Existem ainda outras funcionalidades associadas ao sensor, como o controlo por voz (Microsoft, 2011).

O lançamento do sensor da *Microsoft* ocorreu em 2010, no dia 4 de Novembro na América do Norte e a 10 de Novembro na Europa. Após estar disponível no mercado houve um conjunto de investigadores (individuais e coletivos) que se interessaram por investigar o potencial deste sensor para além da área do entretenimento, o que levou ao desenvolvimento de diversos projetos. Surgiram assim um conjunto de *hacks*<sup>26</sup> de que é exemplo o *driver*<sup>27</sup> para o sistema operativo Linux, anunciado por Héctor Martin no próprio dia 10 de Novembro de 2010, que permite aceder à

---

<sup>24</sup> <http://pt.playstation.com/ps3/news/articles/detail/item85718/Apresentamos-a-PlayStation-Eye/>

<sup>25</sup> Imagem retirada de <https://images-na.ssl-images-amazon.com/images/G/01/videogames/detail-page/B000VTQ3LU.01.lg.jpg>

<sup>26</sup> **Hacks** – Reconfiguração de um sistema ou dispositivo de forma não autorizada pelo proprietário.

<sup>27</sup> **Driver** - (em português controlador) é um *software*, que é instalado no computador, e permite a comunicação com dispositivos externos ao computador, ou seja, um driver é um *software* que permite ao sistema operativo usar as funcionalidades do dispositivo externo.

câmara RGB e ao emissor de infravermelhos do sensor. Assim surgiu o *driver OpenKinect*. Trata-se de um marco importante, pois este *driver* permitiu utilizar o sensor para além das funções associadas à consola de jogos, passando a ter uma utilização em computadores de tipo *general purpose*.

#### 2.4.4.1 Evolução /História do sensor *Kinect*

O *Kinect* é apresentado como um produto da *Microsoft* (Microsoft, 2011a) embora resulte de contributos de investigação de várias entidades. O *hardware* presente no dispositivo foi essencialmente desenvolvido pela empresa *PrimeSense*<sup>28</sup>, que trabalhou muito de perto com os desenvolvimentos ao nível do *software* de reconhecimento em profundidade da *Microsoft*.

Inicialmente o *Kinect* tinha como nome de código "*Project Natal*", e quando foi colocado à venda foi lançado com o nome "*Microsoft Kinect*". Foi um sucesso inicial e contou com 10 milhões de dispositivos vendidos no primeiro mês, fazendo história como o dispositivo periférico com um maior número de vendas num mês (Sidik, Sunar, Ismail, Mokhtar, & Jusoh, 2011)(Borenstein, 2011).

A empresa *Adafruit*<sup>29</sup> lançou um concurso para o desenvolvimento de um *driver* para o *Kinect*, com o principal objetivo de tornar o dispositivo acessível a partir do computador. Hector Martin ganhou o concurso com o desenvolvimento da *Open Kinect* (Giles, 2010). Foi o primeiro *driver* disponível que permitia aceder, através do computador, aos dados recolhidos pelo sensor. A criação deste *driver open source* rapidamente possibilitou o desenvolvimento de outros projetos de desenvolvimentos de aplicações tendo como base o sensor (Keane et al., 2012).

Mais tarde, a empresa *PrimeSense* disponibilizou os *drivers* do *software* desenvolvido para o *Kinect*, a *OpenNI*, permitindo assim o acesso à imagem em profundidade, deteção dos utilizadores e sistema de ossos em três dimensões. A disponibilização da *OpenNI* permitiu também aceder a um conjunto de informações que outros *drivers* não permitiam. Para além de fornecer mais informações, estas contêm um maior detalhe e profundidade.

*"The user data provided by OpenNI gives an application accurate information on the position of each user's joints (head, shoulders,*

---

<sup>28</sup> **PrimeSense** - <http://www.primesense.com/> A PrimeSense é uma empresa Israelita fundada em 2005 e foi selecionada pelo MIT com uma das 50 empresas mais inovadoras no ano de 2011.

<sup>29</sup> **Adafruit** - <http://www.adafruit.com/> é uma empresa com sede em Nova York e vende *kits open source* para projetos eletrónicos, como por exemplo o *arduino*.

elbows, wrists, chest, hips, knees, ankles, and feet) at all times while they're using the application. " (Borenstein, 2011).

#### 2.4.4.2 Especificações técnicas do Microsoft Kinect para a Xbox

O Kinect é uma *depth camera*, ou seja, um sensor de profundidade, devido ao *hardware* presente no sensor: uma câmara normal, um projetor de infravermelhos, um recetor de infravermelhos. A Figura 9, mostra com mais detalhe a estrutura interna do sensor.

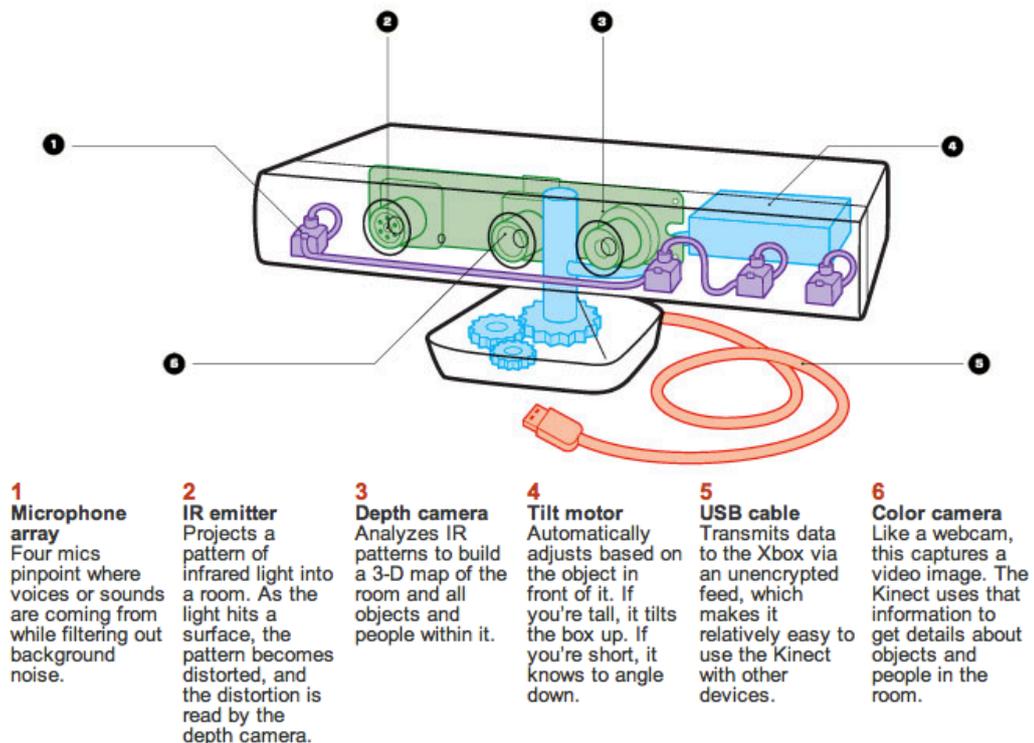


Figura 9 – Estrutura interna detalhada do sensor Microsoft Kinect<sup>30</sup>.

O projetor de infravermelhos projeta uma grelha de pontos que é captada pela câmara de infravermelhos. A profundidade de um objeto é calculada tendo em conta o tempo que a luz infravermelha demora a retornar ao sensor de infravermelho depois de refletida pelo objeto. Desta forma é construído um cenário a três dimensões. A técnica denomina-se por "*time of flight*" (TOF) e funciona como um sonar, ou seja, calculando quanto tempo a luz demora a chegar ao recetor, é possível saber a que distância o objeto está do sensor. A utilização de luz infravermelha apresenta ainda outra vantagem: uma vez que o sensor não foi desenvolvido para receber luz visível, a luz ambiente (do espaço onde está colocado o sensor) não interfere com a construção do cenário.

<sup>30</sup> Imagem retirada de [http://www.wired.com/magazine/2011/06/mf\\_kinect/all/1](http://www.wired.com/magazine/2011/06/mf_kinect/all/1)

A combinação do *hardware* e do *software* do *Kinect* (Figura 10) permite essencialmente, a geração de imagens a três dimensões (em movimento) dos elementos em frente à câmara e do reconhecimento dos movimentos e gestos do utilizador. O *Kinect* está equipado com tecnologia USB (*Universal Serial Bus*), que permite a sua ligação ao computador, da mesma forma que outros dispositivos (como um rato ou teclado).

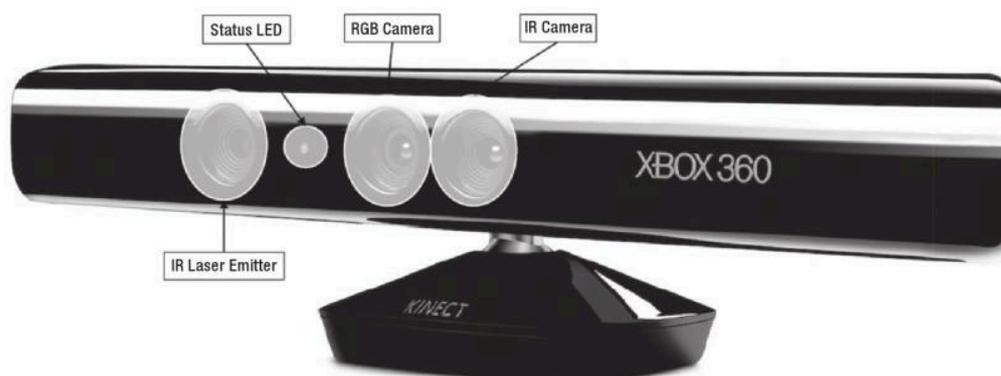


Figura 10 – Estrutura externa do Kinect para a Xbox<sup>31</sup>.

O sensor *Kinect* consegue identificar até dois esqueletos no mesmo cenário (ou seja, é possível estes dois utilizadores controlarem a aplicação através dos gestos) e consegue fazer um reconhecimento de até seis pessoas que estejam no ângulo e distância de visão do sensor. Em relação ao esqueleto que consegue identificar, o sensor distingue até vinte pontos do corpo humano do utilizador, no caso de este estar de pé, e até dez pontos no caso de se encontrar sentado (Microsoft, 2012).

#### 2.4.4.3 Usabilidade nas aplicações com a utilização do *Kinect*

Nielsen (Nielsen, 2010) refere que o "*Kinect presents a far more advanced gesture-based user experience than any previous system seen outside the fancy research labs.*" Nielsen indica o sensor *Kinect* como um importante dispositivo na promoção da experiência do utilizador, especificamente em interfaces controladas por gestos. Este autor menciona ainda que outra característica favorável deste sensor é ter um preço acessível. No mesmo artigo e para reforçar a ideia do potencial futuro deste tipo de dispositivos, Nielsen faz notar o espaçamento temporal que frequentemente existe entre a investigação e a disponibilização da tecnologia à generalidade dos utilizadores, dando como exemplo o tempo que decorreu entre a apresentação do rato por *Doug Engelbart* em 1964 e a sua comercialização apenas em 1984.

<sup>31</sup> Imagem retirada do livro "*Meet the Kinect*" (Keane et al., 2012)

Nesse mesmo artigo, Nielsen apresenta ainda uma listagem de fraquezas ao nível da usabilidade nas interfaces controladas por gestos (em específico dos jogos da Xbox), com utilização do sensor *Kinect*:

**Visibilidade** - Necessidade de memorizar as instruções, previamente ao jogo começar. Inexistência de *affordances*.

**Feedback** - Dificuldade em relacionar a causa-efeito. O utilizador tem dificuldade em perceber porque determinadas ações ocorrem e pode afetar a interação do utilizador com o sistema.

**Consistência e standards** - Não existem, ainda, *standards* para interações gestuais. O exemplo do pausar no jogo da Xbox - colocar o braço num ângulo de 45 graus. Trata-se de um gesto que não é natural ao utilizador.

Estes são alguns problemas de usabilidade apontados por Nielsen, resultado das suas experiências com os jogos da Xbox e com o *Kinect*.

#### 2.4.4.4 Exemplos de utilização do sensor em interfaces controladas por gestos

O sensor de movimento Microsoft *Kinect* está a ser utilizado em vários contextos como, por exemplo, na política. O primeiro-ministro Polaco efetuou uma apresentação multimédia controlada por gestos através da utilização do sensor *Kinect*, como se pode ver na Figura 11.



Figura 11 – Utilização do controlo por gestos pelo Primeiro Ministro Polaco.

No vídeo de apresentação<sup>32</sup> é possível observar como ele utilizou o sistema interativo para ajudar na visualização dos conteúdos que suportam exposição oral. O sensor *Kinect* fez a deteção do utilizador e dos seus gestos, através do qual foram controlados os conteúdos apresentados (“Polish Prime Minister uses the kinect,” 2011).

<sup>32</sup> Vídeo de apresentação do Primeiro Ministro Polaco <http://vimeo.com/32501348>

Outro contexto no qual o sensor está a ser utilizado é na área da manutenção e treino em situações simuladas (Figura 12). Um exemplo<sup>33</sup> é uma aplicação que permite a exploração e manipulação de elementos 3D em cenários de realidade virtual, num contexto industrial de transporte de resíduos radioativos.

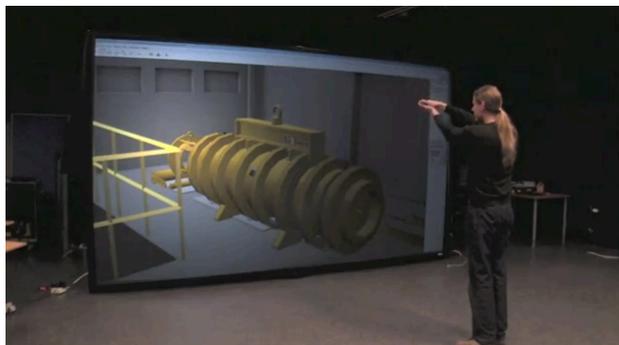


Figura 12 – Interfaces controladas por gestos em situações de simulação<sup>34</sup>.

O jogo *Kinect Juicy Adventure*<sup>35</sup>, desenvolvido pela empresa *Monolith Advertising*<sup>36</sup>, permite o reconhecimento do utilizador e o controlo da interface através dos gestos. O jogador encarna a personagem de um agricultor e este tem de recolher a maior quantidade de frutos e colocá-los nos cestos disponíveis. A Figura 13 mostra o exemplo de um jogador a interagir com o jogo.

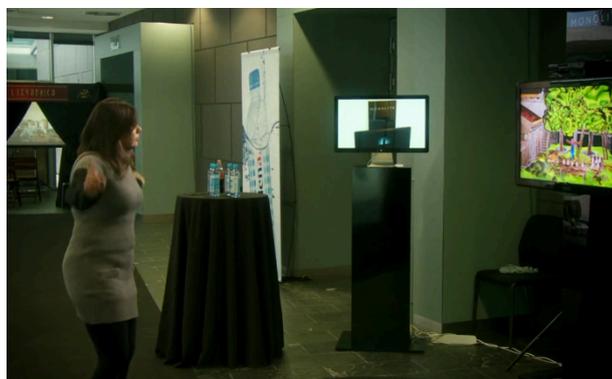


Figura 13 – Interfaces controladas por gestos nos jogos<sup>37</sup>.

<sup>33</sup> Treino e simulação [http://www.youtube.com/watch?feature=player\\_embedded&v=\\_904Cppeb4o](http://www.youtube.com/watch?feature=player_embedded&v=_904Cppeb4o)

<sup>34</sup> Imagem retirada do vídeo de apresentação da aplicação

<sup>35</sup> Vídeo do jogo <http://vimeo.com/31740172>

<sup>36</sup> Website da empresa <http://www.monolithadvertising.com/>, consultado a 8/05/2012

<sup>37</sup> Imagem retirada do vídeo de apresentação do jogo

### 2.4.5 Asus Xtion PRO

“The world’s first and exclusive professional color (RGB) PC motion sensing software development solution with plug-n-play USB design” (“Asus Xtion Pro Live,” 2012).

A primeira versão do sensor, denominado por Xtion Pro<sup>38</sup>, foi apresentada no evento *Consumer Electronics Association* (CES) em 2011. A Figura 14 ilustra a primeira versão do sensor mostrada ao público. Na apresentação foi referido também que a tecnologia *Primesense* (líder no mercado dos sensores com tecnologia *infrared structured light systems*) está na base da construção deste sensor, exclusivo para computador (Keane et al., 2012).



Figura 14 - Asus Xtion Pro<sup>39</sup>.

O **Asus Xtion PRO LIVE**<sup>40</sup> (Figura 15) é um sensor de movimento desenvolvido pela empresa Asus, para ser utilizado no computador. Este sensor está disponível apenas para *developers*, contrariamente ao sensor *Kinect da Microsoft* que pode ser encontrado numa qualquer loja de tecnologia. Este modelo mais recente inclui uma câmara RGB. É compatível com a *framework OpenNI* e o *middleware NITE* e disponibiliza um SDK (*Software Development Kit*) para o desenvolvimento de aplicações controladas por gestos. A empresa disponibiliza ainda um loja *online* onde é possível comercializar as aplicações desenvolvidas para o sensor. O sensor da *Asus* é muito semelhante ao sensor *Kinect*, tanto na tecnologia de suporte (*Primesense*) como mesmo no aspeto físico (Keane et al., 2012).



Figura 15 - Asus Xtion Pro Live<sup>41</sup>.

<sup>38</sup> [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO/#overview](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/#overview) consultado a 17/10/2012

<sup>39</sup> Imagem retirada de [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO/](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO/)

<sup>40</sup> [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO\\_LIVE/](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/) consultado a 17/10/2012

<sup>41</sup> Imagem retirada de [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO\\_LIVE/](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/)

### ***2.4.6 Aspetos a acautelar no desenvolvimento de aplicações com dispositivos de captura de movimento***

No desenvolvimento de aplicações com sistemas de reconhecimento de gestos, existe um conjunto de aspetos a acautelar no que diz respeito à utilização de sensores. Os sensores apresentam características muito específicas, pelo que é importante identificar os potenciais problemas associados.

Uma das primeiras recomendações apresentadas por Dan Saffer (Saffer, 2009) diz respeito à escolha do tipo de sensor que se pretende para a aplicação. O tipo de sensor, ou sensores no caso de haver uma conjugação de vários, determina o tipo de interação gestual que se pode ter na aplicação. Esta questão está relacionada com os vários aspetos técnicos presentes nos vários sensores, como é o caso da distância limite que o sensor consegue detetar (LaViola & Keefe, 2011).

A necessidade de calibrar o sensor ao nível da sensibilidade é outro aspeto importante no desenvolvimento de aplicações controladas por gestos. A calibração do sensor irá evitar que o sistema seja demasiado sensível e que responda sem o utilizador o desejar, ou de ter um sistema que não responde quando solicitado pelas interações do utilizador (Saffer, 2009).

Os sensores presentes nas aplicações controladas por gestos são muitas vezes sensíveis à luz ambiente, pelo que é recomendada a utilização da aplicação num ambiente com luz estável. O mesmo cuidado se aplica no caso da comunicação por voz em que o som ambiente pode perturbar o bom funcionamento da aplicação.

Outro conselho está relacionado com a prevenção de erros nas interfaces controladas por gestos. Uma das formas sugeridas de evitar erros, é a confirmação explícita das ações críticas realizadas pelo utilizador.

O *feedback* é um outro aspeto importante a ter em consideração no desenho de aplicações interativas, mas deve ser ainda mais potenciado quando existe um sistema de reconhecimento de movimentos. No caso de ocorrer um erro, uma falha do sistema, o utilizador deve saber o que está a acontecer com a aplicação, caso contrário pode pensar que está a proceder de forma incorreta (quando na verdade o problema reside na aplicação).

*“Usually, if there is any chance of recognition failure, the system should provide feedback on all recognition results, errors or otherwise. If possible, the system should provide some indication*

of the nature of any failure so that the user can modify his or her own behavior to meet the system's expectations.” (Wilson, 2008).

## 2.5 Tecnologias de reconhecimento 3D para o Kinect

Após a apresentação dos dispositivos de captura de movimento é necessário identificar as principais tecnologias que estabelecem a comunicação com os sensores e permitem fazer a deteção de gestos e renderização 3D.

Através das tecnologias de reconhecimento 3D para o sensor *Kinect* é possível aceder aos dados fornecidos pelo sensor. Estes *softwares* permitem recolher e processar as imagens do sensor disponibilizando essa informação ao *software* de desenvolvimento da aplicação final (onde se processa a interação controlada por gestos).

As subsecções seguintes apresentam os *softwares* de reconhecimento 3D disponíveis, as principais especificidades e funcionalidades.

### 2.5.1 Microsoft Kinect for Windows SDK

O *Microsoft Kinect for Windows SDK* (*Software Development Kit*) é um *software* de desenvolvimento de aplicações que permite aos programadores aceder e processar a informação recolhida pelo *Kinect*, permitindo assim a construção de interfaces controladas por gestos. O *Kinect*, aquando da sua conceptualização e desenvolvimento inicial, não foi pensado para funcionar na plataforma *Windows*. O *Kinect* foi desenvolvido para funcionar na consola de jogos *Xbox*. Existem muitas diferenças ao nível do suporte e implementação de *software* entre a plataforma *Xbox* e o *Windows*. Assim, de forma a ser possível aceder aos dados recolhidos pelo sensor, foi necessário adaptar um conjunto de aspetos relacionados com o *software* de forma a permitir aceder às informações no computador (Microsoft, 2011b).

O *Microsoft Kinect SDK* para o *Windows* permite aceder às informações recolhidas pelo sensor, como as imagens a cores, imagens em profundidade e áudio. A Figura 16 mostra a arquitetura do funcionamento do *Kinect* no *Windows*.

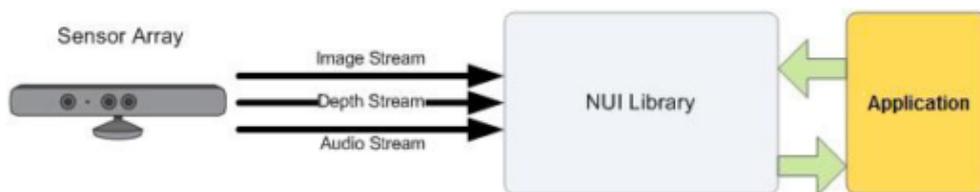


Figura 16 - Resumo da arquitetura do funcionamento do *Kinect* no *Windows* (Microsoft, 2011b).

*“The kit makes it easier for the academic research and enthusiast communities to create rich experiences by using Kinect for Xbox 360 sensor technology on a PC that is running the Windows 7 operating system. This beta SDK is provided by Microsoft Research to enable researchers and programmer enthusiasts to explore the development of natural user interfaces.”* (Microsoft, 2011).

Os requisitos de *hardware* para o funcionamento do *Microsoft Kinect SDK* num computador são os seguintes:

- Processador 32 bit (x86) ou 64 bit (x64)
- Processador Dual-core 2.66-GHz ou superior
- USB 2.0 bus
- 2 GB RAM
- *Kinect* para a Xbox 360

Requisitos de *software*:

- *Microsoft Visual Studio 2010*
- *.NET Framework 4.0*

No caso de se pretender aceder ao funcionamento do *Microsoft Kinect SDK* no computador é necessário conectar-lhe o sensor via USB. A Microsoft disponibiliza a informação relativa ao *Skeletal Viewer* (visualização da estrutura em “esqueleto” atribuída ao corpo de cada utilizador, formada através da imagem proveniente do sensor) sendo necessária a instalação dos seguintes componentes:

- *Microsoft DirectX SDK* – a versão mais recente
- *Runtime for Microsoft DirectX 9*

O *Microsoft Kinect SDK* possibilita desenvolver as aplicações com as linguagens de programação C++, C# ou *Visual Basic*. De momento, o licenciamento do SDK é limitado a aplicações não comerciais.

### **2.5.2 OpenNI (Open Natural Interaction)**

*OpenNI* ou *Open Natural Interaction* (Figura 17) é uma *framework* de desenvolvimento de aplicações controladas por gestos. Esta *framework* caracteriza-se essencialmente por ser multiplataforma (*Windows*, *Linux* e *Mac OS X*) e tem suporte para a programação em varias

linguagens de programação. A *framework* foi tornada pública em Novembro de 2010. O seu principal objetivo foi o de melhorar a compatibilidade e interoperabilidade de dispositivos de interface natural, aplicações e *middleware*<sup>42</sup>, ou seja, a disponibilização de uma API (*Application Programming Interface*) para permitir o desenvolvimento de aplicações que façam recurso a voz e gestos para a interação. A forma como esta *framework* está estruturada e desenvolvida permite ser independente do sensor que está a ser utilizado (“Open NI User Guide,” n.d.).



Figura 17 – Logótipo *OpenNI* (“OpenNI,” n.d.).

A Figura 18 apresenta a estrutura em camadas da *framework OpenNI*. A camada dos sensores (*Hardware Device Sensor*) representa os dispositivos de *hardware* necessários para a captura de áudio e vídeo. A camada correspondente à interface *OpenNI*, representa a *OpenNI* e as interfaces de comunicação entre os sensores e as interfaces de *middleware* que analisam os dados dos sensores. A última camada representada no esquema é a de aplicação e representa o *software* que permite desenvolver aplicações controladas por gestos e voz para a interação.

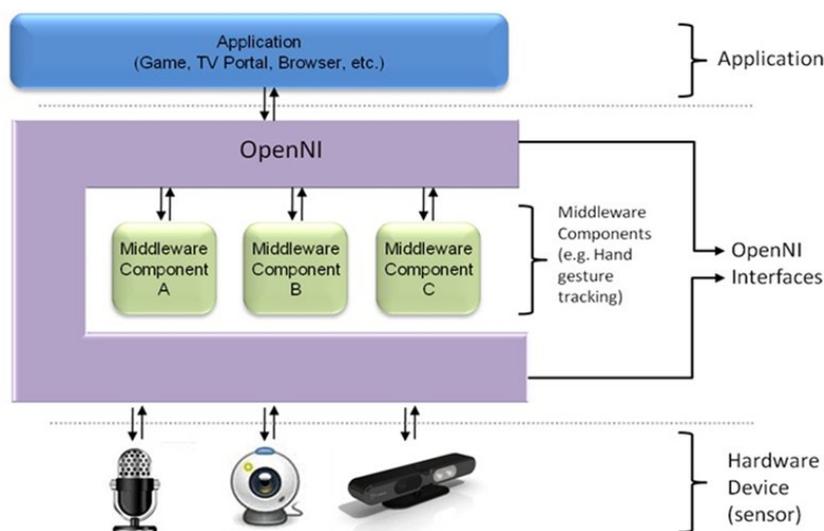


Figura 18 – Estrutura da *framework OpenNI* (“Open NI User Guide,” n.d.).

<sup>42</sup> **Middleware** - *Software* essencialmente utilizado permitir a troca de informações e dados entre programas de diferentes protocolos de comunicação, plataformas ou sistemas operativos. O *middleware* permite criar uma ponte de troca de dados entre dois *softwares* distintos.

A *framework open source OpenNI* permite escrever aplicações para interação natural. Esta *framework* está escrita originalmente em C++ mas agora também permite programar em C#. A API serve como uma camada intermediária entre dispositivos de baixo nível (câmaras, sensores, microfones e etc.) e soluções *middleware* de alto nível (algoritmos de visão por computador, por exemplo).

Os *drivers* fornecidos pela *OpenNI* permitem aceder às informações provenientes dos sensores de áudio, vídeo e profundidade do sensor do *Microsoft Kinect*.

*“OpenNI supplies a set of APIs to be implemented by the sensor devices, and a set of APIs to be implemented by the middleware components. By breaking the dependency between the sensor and the middleware, OpenNI’s API enables applications to be written and ported with no additional effort to operate on top of different middleware modules (“write once, deploy everywhere”).*

*“The OpenNI standard API enables natural-interaction application developers to track real-life (3D) scenes by utilizing data types that are calculated from the input of a sensor (for example, representation of a full body, representation of a hand location, an array of the pixels in a depth map and so on). Applications can be written regardless of the sensor or middleware providers” (“Open NI User Guide,” n.d.).*

### 2.5.3 OpenKinect

O *OpenKinect* é uma comunidade aberta que desenvolve bibliotecas que permitem aceder aos dados recolhidos pelo sensor de movimento da *Microsoft*. O *OpenKinect* foi um resultado do concurso realizado pela empresa *Adafruit*, onde Héctor Martin apresentou a sua solução para aceder aos dados do *Kinect* através do computador. A Figura 19 mostra o logótipo da comunidade de desenvolvimento.



Figura 19 – Logótipo do *OpenKinect* (OpenKinect, 2011).

Ao vencer o concurso, Héctor continuou o desenvolvimento destas bibliotecas às quais chamou *OpenKinect*. Colocou as bibliotecas em acesso público na comunidade de programadores

GitHub<sup>43</sup>, o que permitiu à comunidade consultar, partilhar e colaborar no desenvolvimento (Giles, 2010).

*“OpenKinect is an open community of people interested in making use of the amazing Xbox Kinect hardware with our PCs and other devices. We are working on free, open source libraries that will enable the Kinect to be used with Windows, Linux, and Mac.”*  
(OpenKinect, 2011).

Esta *framework* está em contínuo desenvolvimento e sempre que possível novas funcionalidades são adicionadas por forma a criar uma *framework* mais estável e completa.

### 2.5.4 ZigFu

O *ZigFu* é um *software* de apoio ao desenvolvimento de aplicações controladas por gestos com base em sensores de movimento, como os sensores *PrimeSense*, *Microsoft Kinect* para a *Xbox* e *Asus Xtion Sensor*.

Este pacote de *drivers* permite instalar vários componentes necessários para a recolha da informação proveniente do sensor. O pacote de instalação (tal como disponível em 18 de Janeiro de 2012) inclui a última versão dos seguintes componentes: *OpenNI*, *NITE*, *Sensor*, *SensorKinect*. Esta última versão do *NITE* permite o desenvolvimento de aplicações sem ser necessário fazer calibração do utilizador, ou seja, o utilizador não necessita de estar numa pose específica para o sensor fazer a deteção.

O pacote *ZigFu* inclui os componentes identificados nas seguintes versões:

- *NITE*: 1.5.2.21
- *OpenNI*: 1.5.2.23
- *Sensor*: 5.1.0.41 (*Asus Xtion*)
- *SensorKinect*: 5.1.0.25 (*Kinect*)

No site do *ZigFu*<sup>44</sup> está disponível um conjunto de pacotes de acordo com as diferentes necessidades de desenvolvimento dos utilizadores. Se o desenvolvimento for orientado para a *Web*, o pacote de instalação ***ZigJS*** (*Client-side JavaScript*) permite desenvolver aplicações para a *Web* controladas por gestos. O pacote de instalação ***ZigUnity3D*** permite instalar os componentes

---

<sup>43</sup> Website da comunidade Github <https://github.com/>, consultado a 7/09/2012

<sup>44</sup> Website do Zigfu <http://zigfu.com/>, consultado a 10/01/2012

necessários para o desenvolvimento de aplicações controladas por gestos no *Unity3D*. Esta nova versão do pacote de instalação funciona simultaneamente com a *framework OpenNI* e com o *Kinect SDK*. Está, ainda, disponível um pacote direcionado para aplicações desenvolvidas com o *software adobe flash*, o **ZigFlash**.

O pacote relativo ao *Unity3D* contém um conjunto de scripts que podem ser anexados aos componentes 3D presentes no *Unity3D*, criando assim a ligação entre os dados provenientes do sensor e o modelo 3D. Através da comunicação Sensor – Pacote *ZigFu – Unity3D* é possível efetuar a captura de movimentos do utilizador que será representada no avatar.

### **2.5.5 Kinetic Space**

O *Kinetic Space* é uma *framework* de reconhecimento 3D que funciona com os sensores *PrimeSense*, *Microsoft Kinect* para a *Xbox* e o *Asus Xtion Sensor*.

Este *software* procede a uma interpretação dos movimentos gerados pelo utilizador e processa esses dados produzindo informação de acordo com um reportório de eventos, tais como clicar, puxar, entre outros.

Existe, no entanto, ainda reduzida informação disponível e poucos projetos desenvolvidos com recurso a esta *framework*.

### **2.5.6 NI Mate**

*“Kinect based motion capture and sound control software.” (“NI Mate,” 2012).*

A *framework* de reconhecimento 3D **NI (natural interaction) Mate** teve a primeira versão de teste disponível para *download*, para o público em geral, no dia 26 de Janeiro de 2012. Esta *framework* permite recolher os dados fornecidos pelo sensor de movimento *Kinect* e, posteriormente, usar esses dados para a animação e o controlo de elementos 3D.

Este *plug-in* permite a integração com os seguintes *softwares*: **Blender**, **Cinema 4D**, **Animata** e permite ainda a conversão dos dados para **MIDI** ou **OSC**.

*“Software that takes real-time motion capture data from a Kinect or other supported device and turns it into two industry standard protocols: OSC (Open Sound Control) and MIDI (Musical Instrument Digital Interface) “(“NI Mate,” 2012).*

## 2.6 Interação e interfaces controladas por gestos

*“The body is an amazingly flexible tool with a wide variety of gestures. The hands alone have the potential for hundreds of possible configurations that could be used to trigger a system response. It is just up to designers to employ them correctly and not to overstrain users.” (Saffer, 2009).*

Independentemente da tecnologia utilizada para a captura dos gestos, o processo operacional de recolha das informações é o mesmo. O processo começa com o reconhecimento do movimento do utilizador, através dos sensores, e o *software* tenta identificar a correspondência desses movimentos com padrões de movimento pré-definidos. No caso de um gesto ser identificado, é verificado o seu significado sendo despoletado no sistema o evento que lhe esteja associado.

Nesta secção é discutido o conceito de gesto e as suas principais características, sendo também apresentada uma breve descrição sobre a evolução das interfaces controladas por gestos. São também discutidos alguns problemas de usabilidade associados a este tipo de interfaces.

### 2.6.1 Gestos

*“Gesture is a motion of any part of the body that contains information” (Svedström, 2010).*

O ser humano usa os gestos diariamente sempre que comunica com outras pessoas ou interage com objetos. Os gestos são uma componente inerente à comunicação não-verbal humana e à interação com o mundo físico (Svedström, 2010).

Um gesto pode ser definido como uma posição específica da mão, uma trajetória espacial das mãos, apontar ou outro tipo de movimento que indique um objeto ou o equivalente ao movimento de qualquer outra parte do corpo referida a um determinado contexto (Wilson, 2008).

Thomas Svedström (Svedström, 2010) define gesto como sendo o movimento de qualquer parte do corpo que contenha informação. As interfaces controladas por gestos permitem uma interação mais dinâmica e expressiva no controlo de atividades que necessitem de uma resposta rápida. Este autor refere ainda que, em determinados contextos, o controlo de interfaces por gestos providencia uma interação mais intuitiva do que a tradicional utilização do rato e teclado.

No contexto das interfaces gestuais, Dan Saffer (Saffer, 2009) define gesto como qualquer movimento físico protagonizado pelo utilizador, que um sistema digital pode reconhecer e responder sem a necessidade do suporte dos tradicionais dispositivos de entrada.

Nielsen et al. (M. Nielsen, Störring, Moeslund, & Granum, 2004) classificaram os gestos em duas categorias: gestos estáticos e gestos dinâmicos. Gestos estáticos, como o próprio nome indica, é uma postura como, por exemplo, uma mão aberta parada que é reconhecida pelo sistema. Gestos dinâmicos incluem movimentos, uma ação que decorre no tempo, como, por exemplo, acenar com a mão para desencadear um determinado evento.

A *Microsoft* (Microsoft, 2012) considera, por outro lado, que os gestos podem ser de três tipos distintos: gestos estáticos, dinâmicos e contínuos. Os gestos estáticos caracterizam-se por serem poses ou posturas em que o utilizador “segura” a mão, braço ou parte do corpo em questão, até que a pose seja reconhecida pelo sistema. Os gestos dinâmicos dão *feedback* ao utilizador enquanto estão a ser executados e caracterizam-se por serem limitados no tempo e no espaço, ou seja, têm início e fim delimitado. Os gestos do tipo contínuo são gestos, ou movimentos, que são constantemente reconhecidos e detetados pela aplicação.

Alguns gestos caracterizam-se por serem intuitivos, ou seja, que o utilizador executa de forma natural e intuitiva de acordo com a sua interpretação individual. Outros gestos podem ser pré-definidos sendo necessário o utilizador aprendê-los para, posteriormente, os poder utilizar.

No contexto desta investigação, um gesto é definido como qualquer movimento físico das mãos, braços, rosto e corpo com a intenção de transmitir informação (Beurden & IJsselsteijn, 2011).

### **2.6.2 Indicadores de caracterização dos gestos**

As interfaces gestuais, quer se considerem as *touchscreen* ou as *free-form*, apresentam especificidades diferentes, mas relativamente aos gestos e suas particularidades existem poucas diferenças. Em seguida apresentam-se os principais indicadores utilizados para caracterizar os gestos, tal como proposto por Dan Saffer (Saffer, 2009).

**Presença** – O gesto necessita de ser executado para despoletar a interação. Em determinados sistemas, a simples presença próxima do utilizador é suficiente para desencadear algum evento na aplicação.

**Duração** – A duração é o intervalo de tempo desde momento em que o gesto é detetado até ao momento em que a execução desse gesto termina. Dependendo da natureza das aplicações, do tipo de suporte ou do *input*, esta característica reveste-se de particular relevância. Por exemplo, no caso dos jogos este aspecto tem uma maior relevância sendo necessária, por questões de jogabilidade, uma correspondência rigorosa entre a duração do gesto protagonizado pelo utilizador e a representação do gesto feita pelo sistema.

**Posição** – A posição define onde é que o gesto é executado, tipicamente num eixo X/Y, mas, em determinados contextos, o eixo Z também é incluído nos movimentos.

**Movimento** – Indica se o utilizador está, ou não, a deslocar-se.

**Pressão** – Esta característica dos gestos é associada apenas a gestos executados em dispositivos *touchscreen* e indica o grau de pressão que o utilizador está exercer sobre o dispositivo.

**Tamanho** – Através da altura e largura da área em que o gesto ocorre é possível medir o tamanho desse gesto.

**Orientação** – Esta característica diz respeito à orientação espacial do utilizador quando executa o gesto. A orientação é determinada através de pontos fixos, como o ângulo do utilizador em relação ao sensor.

**Inclusão de objetos** – Algumas interfaces gestuais permitem ao utilizador a utilização de objetos. Sistemas mais simples interpretam estes objetos como uma simples extensão do corpo do utilizador, mas sistemas mais complexos podem reconhecê-los e permitir a sua utilização no contexto da aplicação.

**Número de pontos de contacto/combinções** – No contexto de aplicações *touchscreen*, o multitoque permite que o utilizador utilize mais do que um dedo ou mão simultaneamente para controlar a aplicação. Existe ainda a possibilidade de usar combinações de gestos que resultam em eventos específicos (por exemplo usar as duas mãos para redimensionar uma imagem, colocando cada mão num canto da imagem e arrastando-as até obter o tamanho de imagem desejado).

**Sequência** – Dois gestos executados em separado podem ter um resultado diferente de quando são executados um seguido do outro. Esta característica implica um sistema de reconhecimento mais complexo e com capacidade de registar os estados anteriores.

**Numero de participantes** – O numero de participantes depende do tipo de atividade, dos dispositivos e se a aplicação deteta os diferentes utilizadores. Esta característica permite a interação em atividades colaborativas.

### ***2.6.3 Evolução das interfaces controladas por gestos***

Uma das grandes diferenças que se verifica entre as interfaces controladas por gestos e as anteriores (interfaces gráficas ou linha de comandos) é que as gestuais permitem novas formas de

interação com a aplicação. Para além da interação com o rato e teclado, que permite escrever ou fazer *scroll*, a interação através dos gestos permite a utilização do corpo humano como dispositivo de entrada de dados (Saffer, 2009).

*“As sensors and microprocessors become faster, smaller, and cheaper, reality has started to catch up with the vision, although we still have quite way to go.”* (Saffer, 2009).

O primeiro dispositivo tátil, o *Elograph*<sup>45</sup>, ilustrado na Figura 20 (*electronic graphics*) foi apresentado por Samuel C. Hurst em 1971. Em 1974, a empresa *Elographics* desenvolveu a tecnologia *five-wire*<sup>46</sup>, sendo ainda atualmente uma das tecnologias tácteis mais utilizadas.



Figura 20 – Elograph<sup>47</sup>.

A empresas *Elographics* e *Siemens* desenvolveram conjuntamente o *Accutouch* (Figura 21), considerado o primeiro dispositivo tátil verdadeiro.

<sup>45</sup> <http://www.elotouch.com/AboutElo/History/default.asp> consultado a 6 de dezembro de 2011

<sup>46</sup> <http://www.elotouch.com/Products/Touchscreens/AccuTouch/default.asp> consultado a 6 de dezembro de 2011

<sup>47</sup> Imagem retirada de <http://www.elotouch.com/images/aboutelo/gdt100.jpg>

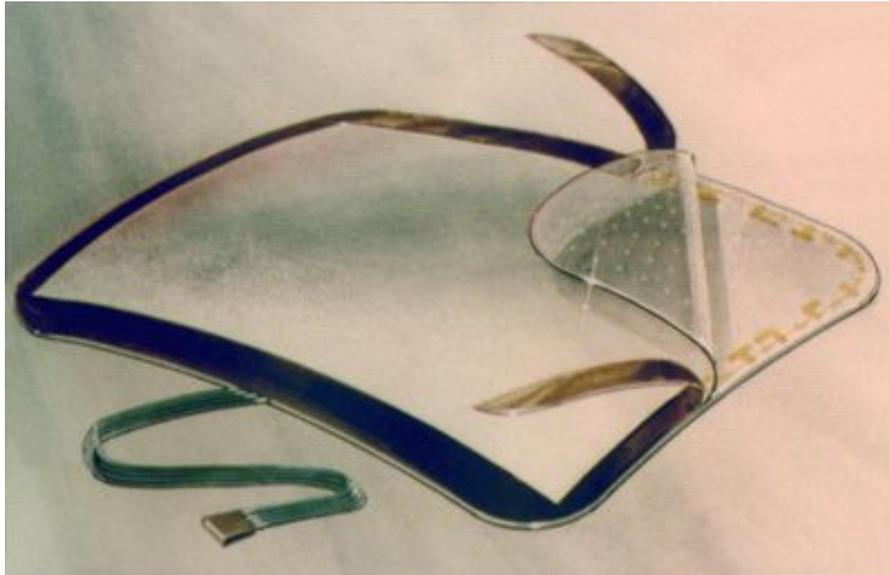


Figura 21 – *Accutouch*<sup>48</sup>.

Mais tarde, nos anos 70, Myron Krueger criou um sistema constituído por um conjunto de projetores, câmaras de vídeo e outros equipamentos no qual permitia aos utilizadores interagirem sem a necessidade de utilização nas mãos de qualquer tipo de equipamento especial, como a utilização do rato ou luvas especiais (M. Krueger & Gionfriddo, 1985).

Myron Krueger refere em (M. W. Krueger, 1993) que idealizou um sistema em que o utilizador, em vez de se sentar e utilizar o rato e teclado, interagisse com o sistema através da utilização do corpo.

*“Rather than sitting down and moving only your fingers, you interacted with your body. In one demonstration, a simple 3D scene was displayed on a projection screen.”* (M. W. Krueger, 1993).

Nos anos 70, Krueger projetou um sistema que permitisse a comunicação através de uma rede de comunicação entre indivíduos em locais físicos separados, mas onde a interação virtual aparentasse ser real. O resultado desta ideia foi uma aplicação na qual dois utilizadores podiam interagir e partilhavam o resultado dessa interação (M. W. Krueger, 1993). Assim nasce o projeto VIDEOPLACE (Figura 22), um espaço de partilha e comunicação partilhado e com a possibilidade de os gráficos serem manipulados por gestos.

---

<sup>48</sup> Figura retirada de <http://www.elotouch.com/images/aboutelo/accusk.jpg>

“*VIDEOPLACE is a computer graphic environment It may be alone people at different locations. In addition, or there may be images of other there may be graphic objects and creatures which interact with the participant's image.*” (M. Krueger & Gionfriddo, 1985).



Figura 22 – VIDEOPLACE<sup>49</sup>.

Em 1982, Nimish Mehta, no âmbito da sua dissertação de mestrado, desenvolveu o que é considerado pelos especialistas o primeiro sistema *multitouch*. A combinação da deteção da pressão dos dedos na interface com o processamento da imagem permitiu mais do que um contacto simultaneamente com o sistema possibilitando assim a manipulação de alguns elementos gráficos com as duas mãos (Saffer, 2009).

O Hewlett-Packard 150 foi possivelmente o primeiro computador com sistema *multitouch* para uso pessoal, tendo sido inicializada a sua comercialização em 1983 (Figura 23). Este sistema permitia mover o cursor e seleccionar botões no ecrã, ainda que a precisão fosse reduzida (Saffer, 2009).

---

<sup>49</sup> Figura retirada do artigo “*Videoplac*e - An artificial reality” (M. Krueger & Gionfriddo, 1985)



Figura 23 - Hewlett-Packard 150<sup>50</sup>.

Nos anos 90, Pierre Wellner desenvolveu a *Digital Desk* (Figura 24). Através da utilização de câmaras de vídeo e projetores, vários conteúdos são projetados na mesa e podem ser manipulados pelos utilizadores (Saffer, 2009).

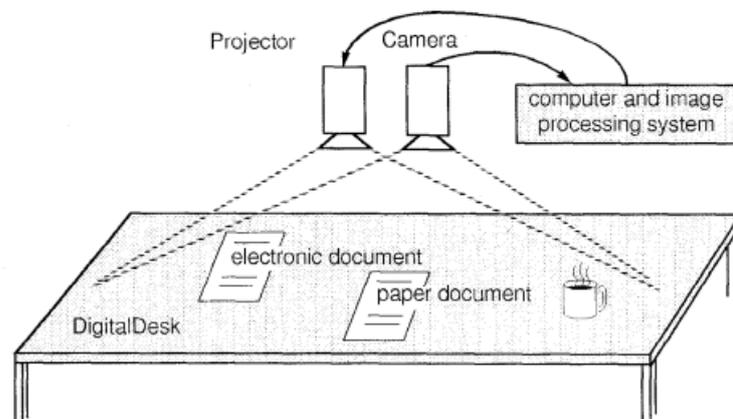


Figura 24 – Esquema do funcionamento da *Digital Desk*<sup>51</sup>.

Em 2001, a empresa *LionHead Studios* lançou uma luva especial, o primeiro produto comercial destinado ao controlo de interfaces no âmbito dos jogos (Saffer, 2009).

<sup>50</sup> Imagem retirada de <http://www.hp.com/hpinfo/abouthp/histnfacts/museum/personalsystems/0031/0031threeqtr.html>

<sup>51</sup> Imagem retirada do artigo “*The DigitalDesk calculator: tangible manipulation on a desk top display*” (Wellner, 1991)

*“We've entered a new era of interaction design. For the past 40 years, we have been using the same human-computer interaction paradigms that were designed by the likes of Doug Engelbart, Alan Kay, Tim Mott, Larry Tesler, and others at Xerox PARC in the 1960s and 1970s. Cut and paste. Save. Windows. The desktop metaphor. And so many others that we now don't even think about when working on our digital devices. These interaction conventions will continue, of course, but they will also be supplemented by many others that take advantage of the whole human body, of sensors, of new input devices, and of increased processing power. We've entered the era of interactive gestures.”* (Saffer, 2009).

#### **2.6.4 Problemas de Usabilidade nas interfaces controladas por gestos**

*“When developing a gesture interface, the objective should not be “to make a generic gesture interface”. A gesture interface is not universally the best interface for any application. The objective is “to develop a more efficient interface” for a given application.”* (M. Nielsen et al., 2004).

A usabilidade, no âmbito da interação humano-computador e do design de interfaces, caracteriza a simplicidade ou a facilidade com que uma interface pode ser utilizada. Os princípios de usabilidade dão indicações de como desenvolver aplicações que permitam a criação de uma interface fácil de aprender, eficiente, e que reduza a probabilidade de o utilizador cometer erros. A *International Organization for Standardization* (ISO), define a usabilidade como *“Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.”* (ISO, 1998).

A usabilidade está relacionada com a capacidade da aplicação permitir ao utilizador cumprir com as suas metas de interação com a aplicação. Quando se desenvolve uma interface controlada por gestos, esta deverá ser o mais eficiente possível de acordo com os objetivos e os contextos propostos.

Tradicionalmente, conforme referido por Nielsen (2004), a usabilidade tem sido descrita com um conjunto de atributos, que permitem fazer com que seja algo mensurável e que possa ser avaliada (M. Nielsen et al., 2004). Os princípios da usabilidade são, naturalmente, muito importantes para o desenvolvimento de interfaces controladas por gestos e são os seguintes:

- **Learnability:** Este princípio está relacionado com o tempo e o esforço necessários para serem atingidos os objetivos de utilização com uma determinada performance. A utilização de uma aplicação nova implica um tempo de aprendizagem.
- **Efficiency:** Uma forma de avaliar quantitativamente a eficiência é através da medição do tempo que um utilizador que já tenha aprendido a trabalhar com a aplicação necessita para executar um conjunto de tarefas específicas. por
- **Memorability:** As regras de utilização do sistema devem poder ser facilmente recordadas após um tempo sem o utilizador as executar, para que utilizador não tenha de reaprender a funcionar com o sistema após algum tempo sem o fazer. No âmbito das interfaces controladas por gestos, é importante referir que se os gestos forem naturais o esforço cognitivo de memorização pedido ao utilizador é menor do que noutro tipo de paradigmas de interação (Besada, Bernardos, & Portillo, 2011).
- **Errors:** O sistema deve estar desenvolvido para que os utilizadores cometam o menor número possível de erros. A aplicação deve assegurar que a possibilidade de voltar atrás (recuperação) deve ser tão fácil para o utilizador quanto possível. Ao nível das interfaces controladas por gestos, é importante definir claramente as ações por forma a que os sensores de movimento não cometam erros. Nestas interfaces o conjunto de gestos através do qual a interação se processa deverá ser suficientemente diversificado para evitar erros de interpretação pelos sensores de movimento utilizados.
- **Satisfaction of use:** Este princípio é muito subjetivo de avaliar, pois pretende caracterizar a satisfação de utilização do sistema pelo utilizador. No caso das interfaces controladas por gestos, que envolvem a realização de movimentos corporais durante determinados períodos de tempo, um dos principais aspectos que pode influenciar este parâmetro é o eventual cansaço a que o utilizador está sujeito.

Tendo em conta a definição de usabilidade e a necessidade de contemplar os princípios que a caracterizam, uma aplicação com reconhecimento de gestos deverá oferecer ao utilizar uma interface fácil de memorizar, intuitiva e que evite que os utilizadores cometam erros (Besada et al., 2011).

*“With a gesture-based input it is even more necessary to simplify the dialogue and sequencing than the conventional input methods, as there will be less possible commands to give.” (Besada et al., 2011).*

### 2.6.5 Design de interação e interfaces com o sensor Kinect

Nesta subsecção apresentam-se algumas recomendações para o design de interação e interfaces com gestos, especificamente com recurso ao sensor *Microsoft Kinect*. Conceptualizar e implementar uma interface *free-form* controlada por gestos é um processo complexo, as tecnologias são recentes e os paradigmas de interação não estão ainda claramente definidos. É, assim, importante ter em conta várias considerações referentes à utilização dos gestos, à escolha dos mecanismos de interação e à construção das interfaces, para evitar que a aplicação seja confusa ou de utilização complexa.

A interação através dos gestos é uma forma divertida de interagir com o sistema do ponto de vista do utilizador, mas apresenta também dificuldades que precisam de ser ultrapassadas: fazer com que a interação natural seja de facto natural para o utilizador, dar-lhe *feedback* para este não se sentir perdido durante o uso da aplicação, são alguns dos aspectos mais importantes no que concerne aos gestos e à sua utilização (Microsoft, 2012).

Aquando da utilização de gestos em aplicações, existem vários aspetos a considerar para o correto desenvolvimento da aplicação.

- O utilizador deve perceber quando o sensor está em funcionamento e a detetar os movimentos;
- Qual a distância e o ângulo até onde o sensor consegue detetar o utilizador;
- Quando é que o utilizador pode começar a interagir com a aplicação.

O utilizador deve ser capaz de aprender e executar os controlos básicos da aplicação de uma forma simples e rápida. Os gestos devem ser confortáveis de executar e humanamente possíveis. Quando os gestos são executados a aplicação deverá responder e/ou dar *feedback*.

A Figura 25 ilustra alguns aspetos a ter em conta nos gestos, de forma a ser possível evitar conflitos entre gestos ou dificuldades de deteção pelo sistema. A direção dos gestos deve ser tida em conta (dois gestos na mesma direção podem ser mais difíceis de detetar pelo sistema). O trajeto dos gestos deve mudar, especialmente se dois gestos tiverem a mesma direção. Os gestos devem ter um ponto de início e de fim, e o tempo expectável de duração do gesto deve também ser limitado.

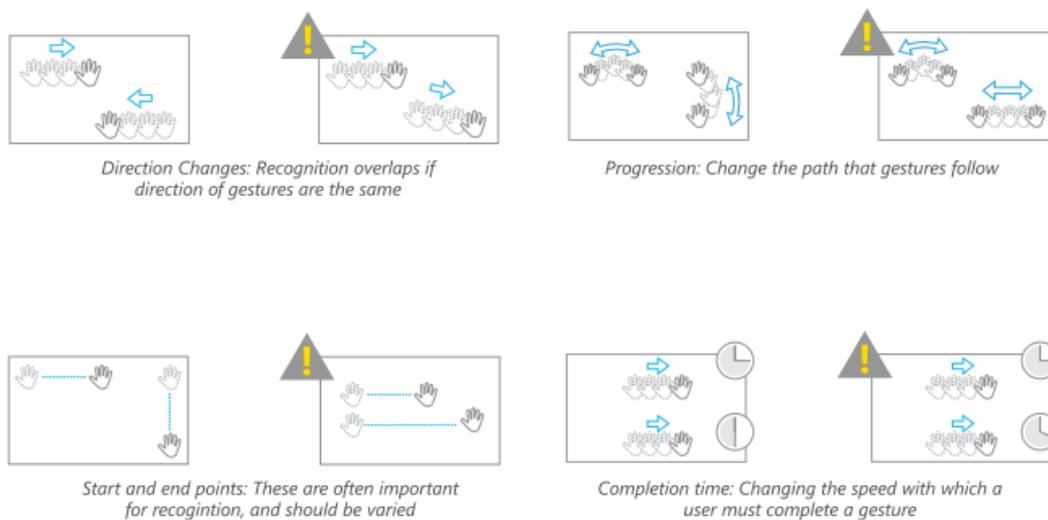


Figura 25 – Direção, progressão e duração dos gestos<sup>52</sup>.

O número recomendado total de gestos na mesma aplicação é de 6 de acordo com o manual de recomendações da Microsoft (Microsoft, 2012), mais gestos podem ser difíceis de lembrar e aumentar o esforço cognitivo. Os gestos devem ser de execução natural para o utilizador: se forem gestos intuitivos, é-lhe mais fácil executá-los e memorizá-los. Devem ser possíveis de realizar com a mão direita ou com a mão esquerda, pois o utilizador pode trocar de mão para evitar a fadiga, ou pode ser esquerdino. No caso de gestos que necessitem das duas mãos, os gestos devem ser simétricos, pois tornam-se mais fáceis de executar. A Figura 26 mostra um utilizador a efetuar gestos com uma mão (lado esquerdo da figura) e gestos com as duas mãos (lado direito da figura), com uma colocação simétrica das mãos.

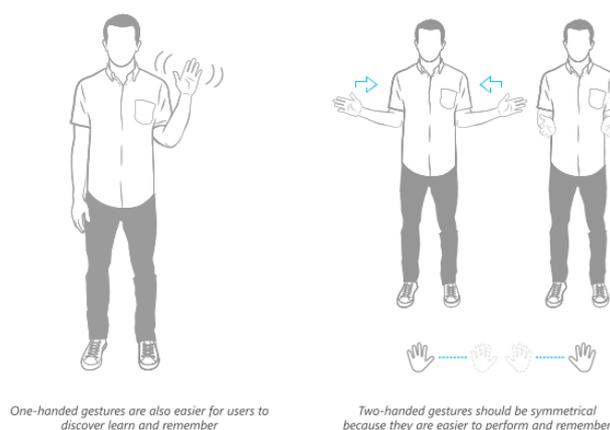


Figura 26 – Gestos com uma mão e gestos com duas mãos<sup>53</sup>.

<sup>52</sup> Imagem retirada do documento “Kinect for Windows - Human Interface Guidelines” (Microsoft, 2012)

Se os gestos forem de difícil execução ou provocarem muito cansaço, o utilizador pode ter uma performance indesejável, os níveis de interesse podem diminuir e proporcionar uma má *user experience*. Assim, a postura do utilizador afeta a sua performance pelo que devem ser considerados quais os movimentos mais confortáveis para o utilizador. A Figura 27 mostra os limites físicos do utilizador. Os gestos definidos para a aplicação devem ter sempre em consideração o público-alvo e as suas limitações físicas.

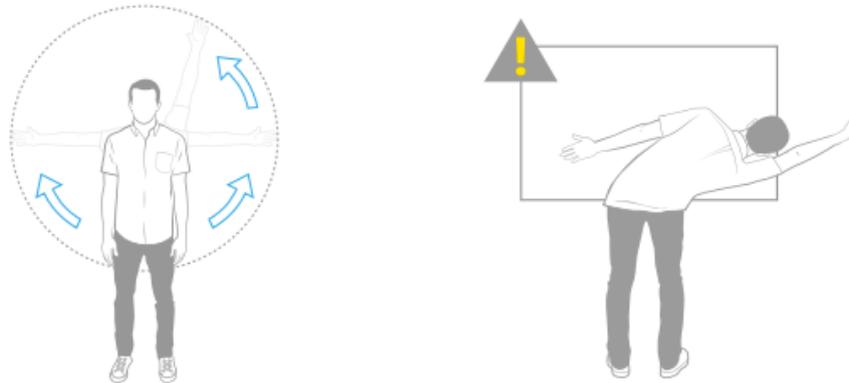


Figura 27 – Postura do utilizador e limites físicos<sup>54</sup>.

### 2.6.5.1 Interfaces para interação com gestos

As interfaces para a interação através dos gestos, devem ser desenvolvidas tendo em conta este contexto específico e as suas especificidades, pois comparativamente a um interface multitoque, as características são muito distintas.

Neste tipo de aplicações, o sistema está sempre ligado, ou seja, está sempre ativo, e é necessário distinguir quando é intenção interagir com o sistema de quando existe uma interação acidental. Determinar quando o utilizador pretende interagir é um desafio deste tipo de aplicações. Nestas situações, para além do reconhecimento dos gestos, o som também pode ser incluído, para evitar utilizações indesejadas.

O design de interfaces destas aplicações deve ter em conta o tipo de interação. Assim, a interface deve ser desenhada especificamente para a interação gestual, e não deve ser uma adaptação de interfaces desenhadas para outros dispositivos ou tipos de interação.

<sup>53</sup> Figura retirada do documento “*Kinect for Windows - Human Interface Guidelines*” (Microsoft, 2012).

<sup>54</sup> Imagem retirada do documento “*Kinect for Windows - Human Interface Guidelines*” (Microsoft, 2012).

*“The experiences and expectations users bring to the system impact their interactions.”(Microsoft, 2012).*

Cada utilizador tem expectativas diferentes em relação à utilização da aplicação. Por exemplo, a execução de determinado gesto pode ser interpretada de forma diferente por diferentes utilizadores. A Figura 28 mostra um exemplo de duas interpretações diferentes do mesmo gesto. Esta dificuldade de descodificação do gesto deve ser ultrapassada dando ajudas visuais ao utilizador.

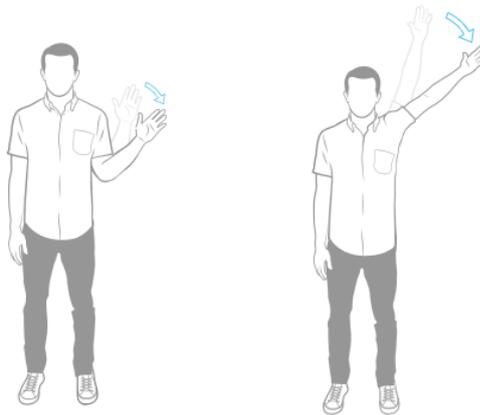


Figura 28 – Duas interpretações diferentes do gesto “Wave”<sup>55</sup>.

Sempre que é possível interagir com gestos devem ser dadas instruções nesse sentido ao utilizador. Estas indicações podem ser de vários tipos, imagens demonstrativas, animações demonstrativas. Se o gesto for do tipo estático, pode ser utilizada uma imagem, mas no caso de gestos dinâmicos é recomendada a animação para demonstrar a execução do gesto. A Figura 29 mostra um dois exemplos de indicações ao utilizador para a execução dos gestos.

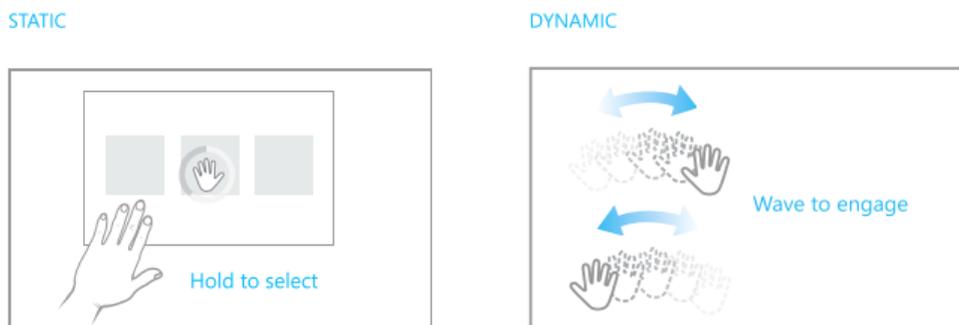


Figura 29 – Ajuda na execução dos gestos através de imagens ou animações<sup>56</sup>.

<sup>55</sup> Imagem retirada do documento “Kinect for Windows - Human Interface Guidelines” (Microsoft, 2012).

O *feedback* ao utilizador é fundamental, permite ao utilizador perceber o que está a acontecer na aplicação, evitando a sensação de estar perdido ou sem saber o que fazer.

No caso de haver problemas de deteção do utilizador, ou problemas com o sensor, é importante informar o utilizador que o sistema não está a funcionar devidamente, mas também dar sugestões de correção do problema identificado.

Os elementos da interface que sejam seleccionáveis devem ter pistas visuais que indiquem ao utilizador a possibilidade de seleção ou de manipulação. A utilização de cores, variação de tamanho, mudança de conteúdo no *hover*<sup>57</sup> dos conteúdos é recomendada. Assim o utilizador tem indicações dos elementos que podem ser seleccionados.

Em situações em que é necessário estar numa determinada posição durante um dado intervalo de tempo, deve ser dado *feedback* visual do progresso do tempo, através da utilização de elementos gráficos animados como barra de progresso. Deve evitar-se colocar o utilizador demasiado tempo numa posição, pois evitar cansaço. Na deteção do utilizador, é necessário dar *feedback* ao utilizador do que o sensor consegue “ver”, para ser mais ágil o processo de deteção; esse *feedback* pode ser obtido através da visualização da informação vídeo obtida do sensor ou através de outras indicações visuais.

## 2.7 *User experience*

*“UX represents the intersection of where technology meets the human being, designing for people and not solely producing for consumption”* (Kelway, n.d.).

A evolução da tecnologia nas últimas três décadas modificou a forma das pessoas interagirem com as aplicações multimédia, o que levou à criação de uma disciplina, a *User experience* (UX). A *user experience* estuda e investiga a forma como o utilizador se sente em relação à utilização de um sistema, inclui os aspetos mais emocionais e, conseqüentemente, mais subjetivos da relação da pessoa com o sistema, como por exemplo a emoção. O sistema pode ser um website, uma aplicação *Web*, um *software* de *desktop*, ou uma aplicação para telemóvel. A *user experience* investiga a forma como o utilizador percebe a utilidade do sistema, a facilidade de utilização ou a sua eficiência (Kelway, n.d.).

---

<sup>56</sup> Figura retirada do documento “*Kinect for Windows - Human Interface Guidelines*” (Microsoft, 2012)

<sup>57</sup> **Hover** – Mudança de estado quando o elemento é intersetado pelo rato, no caso das interfaces controladas por gestos, o elemento gráfico representativo da mão do utilizador.

Don Norman, especialista da área da *user experience* introduz o conceito e justifica a sua necessidade pois, outros termos, como usabilidade e interface humana não eram considerados suficientes para descrever a experiência de um utilizador com um sistema, incluindo os gráficos, a interface e a interação física (“Peter in Conversation with Don Norman About UX & Innovation,” 2007).

James Kelway (Kelway, n.d.) refere que a experiência faz parte da condição humana através de uma mistura de processos cognitivos e da memória. Também a personalidade de cada indivíduo influencia a qualidade da experiência que as pessoas sentem quando utilizam produtos digitais.

Este autor considera existirem seis princípios a ter em conta no contexto da *user experience*, que devem ser levados em conta no desenvolvimento de interfaces que promovam boas experiências de utilização. Os seis princípios a ter em conta são: persuasão, comportamento, design visual, usabilidade, interação e conteúdo. Segundo o autor, estas áreas representam o fundamental da *user experience design* (Kelway, n.d.).

Vários autores de referência na área da interação Humano-Computador definiram/ descreveram o conceito de *user experience*:

*“All the aspects of how people use an interactive product: the way it feels in their hands, how well they understand how it works, how they feel about it while they’re using it, how well it serves their purposes, and how well it fits into the entire context in which they are using it.” (Alben, 1996).*

*“A consequence of a user’s internal state (predispositions, expectations, needs, motivation, mood, etc.), the characteristics of the designed system (e.g. complexity, purpose, usability, functionality, etc.) and the context (or the environment) within which the interaction occurs (e.g. organizational/social setting, meaningfulness of the activity, voluntariness of use, etc.) (Hassenzahl & Tractinsky, 2006).*

*“Every aspect of the user’s interaction with a product, service, or company that make up the user’s perceptions of the whole. User experience design as a discipline is concerned with all the elements that together make up that interface, including layout, visual design, text, brand, sound, and interaction. UE works to*

*coordinate these elements to allow for the best possible interaction by users.” (“Usability Body of Knowledge - Glossary,” n.d.).*

A *user experience* estuda a forma como o utilizador se sente em relação à utilização de uma aplicação, e é uma componente muito importante na área das interfaces controladas por gestos. A investigação na área das interfaces controladas por gestos é recente, e ainda mais recente o estudo da *user experience* neste tipo de aplicações. Verifica-se a necessidade de efetuar estudos que avaliem as questões mais emocionais do utilizador ao interagir com aplicações desta natureza.

Idealmente, as interfaces controladas por gestos devem permitir ao utilizador focar-se na tarefa que estão a desempenhar, pelo que o dispositivo de entrada não deve ser um objeto de distração (Beurden & IJsselsteijn, 2011).

Aquando de uma avaliação da *user experience* devem ser considerados aspetos pragmáticos, relacionados com a eficácia e eficiência do utilizador, mas é igualmente importante considerar os aspetos hedónicos, como a satisfação em geral com a aplicação, apreciação do aspeto gráfico, a diversão ou o conforto na utilização da aplicação. No estudo de natureza comparativa (M. Nielsen et al., 2004), entre os tradicionais dispositivos de entrada como rato e o teclado e as interfaces, interfaces controladas por um dispositivo de controlo de movimento (*Wii remote*) e as interfaces isentas de dispositivos de entrada (corpo do utilizador como elemento que controla a interação), demonstra-se que a interação onde existe um dispositivo apresenta melhores resultados em aspetos mais pragmáticos como o tempo de execução da tarefa ou a precisão. São também referidos alguns aspetos negativos da interação em interfaces controladas por gestos, nomeadamente problemas de natureza ergonómica, como o desconforto físico ou a fadiga.

No site *OpenNI Arena*<sup>58</sup>, uma área de submissão de aplicações de interação natural apresenta um manual de boas práticas (OpenNI, 2012) no desenvolvimento deste tipo de aplicações. Este manual tem um tópico dedicado às *user experience recommendations*. O documento apresenta um conjunto de recomendações ao nível da *user experience* para quem pretender submeter aplicações de interação natural nesta plataforma. Em seguida apresenta-se, de forma resumida, estas recomendações.

O primeiro conselho listado neste documento refere-se ao gesto indicado para sair da aplicação, juntamente com a tecla “ESC”. O gesto sugerido (Figura 30) é o cruzar de braços, com a justificação de que é uma posição/gesto na qual é difícil de ser despoletado de forma acidental,

---

<sup>58</sup> <http://arena.openni.org/> consultado a 12/9/2012

fácil de executar e de aprender. Refere, ainda, que nas interfaces controladas por gestos se deve evitar a utilização do rato e teclado para *input* de dados.



Figura 30 - Gestos sugerido para sair da aplicação<sup>59</sup>.

Outra recomendação é indicar a zona na qual os utilizadores devem estar por forma a serem reconhecidos pelo sensor.

Uma vez que o utilizador não utiliza nenhum dispositivo físico para o controlo da aplicação, e numa aplicação na qual interagem simultaneamente vários utilizadores no mesmo local, dar indicação de qual está de facto no controlo da aplicação. Pode ser também importante dar *feedback* visual quando o utilizador está a controlar a aplicação.

Outra recomendação é utilizar convenções baseadas em aspetos da vida real, como as coisas se comportam e integram no mundo real. Por exemplo, o acenar para chamar à atenção. As convenções do mundo real podem facilitar a aprendizagem dos utilizadores.

Uma outra sugestão é permitir aos utilizadores recuperarem de uma decisão errada, especialmente nas aplicações onde a precisão que se consegue obter com os dispositivos utilizados é limitada.

Procurar evitar o desconforto físico dos utilizadores é outra recomendação apresentada. Verifica-se a necessidade de aconselhar os utilizadores a descansar, sem que por isso experimentem qualquer tipo de penalização na aplicação.

A necessidade de considerar diferentes sequências de gestos, por forma a não haver uma confusão de instruções com os gestos, é um aspecto também contemplado. O manual recomenda planear previamente as diferentes sequencias de movimentos possíveis.

---

<sup>59</sup> Figura retirada do documento intitulado “NI Apps Guidelines” (2012).

## 2.8 Exemplos de projetos de aplicações controladas por gestos

Nesta secção apresentam-se alguns projetos relevantes para exemplificar aplicações da interação controlada através de gestos.

### 2.8.1 Augmented Mirror

A aplicação "Augmented Mirror" (Figura 31) permite o controlo em tempo-real de um Avatar que está a ser visualizado num ecrã gigante para o público.



Figura 31 – Projeto Augmented Mirror<sup>60</sup>.

Um sistema de captura de movimento foi desenvolvido e montado de forma a permitir que uma pessoa possa controlar o avatar. O avatar está inserido num ambiente de vídeo real e pode controlar objetos virtuais presentes no cenário.

A aplicação apresentada é composta essencialmente por dois cenários. O cenário de controlo, onde através dos sistemas de captura de movimento desenvolvidos se procede à captura dos movimentos do utilizador, e o segundo cenário onde a audiência pode controlar o avatar. A ligação entre ambos os cenários é efetuada através de comunicação em rede. O elemento principal da captura de movimento é o sensor *Microsoft Kinect* que, juntamente com a *OpenNI*<sup>61</sup>, efetua o reconhecimento e a deteção do movimento do utilizador, que é utilizado para animar o avatar. Através de um complexo sistema de captura de movimentos, que para além do sensor *Kinect* inclui também um giroscópio e um *Wii remote*, é possível simular o movimento dos lábios do utilizador no avatar. O *Wii remote* é utilizado para a manipulação dos objetos virtuais através do

<sup>60</sup> Imagem retirada do artigo intitulado "Augmented Mirror : Interactive Augmented Reality System Based on Kinect" (Vera et al., 2011)

<sup>61</sup> **OpenNI** - *Open Natural Interaction* é uma *framework* de desenvolvimento de aplicações controladas por gestos.

movimento dos braços do utilizador. Toda a informação recolhida é organizada e enviada para o cenário, tornando possível a comunicação em tempo real entre o avatar e a audiência. O vídeo real do cenário é captado através de outro *Kinect* colocado para a audiência, permitindo assim que a audiência esteja presente no vídeo do avatar (Wilson, 2008).

*“The proposed system is composed of two main parts, called scenarios. In the first one, the control scenario, the actor tracking is performed using a MoCap system. In the second one, the augmented scenario, the audience can interact with the avatar and other virtual objects. These two scenarios are connected using a Client- Server model, via an Ethernet connection, so that the avatar can be remotely controlled.”* (Vera et al., 2011).

### **2.8.2 Project-Saara**

O *Project-Saara* é um projeto de realidade aumentada<sup>62</sup> em que o utilizador pode navegar num ambiente virtual. A base da aplicação é uma enciclopédia cujo conteúdo por ser acedido através da navegação virtual. A Figura 32 mostra o menu principal da aplicação, enquanto a Figura 33 mostra a imagem do posicionamento do utilizador para reconhecimento pelo sistema.

*“This interface is visible through augmented reality glasses. It presents the information (sound, video, picture and text) in a 3D space, a real room. Intuitive gestures (arm movements) and speech recognition to interact with content are allowed through a microphone and a camera which detect and capture the movement (Kinect).”*<sup>63</sup>

---

<sup>62</sup> **Realidade aumentada** - Sobreposição de informações virtuais em visualizações do mundo real. A realidade aumentada caracteriza-se por disponibilizar informação, à qual o utilizador não teria acesso imediato com os seus próprios sentidos (Azuma, 1997).

<sup>63</sup> Retirado de <http://www.projet-saara.com/>, consultado a 1 de Novembro de 2011.



Figura 32 - Menu de utilitários da aplicação – disponível através de comandos por voz<sup>64</sup>.



Figura 33 - Indicação da posição para calibração e reconhecimento do utilizador<sup>65</sup>.

### 2.8.3 Interactive Puppet

Uma instalação multimédia interativa desenvolvida pela empresa yoke.dk<sup>66</sup>, permitia ao público controlar o personagem através da interação mediada pelo corpo. A instalação multimédia esteve exposta na Dinamarca. O principal objetivo desta aplicação era permitir à audiência controlar a personagem apresentada (uma das principais do espetáculo que estava em cena). No vídeo<sup>67</sup> de apresentação da instalação é possível ver que a aplicação está colocada num local público e pode ver-se as pessoas a interagirem com o personagem (“Developkinect -Interactive Puppet,” 2011).

<sup>64</sup> Imagem retirada de <http://www.projet-saara.com/projet.php>

<sup>65</sup> Figura retirada de <http://www.projet-saara.com/projet.php>

<sup>66</sup> Website da empresa <http://www.yoke.dk/>

<sup>67</sup> Vídeo de apresentação da instalação *Interactive Puppet* <http://vimeo.com/33030019>

A Figura 34 mostra um exemplo de interação com a aplicação num espaço público. A aplicação interativa foi desenvolvida com recurso aos *softwares* openFrameworks<sup>68</sup> e Unity3D juntamente com o add-on ofxOpenNI<sup>69</sup>. Um aspeto importante no âmbito desta instalação é que não depende da calibração inicial do utilizador, permitindo uma rápida inicialização do sistema e interação com o personagem (“Developkinect -Interactive Puppet,” 2011).



Figura 34 - Instalação multimédia interativa – *Interactive Puppet*<sup>70</sup>.

#### 2.8.4 Follow This!

A aplicação **Follow this** é um jogo que apresenta uma interface controlada por gestos e foi desenvolvida pelos artistas Nick Hardeman (conceitualização e programação) e por Lauren Licherdell (Design e 3D). O jogo é baseado no sítio Web da rede social Twitter<sup>71</sup>, onde cada pássaro representa uma citação retirada do site. O pássaro explode após ser atingido e forma a citação que representa. O controlo das armas é efetuado através dos movimentos dos braços do utilizador. Cada braço representa uma arma e permite disparar na direção dos pássaros através do controlo gestual. O evento associado à ação disparar é despoletado pelo gesto de abrir as mãos. Como é possível observar no vídeo<sup>72</sup> de apresentação, o jogo apresenta um painel de calibração que permite ajustar as definições de deteção das mãos do jogador.

---

<sup>68</sup> **openFrameworks** é um *software* programado em C++ desenvolvido para o desenvolvimento de aplicações multimédia interativas criativas.

<sup>69</sup> **ofxOpenNI** – Módulo que permite interligar a *framework* OpenNI e NITE e o sensor Kinect no *openFrameworks*.

<sup>70</sup> Imagem retirada de <http://www.behance.net/gallery/interactive-puppet/2577467> 5/11/2012

<sup>71</sup> Site de rede social que permite aos utilizadores atualizarem o seu estado e consultarem o estado dos amigos. O conceito deste site de rede social é o facto limitar o número de caracteres a 140 caracteres. <http://twitter.com/>

<sup>72</sup> Vídeo de apresentação do jogo <http://vimeo.com/29032379>

Esta aplicação foi desenvolvida com o recurso ao sensor *Microsoft Kinect*. A Figura 35 mostra um exemplo de um utilizador a interagir com a aplicação.

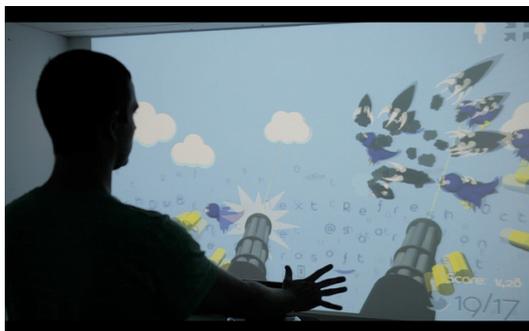


Figura 35 - Jogo interativo controlado por gestos *Follow This*<sup>73</sup>.

*“Follow This! Is a first person shooter game utilizing the Microsoft Kinect camera. Input a twitter search term to create birds, varying based on the loaded tweets.”* (Hardeman & Licherdell, 2011).

### 2.8.5 *Trailerpark interactive visuals*

O projeto *Trailerpark Interactive Visuals* (Figura 36) foi desenvolvido num âmbito artístico e permite aos utilizadores envolverem-se num ambiente colorido e também imersivo. O utilizador controla os elementos virtuais através dos gestos por forma a criar novos efeitos visuais.



Figura 36 - *Trailerpark Interactive Visuals*<sup>74</sup>.

Foi um projeto implementado pela empresa *Yoke.dk*<sup>75</sup> e, de acordo com o vídeo de apresentação<sup>76</sup>, bastante bem recebido pelo público. As tecnologias associadas à implementação técnica são *OpenFrameworks*, *OpenNI*, *Unity3D*.

<sup>73</sup> Retirado de <http://nickhardeman.com/460/follow-this/>

<sup>74</sup> Imagem retirada do vídeo de apresentação da instalação multimídia interativa <http://vimeo.com/28452629>

*“This is a preview of our test setup. We use a kinect to let the audience control aspects of the visuals – for instance the lighting of the scene. The set is modelled in 3D and mapped accurately onto the physical model following a calibration process. We use Openframeworks and OpenNI for the kinect tracking, Unity3D for the interactive 3D, and VDMX and Syphon for mixing Unity with prerendered content.”* (“Kinect interactive visuals,” 2011).

### **2.8.6 Projeto SocialStage**

Este projeto foi desenvolvido pela empresa Snibbeinteractive<sup>77</sup> sendo baseado no filme Avatar<sup>78</sup>, do realizador James Cameron. A aplicação<sup>79</sup> permite ao utilizador "entrar" num ambiente virtual 3D. Através de um sistema de captura de movimento a instalação permite ao utilizador controlar um avatar num ambiente virtual em tempo real. A Figura 37 mostra um exemplo de utilização do sistema.

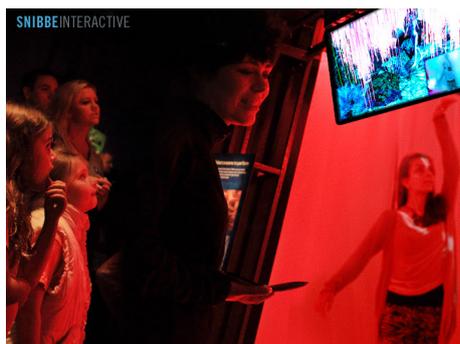


Figura 37 – Projeto SocialStage<sup>80</sup> (Hardeman & Licherdell, 2011).

*“Using the same performance-capture technologies from blockbuster films, Snibbe Interactive's SocialStage translates visitors real-time motions into your own on-screen animated characters and worlds. Light years beyond similar technologies, SocialStage uses a dozen cameras to capture every subtlety of*

<sup>75</sup> Website da empresa <http://www.yoke.dk/>

<sup>76</sup> Vídeo de apresentação da instalação <http://vimeo.com/28452629>

<sup>77</sup> Website da empresa Snibbeinteractive <http://www.snibbeinteractive.com/>

<sup>78</sup> Para mais informações relativas ao filme Avatar <http://www.avatarmovie.com/>

<sup>79</sup> [http://www.youtube.com/watch?v=6LC3bO9D8w0&feature=player\\_embedded](http://www.youtube.com/watch?v=6LC3bO9D8w0&feature=player_embedded)

<sup>80</sup> Imagem retirada de [http://www.snibbeinteractive.com/product\\_images/jpg/SocialStage\\_001.jpg](http://www.snibbeinteractive.com/product_images/jpg/SocialStage_001.jpg)

*visitors' personalities. Anyone can bring a motion capture character to life in real time, regardless of shape or stature to enjoy an experience that makes them part of the action.*" ("Kinect interactive visuals," 2011).

### 2.8.7 Wi-Go

O Wi-Go (Figura 38) é um projeto desenvolvido na Universidade da Beira Interior (UBI) e é um carro de compras automatizado que identifica e segue o utilizador; visa promover e facilitar a mobilidade de pessoas portadoras de deficiência física. A ideia surgiu da dificuldade com que os utilizadores com mobilidade reduzida ou em cadeira de rodas se deparam quando fazem compras num supermercado. Este projeto académico já evoluiu para a criação de uma empresa, a IS2you<sup>81</sup>.



Figura 38 - Wi-Go<sup>82</sup>.

*“Wi-GO é um dispositivo autónomo que está a ser desenvolvido para seguir uma determinada pessoa com deficiência ou mobilidade reduzida, para onde quer que esta se desloque, permitindo assim transportar objetos de forma autónoma, dinâmica e segura, evitando obstáculos e perigos”* (“Wi-GO,” 2012).

O Wi-GO utiliza o sensor *Kinect* para o reconhecimento do utilizador e do meio envolvente, evitando que o carro vá contra objetos ou pessoas. Este carro pode ser utilizado em vários contextos, como por exemplo aeroportos, centros comerciais, hospitais, lojas ou sempre que seja

---

<sup>81</sup> <http://www.is2you.eu/PT/index.html>, consultado a 19 de Novembro de 2012

<sup>82</sup> Imagem retirada de <http://www.wi-go.is2you.eu/>

necessário o transporte de objetos. Esta solução inovadora pode ajudar a melhorar a qualidade de vida de pessoas com mobilidade reduzida.

## **2.9 Mundos Virtuais 3D**

Um mundo virtual é um ambiente desenvolvido através de técnicas computacionais, onde os utilizadores podem interagir uns com os outros através de avatares. Um avatar é uma representação que identifica um utilizador. Essa representação pode ser textual, a duas dimensões ou a três dimensões. Uma das características principais dos mundos virtuais é o facto de permitirem que vários utilizadores interajam em tempo real.

Richard Bartle, o criador do mundo virtual com base em texto entre 1970 e 1980 descreve o mundo virtual como *“a world is an environment that its inhabitants regard as being self-contained. It doesn't have to mean an entire planet: It's used in the same sense as "the Roman world" or "the world of high finance" (Bartle, 2004).*

Raph Koster investigou e desenvolveu na área dos mundos virtuais, nos anos 90, e refere que *“a virtual world is a spatially based depiction of a persistent virtual environment, which can be experienced by numerous participants at once, who are represented within the space by avatars” (Koster, 2004).*

Mais tarde, Edward Castronova, define mundos virtuais como *“crafted places inside computers that are designed to accommodate large numbers of people” (Castronova, 2004).* Castronova não refere na sua definição a ideia de comunicação em tempo real. Como refere Mark Bell, a definição apresentada por Castronova poderia ser aplicada a uma sala de conversações online ou à simples partilha de um documento online. Este autor apresenta, em alternativa, uma proposta de definição para o conceito de mundo virtual: *“a synchronous, persistent network of people, represented as avatars, facilitated by networked computers.” (Bell, 2008).*

A definição de *virtual worlds* não é consensual entre os investigadores e criadores de mundos virtuais. No âmbito de mundos virtuais que não são jogos é também utilizada a designação *Social virtual worlds*.

*“Virtual worlds that are not games are considered to be social virtual worlds. Second Life is the largest social virtual world at the moment with over fifteen million registered users.” (Sanchez, 2009).*

Um mundo virtual é acedido através do computador do utilizador que pode manipular os elementos presentes no mundo virtual. Os mundos virtuais podem ter características tipicamente associadas

ao mundo real, como a gravidade, as ações em tempo real, a movimentação dos utilizadores e a comunicação entre estes, ou podem ter características mais próximas de um mundo de fantasia, com ações que não existem no mundo real.

A comunicação dentro dos mundos virtuais pode ocorrer de várias formas, através de texto escrito, sons, elementos gráficos, gestos ou reconhecimento de voz.

A origem principal dos mundos virtuais está relacionada com o entretenimento, nomeadamente com os jogos MMORPG ("Massively Multiplayer Online Role Playing Game"), mas existem outras áreas de interesse dos mundos virtuais como é o caso da socialização e construção de comunidades online, propósitos educacionais ou de treino de comportamentos. Alguns dos exemplos de MMORPG são jogos como o *Dungeons & Dragons*<sup>83</sup> ou o *World of Warcraft*<sup>84</sup>.

### **2.9.1 Evolução dos mundos virtuais 3D**

O primeiro mundo virtual baseado na Internet, o MUD (*Multi-User Dungeon*), foi lançado em 1978 (Figura 39), apresentava uma interface textual e permitia aos vários jogadores interagirem em tempo real. Mais tarde o nome deste mundo virtual/jogo mudou para *Multi-User Dimension*.

*"Players navigated through the MUDs with text commands such as "/walk north" or "/open" to open a door. In text-based MUDs, directional cues such as north, west, south, and east were used to navigate through the virtual world. The directional cues were so precise that a user could construct a map on paper to get a physical sense of the world. Players of the early MUDs connected with TelNet and would log in and participate in adventures or quests like killing dragons or defeating evil wizards. "* (Sanchez, 2009).

---

<sup>83</sup> Website <http://www.ddo.com/>

<sup>84</sup> Website <http://us.battle.net/wow/pt/>



Figura 39 - Interface do *Starcraft*, um MUD<sup>85</sup>.

As primeiras versões eram constituídas por muito poucas representações gráficas e com um frequente recurso a uma interface baseada em linha de comandos. Os utilizadores comunicavam através de mensagens escritas e viam uma descrição escrita do ambiente virtual.

Mais tarde, em 1989, James Aspnes desenvolveu o TinyMUD de cuja interface se pode ver um exemplo na Figura 40. Esta nova aplicação continha uma componente social mais forte dado que os jogadores podiam criar elementos do jogo. Estes objetos virtuais criados apenas podiam ser visualizados pelos outros jogadores, não podendo ocorrer interações. Mais tarde surgiu a versão MOOs (*Multi-User Dungeons Object Oriented*) que, para além de permitir criar conteúdos, permitia a partilha desses conteúdos com os outros jogadores.

<sup>85</sup> Imagem retirada do artigo "A Social History of Virtual Worlds" (Sanchez, 2009)



Figura 40 - Exemplo do TinyMUD desenvolvido para o Apple II<sup>86</sup>.

A fase seguinte da história dos mundos virtuais surgiu com os MMORPGs (*Massively multiplayer online role-playing game*) e influenciou muito o desenvolvimento dos mundos virtuais 3D sociais, através da criação de sistemas de colaboração e comunicação online. O Second Life<sup>87</sup> é um exemplo de um mundo virtual 3D social (Sanchez, 2009).

### 2.9.2 Principais características dos mundos virtuais

Existem diferentes tipos de mundos virtuais 3D. Betsy Book (Book, 2004) apresenta um conjunto de características no qual se baseia para caracterizar os mundos virtuais 3D:

- Ser um espaço partilhado, ou seja a possibilidade de vários utilizadores partilharem o mesmo espaço virtual simultaneamente;
- A interface gráfica retrata o espaço virtual onde estão os objetos virtuais 3D;
- Imediatismo, ou seja a interação ocorre em tempo real com vários utilizadores conectados ao mesmo tempo num mesmo local virtual;
- A interatividade, pois os ambientes virtuais permitem modificar, desenvolver, construir elementos do cenário e enviar conteúdos personalizados;
- A característica da persistência indica que o mundo virtual existe independentemente de haver, ou não, utilizadores conectados. A socialização é uma característica importante no

<sup>86</sup> Imagem retirada do artigo "A Social History of Virtual Worlds" (Sanchez, 2009)

<sup>87</sup> Website do Second Life <http://secondlife.com/>

âmbito dos mundos virtuais, permitindo a formação de grupos e a interação entre avatares (Book, 2004).

### 2.9.3 Desenvolvimento de ambientes virtuais 3D

Pretende-se nesta secção apresentar os principais ferramentas de desenvolvimento/plataformas de ambientes virtuais 3D.

#### 2.9.3.1 Second Life

O *Second Life*<sup>88</sup> (Figura 41) é um ambiente virtual 3D desenvolvido pela empresa *Linden Labs*<sup>89</sup> e foi lançado em 2003. Após o registo, quando o utilizador acede ao mundo virtual é-lhe atribuído por defeito um avatar que pode, depois, ser personalizado. É então possível explorar o mundo virtual, interagir com outros utilizadores e criar objetos virtuais 3D. A interação pode ser feita através do sistema de *chat* disponibilizado pelo sistema ou mesmo através de comunicação por voz. O *Second Life* tem uma linguagem própria de programação a LSL (*Linden Scripting Language*).



Figura 41 – *Second Life* (Book, 2004).

<sup>88</sup> Website do *Second Life* <http://secondlife.com/>

<sup>89</sup> Website da empresa *Linden Lab* <http://lindenlab.com/>

Este ambiente virtual tem um sistema monetário próprio, o *Linden Dollar*, que permite negociar vários elementos do ambiente (acessórios, casas, entre outros), embora exista um sistema de conversão moeda de *Linden Dollar* para dinheiro real.

### 2.9.3.2 OpenSimulator

O *openSimulator*<sup>90</sup> é um servidor de mundos virtuais que pode ser utilizado para criar ambientes virtuais em 3D. Permite recriar um ambiente similar ao *Second Life*. Apresenta uma característica interessante, que é o facto de poder correr em modo *offline* ou *online*. O *OpenSimulator* foi desenvolvido na linguagem de programação C# e está disponível para os sistemas operativos Windows, Linux e Mac OSX. A Figura 42 apresenta um exemplo da interface deste mundo virtual.



Figura 42 - openSimulator<sup>91</sup>.

Algumas das principais características deste simulador são o facto de suportar ambientes multiutilizador *online* e assegurar a comunicação entre ambientes 3D de diferentes dimensões. O *OpenSimulator* é compatível com vários clientes e protocolos e permite a criação em tempo real de conteúdos 3D.

### 2.9.3.3 Blender 3D

O blender3D<sup>92</sup> (Figura 43) é um *software* desenvolvido pela empresa *Blender Foundation* e permite a modelação, a animação, a renderização e a edição de vídeo para aplicações multimédia

<sup>90</sup> Website do OpenSimulator [http://opensimulator.org/wiki/Main\\_Page](http://opensimulator.org/wiki/Main_Page)

<sup>91</sup> Retirado de [http://opensimulator.org/wiki/File:Osgrid\\_wright\\_plaza\\_shops\\_kg.png](http://opensimulator.org/wiki/File:Osgrid_wright_plaza_shops_kg.png)

<sup>92</sup> Website do *software* <http://www.blender.org/>

interativas 3D. O blender3D tem um motor de jogos, *blender game engine*, que permite a criação de jogos e está disponível para vários sistemas operativos, como o Windows, Linux e Mac OS X.



Figura 43 – Exemplo de modelo desenvolvido no Blender3D<sup>93</sup> / Logótipo do Blender3D.

#### 2.9.3.4 Unity3D

O *Unity3D*<sup>94</sup> é um motor de jogo, ou seja, um sistema desenhado para a criação e desenvolvimento de videojogos, espaços de interação e ambientes 3D. A utilização do *software Unity3D* vai para além da mera construção de jogos, sendo também utilizado para o desenvolvimento de outro tipo de interfaces 3D, como a criação de projetos de natureza arquitetónica. Um motor de jogo é diferente de um *software* de criação 3D dado que permite a junção de vários elementos 3D num cenário tipicamente de jogo, embora estes possam ser modelados e criados noutra *software* 3D, como por exemplo o Maya, Blender ou 3D studio MAX. A Figura 44 mostra o logótipo do *software Unity3D*.



Figura 44 – *Unity3D* Logótipo<sup>95</sup>.

---

<sup>93</sup> Retirado de <http://www.blender.org/features-gallery/gallery/art-gallery/>

<sup>94</sup> Website do Unity3D <http://www.unity3d.com/>

<sup>95</sup> Retirado de <http://unity3d.com/unity/>

O *Unity3D* é um motor de jogo 3D similar ao Blender 3D<sup>96</sup> ou ao Torque Game Engine<sup>97</sup>. Pode ser programado em BOO, C# ou *JavaScript*. Apresenta suporte de *browser*, através do plug-in *Unity web Player* e pode ser acedido através dos principais *browsers*, como *Firefox*, *Internet Explorer*, *Safari* e *Chrome*. O *Unity3D* está disponível para os sistemas operativos Windows e Mac OS X e tem suporte para Xbox 360 e PlayStation 3.

*“For many users, especially enterprise users, this large download and install represents a significant obstacle to virtual world acceptance. The authors describe a technical implementation that uses the Unity 3D browser plug-in as a way to access a virtual world from within a Web browser.”* (Katz et al., 2011).

Algumas das principais características do *Unity3D* são o facto de incluir nas suas funcionalidades a possibilidade de *streaming* de vídeo e de áudio através do *codec Ogg Vorbis*. O *Unity3D* apresenta ainda várias funcionalidades relacionadas com a comunicação em rede (sincronização em tempo real do ambiente virtual) em ambiente multiutilizador. Os protocolos de comunicação permitidos são TCP/IP e UDP. A Figura 45 mostra alguns exemplos de aplicações desenvolvidas com recurso ao *software Unity3D*. O *Unity3D* tem licenciamento grátis para fins educacionais.



Figura 45 – Exemplos de aplicações desenvolvidas com o *Unity3D*<sup>98</sup>.

---

<sup>96</sup> <http://www.blender.org/>

<sup>97</sup> <http://www.garagegames.com/>

<sup>98</sup> Retirado de <http://unity3d.com/gallery/made-with-unity/game-list>



### 3 Metodologia

A presente investigação teve em conta o processo de investigação em ciências sociais proposto por Raymond Quivy e LucVan Campenhoudt. A definição da problemática foi a primeira etapa no processo de investigação, na qual se definiram os contornos da área e da temática da investigação a desenvolver. Seguidamente foi formulada a pergunta de partida, tendo em conta os critérios para construção de uma boa pergunta de partida. Seguiu-se a fase de exploração através das leituras e revisão da literatura relativa aos principais conceitos envolvidos na problemática e na pergunta de partida. A construção do modelo de análise foi o passo seguinte: o modelo de análise apresenta os conceitos, dimensões e indicadores relevantes nesta investigação e que definem quais os aspetos do problema que, neste trabalho, se considera caracterizarem os fenómenos em estudo. Após este trabalho preparatório foi, então, concetualizado e implementado um protótipo funcional que permitiu testar alguns dos conceitos estudados na revisão de literatura. A última etapa deste processo de investigação foi a formulação das conclusões onde se sistematizam as principais ilações do trabalho realizado bem como os contributos da investigação (conhecimentos de âmbito teórico e/ou prático) para a área investigada (Quivy & Campenhoudt, 1995).

A natureza desta investigação é **exploratória** pois pretende-se perceber um fenómeno relativamente recente e do qual não se conhece o suficiente (Carmo & Ferreira, 1998). Ao nível da recolha de dados, esta investigação baseia-se na experiência de desenvolvimento de um protótipo tendo a atenção sido centrada nas áreas da Interação humano-computador, interação gestual, usabilidade e *user experience* em ambientes 3D. De referir ainda que o diálogo com especialistas da área foi também importante no âmbito desta investigação.

Nesta investigação o desenvolvimento de um protótipo funcional teve um papel central, tendo tido como base a investigação teórica realizada, tendo sido concebido e implementado um ambiente de interação com base no paradigma de interação natural num ambiente 3D multiutilizador. O desenvolvimento do protótipo foi alvo de ajustes sucessivos que permitissem alinhar as funcionalidades desenvolvidas com os objetivos inicialmente definidos.

O procedimento metodológico adotado segue, ainda, uma abordagem de tipo investigação de desenvolvimento ou investigação tecnológica através de uma abordagem qualitativa (R. C. Richey & Klein, 2005). Este tipo de procedimento metodológico permite identificar um problema específico, desenhar e identificar possíveis soluções, elaborar estratégias de desenvolvimento e proceder à construção de um protótipo, que é um dispositivo experimental que permite testar e validar as soluções tecnológicas adotadas. Segundo Richey e Nelson, esta abordagem metodológica apresenta duas tipologias principais, em função do objeto ou resultado da

investigação. A primeira tipologia centra-se no estudo de produtos específicos ou desenvolvimento de aplicações, cujo resultado são as lições aprendidas com esses desenvolvimentos específicos e a análise das condições que facilitam o seu uso. A segunda tipologia investiga os processos de design, de desenvolvimento ou avaliação, ferramentas ou modelos, cujo resultado são novos procedimentos de design, de desenvolvimento e de avaliação e/ou modelos e condições que facilitam o seu uso. (Oliveira, 2006). Este trabalho tem uma natureza compatível com a primeira destas tipologias, dado que teve como objetivo conceber um protótipo que permitisse testar as tecnologias, inicialmente identificadas no estado de arte, que possibilitam construir um ambiente 3D colaborativo com interação através dos gestos.

A investigação de desenvolvimento, segundo Van Der Maren (apud Oliveira, 2006), subdivide-se em três tipos: desenvolvimento do conceito, desenvolvimento do objeto e desenvolvimento de habilidades pessoais. Esta investigação enquadra-se no segundo tipo, o desenvolvimento do objeto, pois centra-se na investigação das tecnologias que permitem a construção de um protótipo de interação gestual em ambiente 3D em rede.

### **3.1 Estrutura da investigação**

Esta investigação está organizada em cinco etapas principais, de acordo com a metodologia adotada. A primeira etapa engloba a definição da problemática, a identificação dos objetivos da investigação e a formulação da questão de investigação. A segunda etapa diz respeito à pesquisa e investigação dos conceitos teóricos inerentes à problemática, incluindo as tecnologias que permitem a criação de interação através dos gestos num ambiente virtual 3D, as condições de utilização de uma aplicação desta natureza e as principais orientações numa perspetiva da usabilidade e *user experience* de uma aplicação deste tipo. A terceira etapa compreende a conceptualização do protótipo, a especificação dos requisitos funcionais e das opções tecnológicas para a implementação do protótipo. A quarta etapa corresponde ao processo de desenvolvimento e construção do protótipo, o objeto de estudo. A quinta, e última, etapa é uma reflexão sobre o processo de construção do protótipo, as opções de desenvolvimento tomadas, as tecnologias utilizadas, os problemas encontrados, as soluções adotadas.

### **3.2 Protótipo**

A conceptualização e desenvolvimento do protótipo é uma componente fundamental deste trabalho pois deu suporte à componente empírica da investigação. A conceção e construção deste componente permitiram testar conceitos, experimentar tecnologias e obter ilações práticas sobre este tipo de sistemas.

O protótipo desenvolvido no contexto desta investigação é um protótipo de alta-fidelidade, pois foi desenvolvido nas tecnologias finais de implementação. Esta tipologia de protótipo permite testar o máximo de detalhes possíveis relacionados com as tecnologias (*hardware* e *software*) envolvidas na construção do protótipo. Não se pretende que este tipo de protótipo esteja desenvolvido a um nível industrial, mas que possa ser validado através do *feedback* ao nível do conceito, do comportamento em testes em contexto de laboratório nomeadamente no que respeita ao funcionamento em geral e ao comportamento ao nível dos gestos (Saffer, 2009).

No contexto das interfaces gestuais, especificamente nos protótipos de alta fidelidade, Saffer considera que existem três tipos de protótipos: *Exact Prototype*, *Off-The-Shelf* e *Do-It-Yourself*. O tipo de protótipo desenvolvido no âmbito desta investigação insere-se no tipo *Exact Prototype* pois é desenvolvido com recurso a tecnologias existentes (*drivers*, sensores, *framework*, *software*) e para o ambiente final real, ou seja, não é um ambiente simulado (Saffer, 2009).

Ao desenvolver-se um protótipo com o recurso a tecnologias existentes é necessário consultar todo o material de suporte, do *hardware* ou do *software*, como por exemplo manuais, *drivers*, APIs. Esta consulta permite perceber melhor o funcionamento das tecnologias, o contexto de utilização e as diversas especificidades inerentes a cada uma.

Desenvolver um protótipo deste género, de alta-fidelidade, implica tempo e dedicação para se aprender as tecnologias, pois os ambientes de desenvolvimento são muito específicos e frequentemente em constantes modificações e atualizações por parte das empresas fornecedoras. É, assim, um desafio desenvolver este tipo de aplicações quando as tecnologias de desenvolvimento associadas não são estáveis.

### **3.3 Modelo de análise**

O modelo de análise é um marco importante no planeamento e desenvolvimento da investigação, pois permite clarificar e sistematizar os aspetos relevantes na caracterização do sistema em estudo. A construção dos conceitos, que consiste na definição das dimensões associadas ao conceito e na indicação dos diferentes indicadores que irão permitir medir as dimensões (Quivy & Campenhoudt, 1995).

O modelo de análise foi construído a partir da questão de investigação que orientou esta investigação: qual o potencial e limitações das atuais tecnologias de suporte à interação controlada por gestos na promoção de novas formas de interação em ambientes 3D distribuídos?

Qual o potencial e limitações das atuais <b>tecnologias de suporte à interação controlada por gestos</b> na promoção de novas formas de <b>interação em ambientes 3D</b> distribuídos?			
Conceitos	Dimensões	Componentes	Indicadores
Interação (Humano-Computador)	Tecnologia	Hardware	<ul style="list-style-type: none"> <li>- Sensor de movimento</li> <li>- Tipo de Conectividade</li> <li>- Tipo de dispositivo de saída</li> <li>- Posicionamento do sensor</li> </ul>
		Software	<ul style="list-style-type: none"> <li>- Sistema Operativo</li> <li>- Browser</li> <li>- Plug-ins</li> <li>- Frameworks</li> <li>- Drivers</li> </ul>
	Interação	Experiência do utilizador (UX)	<ul style="list-style-type: none"> <li>- Simplicidade</li> <li>- Visibilidade</li> <li>- Consistência</li> <li>- Estrutura</li> <li>- Feedback</li> <li>- Visualmente agradável</li> <li>- Recompensador</li> </ul>
		Usabilidade	<ul style="list-style-type: none"> <li>- Satisfação</li> <li>- Eficácia</li> <li>- Eficiência</li> <li>- Navegação</li> <li>- Imersão</li> <li>- Conforto</li> </ul>
	Gestos	Atributos	<ul style="list-style-type: none"> <li>- Duração (tempo)</li> <li>- Posição (x, y, z)</li> <li>- Movimento (Cima/baixo/Esquerda/Direita)</li> </ul>
		Patterns (Padrões)	<ul style="list-style-type: none"> <li>- Tipo (Ativação por proximidade, Movimento corporal, Apontar, Wave)</li> <li>- Significado (descrição)</li> <li>- Adequação (intuitivo, complexo, natural)</li> </ul>
Ambientes 3D	Características	Cenário	<ul style="list-style-type: none"> <li>- Organização</li> <li>- Dimensão/Tamanho,</li> <li>- Dinâmico/estático</li> <li>- Adequação ao contexto de utilização</li> </ul>
	Tecnologia	Software	<ul style="list-style-type: none"> <li>- Integração de sensores de movimento</li> <li>- Interfaces gestuais</li> <li>- Linguagem de programação</li> <li>- Plataforma de divulgação (Windows, Mac, Linux)</li> </ul>

Tabela 1 – Modelo de análise

A tabela 1 representa o modelo de análise orientador desta investigação. É constituído pelos conceitos principais inerentes a este estudo, as dimensões e componentes associados e os indicadores relevantes.

O conceito interação humano-computador está dividido em três dimensões: a tecnologia, a interação e os gestos. A dimensão tecnológica está subdividida nos componentes do *hardware* e *software*. Os indicadores associados identificam um conjunto de características e necessidades associadas ao protótipo e dão relevo à influência das opções de *hardware* e *software* na construção e performance do protótipo. A dimensão da interação é constituída pelos componentes experiência do utilizador e usabilidade. Inclui indicadores identificados na revisão bibliográfica e com implicações no desenvolvimento do protótipo. A dimensão gestos é constituída pela componente atributos, onde se encontram indicadores relativos aos gestos efetuados para o controlo da aplicação, como a duração do gesto e orientação no ambiente 3D e pela componente padrões, que está relacionada com a definição de gestos e as suas características (como o tipo de gesto, o seu significado e a adequação).

O conceito de ambiente 3D inclui as dimensões características e tecnologia. Essencialmente a caracterização envolve o componente cenário e tem indicadores relacionados com o aspeto e organização do cenário do ambiente virtual 3D. A dimensão tecnologia inclui a componente *software* com a finalidade de identificar a respetiva adequação: implementação de interfaces e interação através dos gestos, integração de sensores de movimento, sistemas operativos em que está disponível, que linguagens de programação suporta.



## 4 Implementação do protótipo

Neste capítulo descreve-se o desenvolvimento do protótipo o qual tem como principal objetivo a criação de um ambiente colaborativo 3D, onde cada utilizador com recurso às tecnologias de captura de movimento pode controlar um avatar através de interação gestual. Pretende-se também que seja possível uma interação avatar-avatar no ambiente 3D, em que estes possam interagir entre si, com a possibilidade de realizar atividades em grupo.

O desenvolvimento do protótipo apresentou especificidades, não só ao nível do *hardware* e *software* utilizados, mas também quanto à sua implementação. Neste capítulo apresentam-se todas as etapas desse desenvolvimento, desde a discussão da ideia, às tecnologias utilizadas e à própria implementação do protótipo.

### 4.1 Especificação dos requisitos

Neste tópico são descritos os requisitos funcionais mais relevantes para o protótipo no âmbito desta investigação, os quais estão divididos em três componentes que a seguir se apresentam:

#### **Componente de comunicação em rede**

Nesta componente o protótipo deve permitir:

- A interação simultânea entre diferentes utilizadores;
- Um funcionamento numa arquitetura cliente-servidor, assumindo um dos utilizadores o papel de servidor;
- O estabelecimento de comunicações síncronas de voz.

#### **Componente de Interação**

Neste âmbito estão compreendidos os requisitos funcionais relacionados com a interação e a interface do utilizador. Esta área integra também fatores essenciais previamente identificados na literatura, associados à usabilidade e à experiência do utilizador. Assim o protótipo deve permitir:

- A deteção e identificação individualizada do corpo do utilizador;
- O controlo do avatar de cada utilizador através da interação corporal deste último;
- A interação (seleção e manipulação) do avatar com os elementos presentes nos cenários do ambiente virtual e a navegação/movimentação nesse ambiente;
- A atribuição de gestos a eventos (controlo da interface através de comandos gestuais);

- O respeito pelos princípios de interação em ambientes virtuais 3D com interfaces controladas por gestos.

### Componente de modelação

Aqui estão compreendidos os requisitos relativos à construção do cenário 3D e dos avatares. Neste âmbito o protótipo deve permitir:

- Um avatar por defeito para todos os utilizadores;
- A personalização do avatar (incluindo mudanças por exemplo: da cor de cabelo, das roupas, etc.).

## 4.2 Escolha das Tecnologias

Após a especificação dos requisitos funcionais do protótipo, foi necessário proceder à escolha das tecnologias, *hardware* e *software*, necessários à sua implementação. A seleção das tecnologias teve em conta os objetivos subjacentes ao projeto e os requisitos funcionais atrás referidos. Para tal foram investigados vários projetos similares, e realizou-se uma investigação de natureza mais teórica (em livros, artigos científicos, conferências). Estabeleceram-se ainda contactos com outros investigadores na área.

Relativamente ao sensor de movimento utilizou-se o sensor de movimento **Kinect para a Xbox** da Microsoft. Este sensor é referido na bibliografia como uma opção viável para a deteção da imagem em profundidade e é utilizado em aplicações de interação gestual. É também um sensor utilizado noutras aplicações e testes efetuados pela investigadora noutras aplicações<sup>99</sup> de reconhecimento de movimento e interação com gestos. Existem também comunidades de programadores que desenvolvem aplicações com o recurso a este sensor, as quais se constituem como espaços privilegiados para a obtenção de documentação técnica e o esclarecimento de dúvidas. As suas características físicas e especificações de *hardware* fazem do sensor uma opção viável no contexto dos requisitos que se pretendem implementar no protótipo (interação gestual sem necessidade do utilizador segurar comandos). Permite detetar o movimento do utilizador em frente ao sensor, e assim obter a imagem para que esta possa ser processada pelo *software framework* **OpenNI** e do *middleware* **NITE**. Estes componentes através das técnicas de *computer vision* processam a imagem de forma a permitir o reconhecimento do utilizador e dos gestos executados.

O ambiente onde se realiza a interação é o motor de jogo **Unity3D** pois este apresenta características muito interessantes ao nível da integração das *frameworks* e do sensor de

---

<sup>99</sup> Foram desenvolvidas experiências com o sensor *Kinect* com os *softwares* *openFrameworks* e *Processing*.

movimento *Kinect*. Pode ser programado em **JavaScript**, o que do ponto de vista da experiência da investigadora é uma vantagem importante. O *Unity3D* é também referido na bibliografia, ao nível de ambientes virtuais 3D, como uma opção viável, flexível e competitiva entre os motores de jogo existentes no mercado (Keane et al., 2012).

Para a integração dos dados da *framework OpenNI* e do *middleware NITE* é necessário um componente que possibilite que esses dados possam ser processados pelo *Unity3D*. O **ZigFu** é um pacote de *scripts* em C# que permite aceder às funcionalidades e dados fornecidos pelos *OpenNI* e *NITE*. Esta ligação do *ZigFu* com as funcionalidades do *OpenNI* e do *NITE*, permite incluir no *Unity3D* as imagens provenientes do sensor. O *ZigFu* disponibiliza também componentes (ficheiros de código do *ZigFu*) que podem ser adicionados ao projeto.

No decorrer da implementação dos requisitos relacionados com a interação e comunicação em rede, o *ZigFu* revelou ser uma solução inviável. Assim houve um processo de pesquisa de novas soluções de forma a resolver os problemas enfrentados. A solução passou por uma mudança de *framework*, para a *OpenNI Wrapper*. Esta permite também a comunicação do *Unity3D* com os dados recolhidos pelo sensor. Mais à frente na descrição do processo de implementação do protótipo esta questão é abordada em detalhe.

Na Tabela 2 apresentam-se as diferentes tecnologias (*hardware* e *software*) adotadas para a implementação do protótipo.

TECNOLOGIA	DESCRIÇÃO
<b>Microsoft Kinect</b>	Captura do movimento do utilizador
<b>OpenNI e NITE</b>	Captura e deteção dos movimentos/ Formação do esqueleto do utilizador
<b>OpenNI Wrapper</b>	Comunicação entre a <i>Framework OpenNI/NITE</i> e <i>Unity3D</i>
<b>Microsoft Windows</b>	Sistema Operativo (ambiente de desenvolvimento)
<b>Unity3D</b>	Ambiente de renderização da animação

Tabela 2 – Resumo Tecnologias adotadas para o protótipo

Neste caso específico o sistema operativo utilizado para a implementação do protótipo é o *Microsoft Windows 7*, versão de 64bits. Do ponto de vista do utilizador, o protótipo pode ser exportado para o sistema operativo Windows ou Mac OSX.

### 4.3 Pré-requisitos

#### De Hardware e de Software

Para o correto funcionamento do *hardware* e *software* apresentado no ponto anterior, é necessário ter em conta vários fatores que podem interferir na forma como estas tecnologias operam. Por exemplo, um posicionamento incorreto do sensor pode impedir que a deteção do utilizador seja conseguida.

O *Kinect* conecta-se ao computador via porta USB. O computador deverá ser capaz de processar um fluxo de dados de 20MB por segundo de dados (no caso de um *Kinect*, ou multiplicar esse fluxo pelo número de sensores a utilizar). O computador deve ainda estar equipado com uma placa gráfica com suporte *OpenGL*<sup>100</sup>. Como já referido, o posicionamento e o modo utilização do sensor<sup>101</sup> são muito importantes para o seu funcionamento. Apresentam-se a seguir as principais recomendações neste âmbito.

#### Sensor *Kinect* - Iluminação

- Evitar expor o sensor à luz solar direta (Figura 46). Demasiada luz pode interferir com a captura do vídeo e conseqüentemente com a deteção do utilizador através da câmara de deteção em profundidade.



Figura 46 – Exemplo de posicionamento incorreto do sensor<sup>102</sup>.

<sup>100</sup> **OpenGL** (*Open Graphics Library*) API multiplataforma de utilização gratuita utilizada na renderização gráfica. Associada a aplicações de realidade virtual, visualização de informação, simulações e videojogos.

<sup>101</sup> <http://support.xbox.com/pt-PT/xbox-360/kinect/sensor-placement> consultado a 16/10/2012

<sup>102</sup> Imagem retirada de <http://support.xbox.com/en-US/xbox-360/kinect/lighting> consultado a 16/10/2012

### Sensor *Kinect* - Colocação

- O sensor deve estar colocado numa posição paralela ao chão e à parede e numa superfície plana;
- O dispositivo de captura deve estar colocado numa superfície estável e de preferência no limite dessa superfície, caso contrário a própria superfície pode interferir com o campo de visão das câmaras, como se demonstra na Figura 47;
- O sensor deve estar direcionado para o centro de massa do corpo do utilizador.

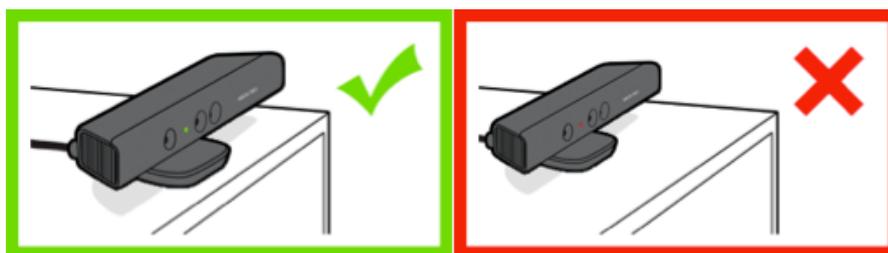


Figura 47 – Posicionamento correto do sensor<sup>103</sup>.

### Sensor *Kinect* - Roupa ou outros objetos

- Para um melhor reconhecimento do utilizador em frente ao sensor recomenda-se a utilização de roupa com cores e formas neutras. A utilização de roupas demasiado largas, ou sem forma, pode prejudicar esse reconhecimento.

### Sensor *Kinect* - Espaço envolvente

- Para a execução dos movimentos e interação com a aplicação é importante ter um espaço amplo para uma utilização confortável e sem objetos que possam interferir com a imagem capturada pelo sensor.

### Sensor *Kinect* – Ambiente com múltiplos utilizadores

- No caso da utilização do *Kinect* acontecer num ambiente com múltiplos utilizadores, é necessário prevenir erros de deteção do utilizador e problemas de interação. Assim para assegurar a forma mais correta de deteção (permitir que somente um utilizador é detetado) deve-se organizar o espaço de forma que os demais utilizadores não interfiram com o utilizador ativo (Microsoft, 2012).

<sup>103</sup> Figura retirada de <http://support.xbox.com/pt-PT/xbox-360/kinect/sensor-placement>

## Sensor *Kinect* – Tamanho do ecrã e resolução

- A escolha do ecrã de saída de imagem e a resolução deste deve ter em conta a distância a que o utilizador irá interagir com protótipo.

## 4.4 *Desenvolvimento prático do protótipo*

### 4.4.1 *Instalação da framework ZigFu no Unity3D*

O protótipo pretende ser um ambiente 3D colaborativo em que cada indivíduo pode controlar o seu avatar através de movimentos do corpo. Com recurso ao sensor *Microsoft Kinect* é possível recolher dados sobre estes movimentos em tempo real. Dados esses que são processados com o auxílio da tecnologia desenvolvida pela empresa Primesense<sup>104</sup> e atribuídos seguidamente a um modelo virtual - um avatar. Desta forma é possível animar este último de acordo com os movimentos naturais do utilizador, e essa animação ser visível para outros utilizadores da plataforma.

Foi desenvolvido um esquema do protótipo a desenvolver, com as principais áreas subjacentes à presente investigação. Disponibilizado no Anexo1 esse esquema permite analisar as componentes de *hardware* e de *software* utilizadas e a forma como estas se relacionam. A versão do protótipo aí mostrada foi sendo modificada no decorrer investigação e essas modificações são descritas nas secções seguintes.

Para a implementação do protótipo com o sensor de movimento *Kinect* foi necessária a integração da framework *ZigFu* para que os dados pudessem ser processados no *Unity3D* e permitir desse modo o controlo do avatar através de gestos.

O *download* do pacote da *ZigFu* pode ser feito no site (<http://zigfu.com/en/>). Aqui é possível fazer o *download* de dois ficheiros distintos, o primeiro pacote (Figura 48) (*ZigFu OpenNI Development Bundle*)<sup>105</sup> engloba os seguintes *drivers*:

- NITE: 1.5.2.21
- *OpenNI*: 1.5.2.23
- *SensorKinect*: 5.1.0.25 (*Kinect*)

---

<sup>104</sup> Website da empresa Primesense <http://www.primesense.com/> consultado em 14/12/2011

<sup>105</sup> Antes da instalação de qualquer tipo de *drivers* ou *software* relacionado com o sensor *Kinect* ou *OpenNI*, o *ZigFu* recomenda a desinstalação de outros *drivers* instalados no computador. O facto de haver outros *drivers* instalados pode interferir com o correto funcionamento do *software*.



Figura 48 – Instalação do Pacote de *Drivers*.

No site do *ZigFu*, (Figura 49), está também disponível para *download* o segundo ficheiro, anteriormente referido, o pacote de instalação **ZigUnity3D**. O *ZigUnity3D* contém os ficheiros que fazem a ligação entre os dados provenientes do sensor, e o modelo 3D.

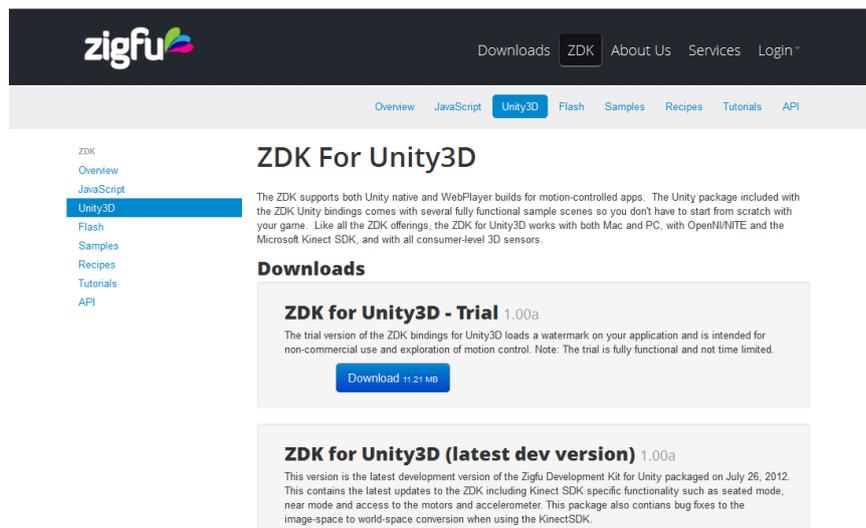


Figura 49 – Site *ZigFu*, *download* pacote para o *Unity3D*.

Após a instalação dos *drivers* necessários para o funcionamento do sensor no computador, iniciou-se o processo de implementação prática do protótipo no *Unity3D*. Esta etapa começou com a criação de um novo projeto no *Unity3D* para o desenvolvimento do ambiente virtual colaborativo com interação gestual. Em seguida descreve-se o processo de criação do protótipo.

A Figura 50 mostra a criação de um novo projeto no *Unity3D*. Nesta etapa não se deve selecionar nenhum *package* (não é necessário para o desenvolvimento). Caso seja necessária alguma funcionalidade de um *package*, existe a possibilidade de importar os mesmos *packages* mais tarde no *Unity3D*.

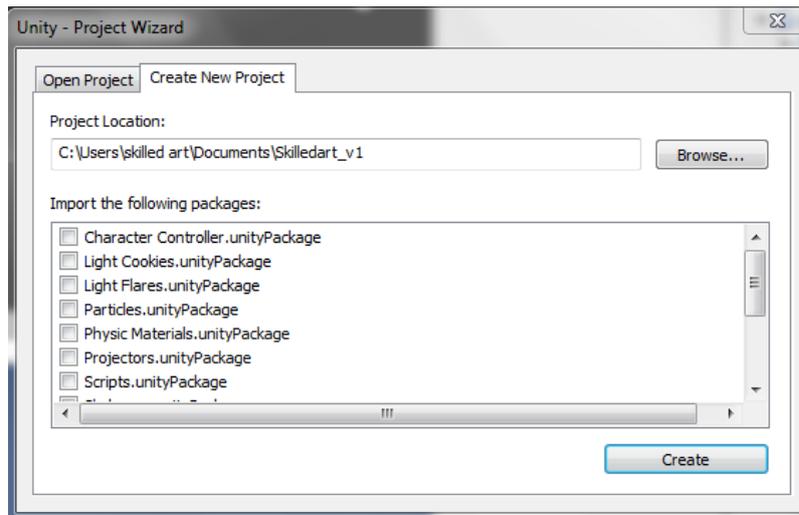


Figura 50 – Unity3D, criação de um novo projeto.

Após a criação do novo projeto, procedeu-se à importação do pacote do *ZigFu* para o projeto (*Assets -> Import Package -> Custom Package*), como demonstra a Figura 51, para assim se ter acesso às funcionalidades do sensor dentro do *Unity3D*.

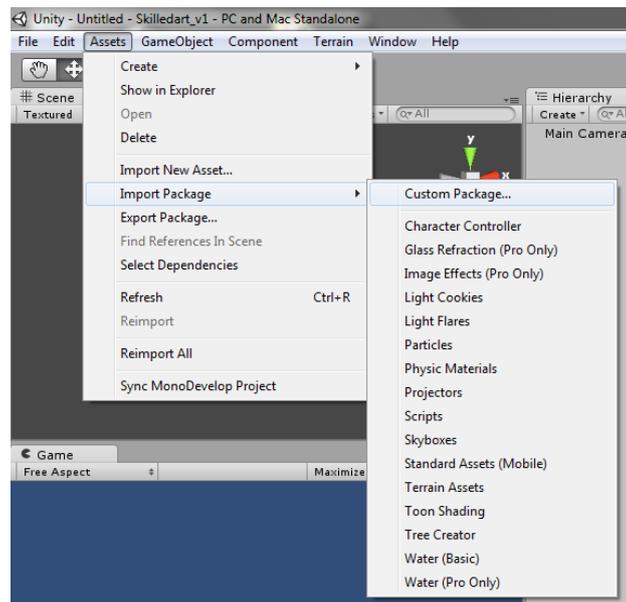


Figura 51 – Menu importação novo *package*.

É possível ver uma apresentação geral do projeto do protótipo na Figura 52 com integração da *framework ZigFu*, em modo de desenvolvimento. O *Unity3D* apresenta várias áreas distintas principais: **Scene**, **Hierarchy**, **Project**, **Inspector** e **Game**. A área **Scene** mostra os objetos no cenário, e pode-se manipular os elementos presentes no cenário (mover de posição, tamanho). Nesta área é também possível mudar a perspetiva de visualização (X,Y,Z). Na área do **Game** é

onde se pode visualizar todos os elementos que estão na cena a partir da perspectiva ativa da câmara. A área **Inspector** é onde se pode ver e modificar as propriedades dos objetos. Para se adicionar ficheiros de código aos objetos é aqui no **Inspector** que se adicionam. Na área de **Hierarchy** está disponível a listagem de todos os objetos que estão na cena. Na área **Project** estão todos os objetos do projeto, independentemente de estarem a ser utilizados na cena atual ou não.

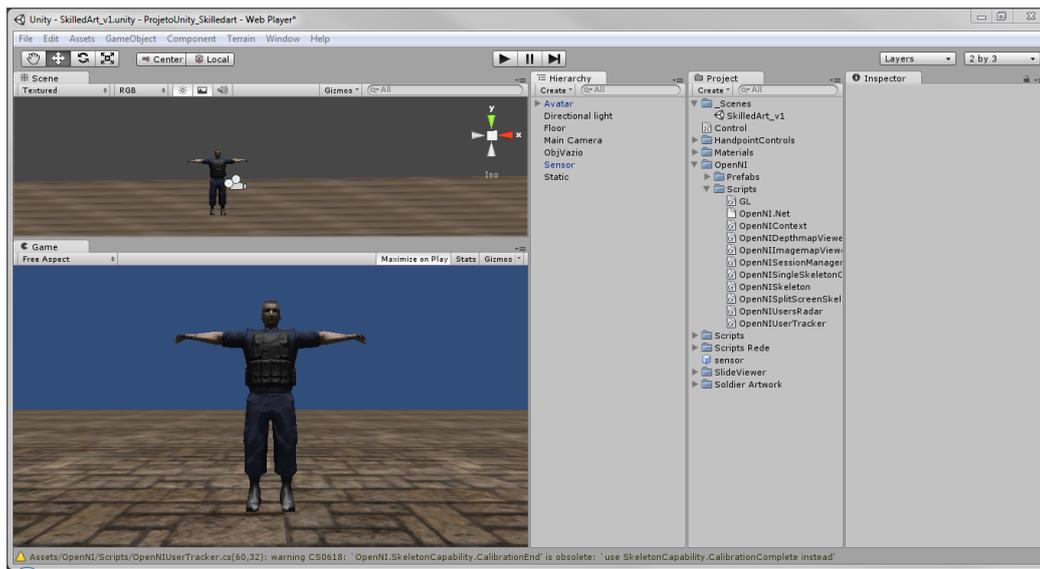


Figura 52 - Visão geral do projeto no *Unity3D* (framework *ZigFu*).

#### 4.4.2 Interação Utilizador - Avatar no ambiente 3D, principais funções da framework *ZigFu*

Os primeiros passos na implementação do protótipo permitiram compreender o funcionamento do *software Unity3D*, as suas características, a interface, e também programação em *JavaScript*. Nesta fase de trabalho foi fundamental a pesquisa na internet de tutoriais e vídeos, guias de utilização e programação na aplicação. Após a exploração da interface e do ambiente de trabalho no *Unity3D* o desenvolvimento começou com a colocação do avatar 3D (um avatar exemplo fornecido pela *framework*) em funcionamento. A Figura 53 mostra os testes realizados com um utilizador a interagir com gestos através dos dados recolhidos pelo sensor *Kinect* com o avatar no ambiente virtual 3D. Esta componente de interação foi bem-sucedida. A janela amarela no canto inferior direito mostra o utilizador em frente ao *Kinect*. Esta janela contém a imagem em profundidade (*depth map*) captada pelo sensor de movimento *Kinect*, e a informação relativa à distância entre os objetos e o sensor. Neste caso em específico a informação da distância entre o utilizador e o sensor.

“A depth map is an image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint” (Keane et al., 2012).

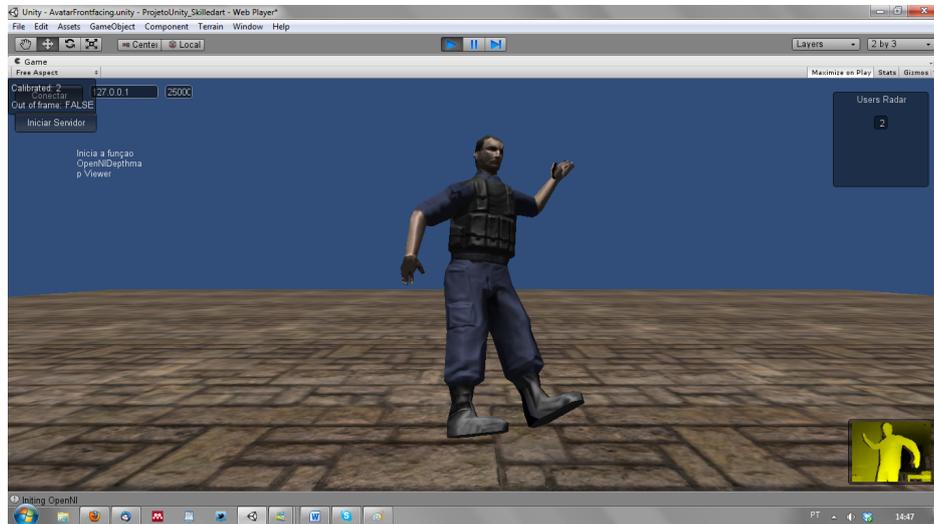


Figura 53 – Visão geral interação utilizador-avatar (framework ZigFu).

A Figura 54, na área *Inspector*, mostra a lista das principais “*joints*” (articulações) dos dados do esqueleto do utilizador que se pode obter através da *OpenNI*, e atribuir ao modelo 3D. A *OpenNI* comunica diretamente com o sensor de movimento *Kinect* e fornece a imagem em profundidade ao *middleware* NITE. O NITE processa a imagem sendo assim possível obter dados do “esqueleto” do utilizador. A classe responsável pela atribuição dos dados recebidos pelo NITE ao avatar é a classe “*OpenNISkeleton*”. Esta classe atribui os valores recebidos da posição dos principais *joints* (*mãos, pés, joelhos, cabeça*) ao modelo 3D, tornando assim possível o utilizador controlar o avatar através dos movimentos do corpo. Podem ser modificados vários parâmetros como a escala ou orientação no *Inspector* da classe.

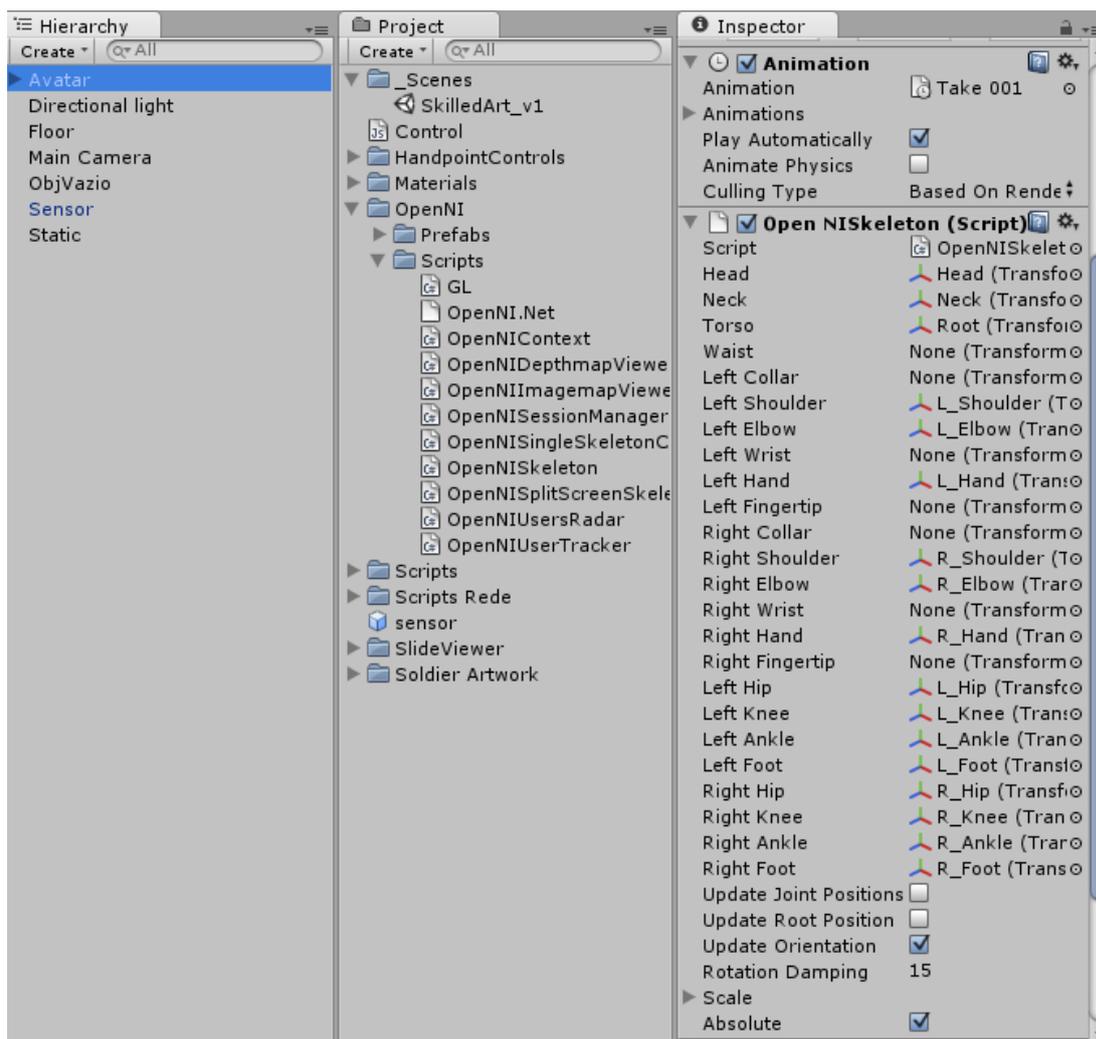


Figura 54 - Propriedades do Avatar 3D.

Na Figura 55 é possível ver as principais características do objeto **sensor**. Este tem associados vários scripts da *OpenNI*. O sensor é um objeto vazio (*empty object*), e tem apenas anexados *scripts* relacionados com a funcionalidade da *OpenNI* (fundamentais para a recolha da informação proveniente do sensor *Kinect*). Este objeto deve estar presente na cena de forma a permitir que os scripts sejam executados, juntamente com os outros conteúdos da cena.

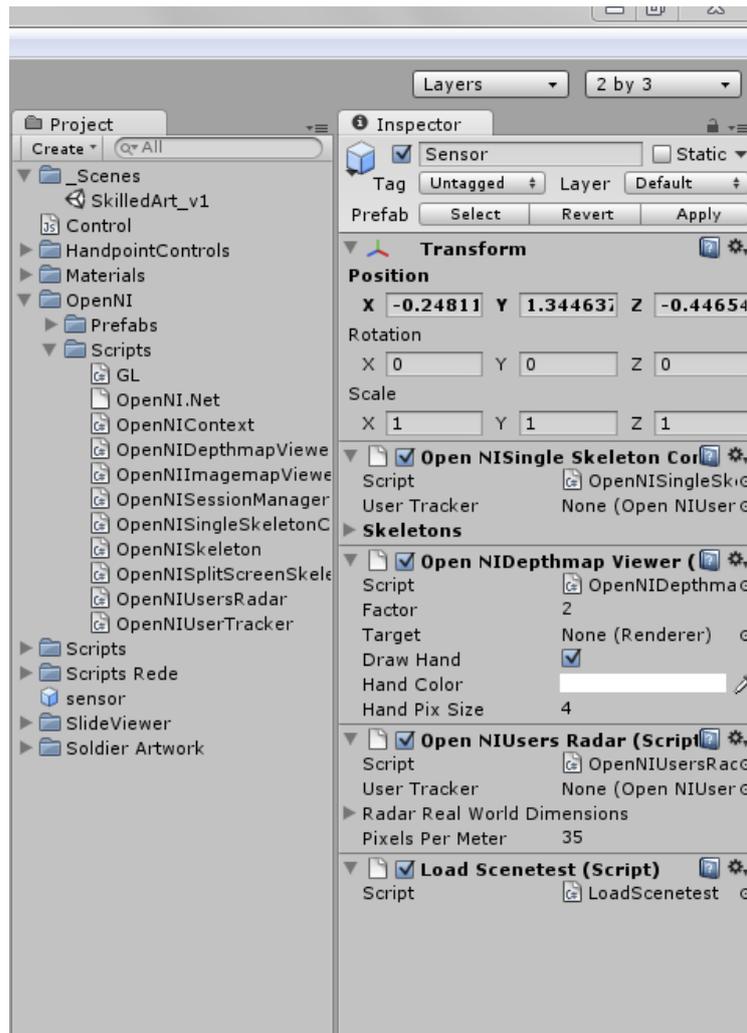


Figura 55 - Características do objeto "Sensor" (ZigFu).

Estas classes (*OpenNISingleSkeletonController*, *OpenNIDepthmapViewer* e *OpenNIUsersRadar*) estão disponíveis no pacote desenvolvido pela empresa *ZigFu*. São as classes que permitem aceder à imagem em profundidade proveniente do *Kinect* e depois processar as imagens pela *OpenNI* e *NITE* de forma a controlar o modelo 3D.

### 1. *OpenNISingleSkeletonController*

Atualiza a lista de objetos *OpenNISkeleton* e dos dados recolhidos de um utilizador. (Classe que regulamenta a classe do *skeleton*, responsável por regular mais do que um utilizador no mesmo computador a utilizar o mesmo dispositivo *Kinect*)

## 2. *OpenNIDepthmapViewer*

Esta classe faz o render da imagem de profundidade RGB recebida do sensor. A imagem em profundidade apenas é visível quando o programa estiver no modo *play*. Permite obter a imagem do utilizador onde depois é retirada a informação necessária para a construção do esqueleto.

## 3. *OpenNIUsersRadar*

Janela presente no canto superior direito (Figura 53), com uma pequena caixa cinza e um número. Sempre que se inicia a aplicação e é detetado o utilizador, a caixa representa o utilizador e o número é o ID.

### 4.4.3 *Comunicação em rede, criação de um servidor e de um cliente (Projeto com framework ZigFu)*

Após os testes de interação do utilizador com o avatar terem sido bem-sucedidos, deu-se início ao desenvolvimento da componente de rede do projeto. Esta foi identificada claramente como a área mais complexa do desenvolvimento, quer pelas suas características e exigências ao nível da programação, quer pelo seu carácter de novidade dado que não foram encontrados projetos multiutilizador em rede com sensores *Kinect*. Nesta etapa foi fundamental a consulta de vários tutoriais<sup>106</sup> sobre esta temática.

O conceito base da **comunicação em rede** consiste na transmissão de dados entre vários computadores. Para tal é necessário um **cliente** e um **servidor**, em que o servidor pode ser um computador especialmente dedicado a atender pedidos dos clientes, ou pode ser um cliente que funcione também como servidor. Uma vez estabelecida a comunicação entre as máquinas (cliente e servidor) é possível fazer a troca necessária de dados.

Para estes testes, o protótipo foi desenvolvido tendo em conta apenas dois computadores (ambos em ambiente *Windows*) de forma a colocar um servidor e um cliente em comunicação. Foi montada uma pequena rede para a realização das experiências, a qual permitia ter um total controlo sobre o seu funcionamento.

O *Unity3D* nas questões de rede, especificamente em relação ao tipo de servidor, apresenta duas tipologias distintas: *Authoritative Server* e *Non-Authoritative Server*. A configuração do servidor do tipo *Authoritative Server* implica a existência de um computador que efetua todo o processamento e lógica de rede. O cliente não executa a lógica, apenas "informa" o servidor do que pretende

---

<sup>106</sup> <http://unity3d.com/support/resources/tutorials/iphone-multiplayer-tutorial>

fazer. Na tipologia de servidor *Non-Authoritative Server* não existe um computador específico com a função de servidor. Cada computador cliente processa os dados e envia somente os resultados desse processamento para o servidor para depois o servidor atualizar e sincronizar as ações de todos os computadores (Unity3D, 2012).

No contexto do desenvolvimento deste protótipo, o tipo de servidor que se criou é do tipo ***Non-Authoritative Server***. Cada computador processa os dados e envia somente os resultados desse processamento para o servidor, depois o servidor atualiza o estado dos objetos no cenário de todos os outros computadores ligados à rede. Nesta tipologia cada computador tem de lidar com os seus eventos localmente. E apenas enviam a atualização para o servidor para posterior sincronização com os restantes elementos da rede. Do ponto de vista da estrutura, este sistema é mais fácil de implementar, uma vez que o servidor apenas recebe as mensagens de atualização de estado e apenas tem de sincronizar a atividade.

A Figura 56 mostra a declaração das variáveis de rede. As variáveis são declaradas no início do ficheiro *ConnectionGUI.js*. Este ficheiro está associado ao objeto câmara presente no cenário. Pretende-se que este ficheiro seja executado assim que a cena é iniciada. É neste ficheiro de código que é criada a interface que permite ligar/desligar o servidor ou ligar a um servidor existente (no caso do cliente). O código está escrito em *JavaScript*.

```
var remoteIP = "127.0.0.1";
var remotePort = 25000;
var listenPort = 25000;
var useNAT = false;
var yourIP = "";
var yourPort = "";
```

Figura 56 - Declaração das variáveis de rede.

A Figura 57 mostra a primeira parte da função (OnGUI) da criação da referida interface. A primeira verificação da função permite saber se já existe uma ligação efetuada ao servidor, no caso de não haver, cria a interface para o utilizador se poder conectar ao servidor (*Network.Connect(remoteIP, listenPort)*) ou cria um novo servidor (*Network.InitializeServer(32, listenPort)*).

Na interface estão disponíveis dois campos de texto para se inserir o IP (*Internet Protocol*) do computador servidor ao qual o utilizador se pretende ligar e a porta. Os valores introduzidos nestas áreas de texto são guardados nas variáveis *remoteIP* e *remotePort*.

```

function OnGUI () {
    //verifica se a ligação ao servidor está estabelecida;
    if(Network.peerType == NetworkPeerType.Disconnected) {

        if(GUI.Button (new Rect(10,10,100,30), "Conectar")) {

            Network.useNat= useNAT;
            //ligação ao servidor (esta noutra pc)
            Network.Connect(remoteIP, listenPort);

        }

        if(GUI.Button (new Rect (10,50,120,30), "Iniciar Servidor")) {

            Network.useNat =useNAT;

            //criação do servidor
            Network.InitializeServer (32, listenPort);

            //GUILayout.Box("Servidor inicializado com sucesso =)");
            //notificar os objetos do jogo que a rede esta pronta
            for(var go:GameObject in FindObjectsOfType(GameObject)) {
                go.SendMessage("OnNetworkLoadedLevel", SendMessageOptions.DontRequireReceiver);
            }

            //campos para a inserção dos dados para ligar a um servidor
            //campos para a inserção do IP e a porta
            remoteIP = GUI.TextField(new Rect(120,10,100,20), remoteIP);
            remotePort = parseInt(GUI.TextField(new Rect (230,10,40,20), remotePort.ToString()));

        }else { //se estiver conectado

```

Figura 57 - Função OnGUI (primeira parte).

A segunda parte da função OnGUI (Figura 58) mostra o código que faz a verificação do estado da ligação ao servidor. Se o servidor já estiver criado, é colocado o IP do utilizador e a porta que está a usar para a conexão.

```

}else { //se estiver conectado

    //ip e porta do pc do utilizador
    ipaddress = Network.player.ipAddress;
    port = Network.player.port.ToString();

    //display dos dados
    GUI.Label( new Rect(140, 20, 250, 40), "Endereço IP: "+ipaddress+" "+port);
    if (GUI.Button (new Rect(10, 10, 100, 50), "Desconectar")) {
        //desconexao do servidor
        Network.Disconnect(200);
    }

}

}
}

```

Figura 58 - Função OnGUI (segunda parte).

A Figura 59 mostra a interface que permite a iniciar o servidor ou conectar a um servidor já criado, criada através do código previamente apresentado. Contém dois campos de texto para inserir o IP e a porta do servidor.

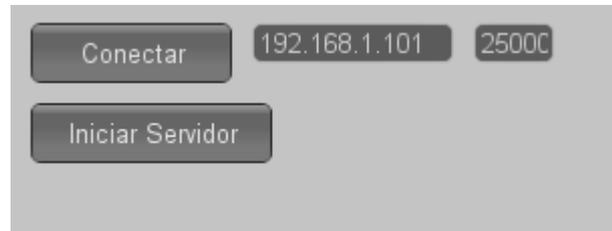


Figura 59 - Interface da rede (criar ou conectar a um servidor).

Se o resultado da verificação do estado da ligação ao servidor for verdadeiro, ou seja se já existir uma ligação ao servidor, ou um servidor criado, é colocado o IP do utilizador e a porta que está a usar (Figura 60). Na interface o botão de desconectar dá ao utilizador a possibilidade de se desligar do servidor.



Figura 60 – Desconectar do servidor.

A Figura 61 mostra a função *OnConnectedToServer*, que executada (noutro ficheiro) sempre que alguém se liga com sucesso à rede e quando isso acontece todos os objetos no cenário são notificados de que a rede está a funcionar.

```
function OnConnectedToServer () {  
  
    //notificação dos objetos de que o nível e a rede está pronta  
    for ( var go : GameObject in FindObjectsOfType(GameObject))  
        go.SendMessage("OneNetworkLoadedLevel", SendMessageOptions.DontRequireReceiver);  
}
```

Figura 61 – Função OnConnectedToServer.

Este tipo de servidor como está criado, apenas funciona na mesma rede, o que para efeitos de teste do protótipo é suficiente. Fica assim desenvolvida a interface de criação de um servidor e a possibilidade de vários computadores (clientes) se ligarem.

Agora é necessário atribuir a componente de rede aos objetos que se querem na rede, ou seja, dizer que o objeto, por exemplo o avatar, é para sincronizar na rede e ser visto por todos os outros utilizadores.

Assim é necessário atribuir a componente **Network View** aos objectos (*Components-> Miscellaneous-> Network View*). A componente *Network View* é responsável pela transmissão dos dados (por exemplo, a posição ou a orientação) do objeto na rede. Esta componente deteta as alterações de estado do objeto e assegura-se de que essas alterações são visualizadas também nos outros clientes ligados à rede, ou seja, é efetuada uma sincronização do estado do objeto (*state synchronization*). *State Synchronization*, é o método utilizado pois pretendeu-se que todos os movimentos dos objetos na cena que estejam em constante modificação fossem sincronizados (como por exemplo o caso dos movimentos do avatar de cada utilizador). O outro método possível *Remote Procedure Calls*, é utilizado sempre que se pretende executar funções nos computadores dos clientes

**Network.Instantiate**, é a função que permite que o *prefab*<sup>107</sup> avatar seja instanciado em todos os clientes da rede, ou seja, o objeto vai ser sincronizado na aplicação de todos os clientes.

#### 4.4.4 Framework ZigFu – Problemas no desenvolvimento

No decorrer do desenvolvimento do protótipo, com a utilização da *framework ZigFu*, ocorreram vários problemas, os quais se descrevem a seguir.

Através de vários testes efetuados verificou-se a incompatibilidade na atribuição da componente de rede "*Network View*" e da função "*OpenNIUserTracker*" ao avatar. Pretendeu-se sincronizar o avatar na rede e simultaneamente recebesse os dados da *framework ZigFu* para ser controlado através dos gestos. Assim quando se fez a instanciação (colocando o avatar disponível para sincronizar na rede) o avatar não deveria estar no cenário, pois este só era colocado no cenário após a inicialização das opções de rede. Em contrapartida, para o reconhecimento dos gestos funcionar era necessário o objeto estar no cenário, (para que se pudesse atribuir as funções necessárias ao seu funcionamento com o sensor).

Para resolver este problema foram efetuados vários testes, como por exemplo, instanciar um objeto (neste caso o avatar) na rede, e que está presente no cenário. Estes testes deram erro na deteção e atribuição do movimento ao utilizador, ou seja, a componente de interação não era iniciada.

---

<sup>107</sup> Um **prefab**, no contexto do Unity3D, é um conjunto de *gameObjects* e componentes que podem ser reutilizados sempre que necessário.

Foi criado um teste com dois avatares e um sensor de movimento. Um primeiro avatar com a componente de interação e o segundo avatar que reproduz os movimentos do primeiro, mas que é sincronizado na rede. A ideia subjacente a este teste é de ter um avatar no cenário para a componente de interação, e um segundo avatar que é instanciado posteriormente na rede. Este sistema não funcionou na sua totalidade, e numa possível extensão a outros objetos poderia tornar o sistema complexo e lento.

Esta *framework* inviabilizou a possibilidade do protótipo ser simultaneamente de interação gestual e funcionar em rede. A *ZigFu* deixou de ser uma solução viável pois não permitiu a implementação de alguns requisitos definidos para o projeto.

Neste ponto do problema, foram colocadas várias dúvidas *online* em fóruns de debate e suporte do *Unity3D*, com o principal objetivo de tentar resolver este problema, ou soluções alternativas. As várias tentativas de contacto via fóruns e mesmo email não foram bem sucedidas.

Os problemas identificados com a utilização da *framework ZigFu* não permitiram os avanços desejados na implementação do protótipo. A incompatibilidade verificada entre a componente de rede e a de interação impossibilitaram a criação de um ambiente virtual 3D distribuído.

#### **4.4.5 Framework OpenNI Wrapper**

Como referido atrás, o desenvolvimento do ambiente virtual 3D com a utilização da *framework ZigFu* não foi bem sucedido. O facto da *framework* não estar preparada para ser utilizada num contexto de comunicação em rede impediu a implementação das funcionalidades de rede e interação gestual com o avatar na sua totalidade. Fizeram-se várias pesquisas em fóruns e grupos da especialidade na tentativa de resolver os problemas, mas não foram encontradas respostas. Assim, em conversa com outras pessoas a trabalhar na área, foi apresentada uma *framework* que estava a ser usada num projeto académico<sup>108</sup>, na Universidade de Aveiro. A partir deste momento foram estabelecidos contactos, com um colega de outra licenciatura, em ECT (Engenharia dos Computadores e Telemática) onde foi possível trocar ideias ver o problema de uma forma diferente. A possibilidade de mudar de *framework* foi uma sugestão que foi considerada. A *framework OpenNI Wrapper* possibilita integrar os dados recolhidos pelo sensor através da *OpenNI* e NITE para utilizar no *Unity3D*.

---

<sup>108</sup> Vídeo do Projeto *Kinect Challenge* <http://videos.ua.sapo.pt/paulo-fonseca/0cZBe2ZFpgK3zTK0e5gE>

Nesta fase do desenvolvimento, a ideia foi conseguir ultrapassar o problema encontrado com a *framework* anterior (*ZigFu*), ou seja, a incompatibilidade entre a animação do avatar e a respetiva sincronização na rede.

#### 4.4.5.1 Instalação dos *Drivers*

Para o trabalho com a nova *framework* foi necessário recomeçar o processo, para garantir que tudo ficaria operacional. Assim procedeu-se a uma desinstalação de todos os *drivers* instalados. Neste recomeçar da implementação do protótipo, o *Unity3D* também foi desinstalado e instalada a versão 3.5.2 do *Unity3D* para Windows<sup>109</sup>, disponibilizada gratuitamente no site.

Para o funcionamento do sensor *Kinect* no computador, ou seja, para ser possível aceder às imagens provenientes do sensor no *Unity3D* foi necessário instalar os *drivers*.<sup>110</sup> Os *drivers* instalados no computador foram os seguintes:

- *Framework OpenNI* 1.5.4
- *Middleware NITE* 1.5.2.21
- *Driver* do Sensor *Kinect* 5.1.2.1

Para fazer a ligação entre o sensor e o *Unity3D*, o recurso foi a *framework* ***OpenNI Unity Wrapper***<sup>111</sup>. Este pacote desenvolvido especialmente para o *Unity3D* é baseado na *framework* *OpenNI* e permite desenvolver aplicações controladas por gestos de uma forma mais rápida no *Unity3D*, evitando a necessidade de conhecer de uma forma detalhada a *OpenNI*. Esta *framework* está assim desenvolvida e preparada para ser utilizada no *Unity3D*, disponibilizando um conjunto de *prefabs*, objetos configurados para utilizar nos projetos.

#### 4.4.5.2 Criação do Projeto no *Unity3D OpenNI Wrapper*

No *Unity3D* criou-se um novo projeto, sem selecionar qualquer tipo de *asset* disponível na lista (como aconteceu na criação do primeiro projeto). Depois foi necessário importar a *framework* do *OpenNI*, após o *download* da área de recursos do *OpenNI*. Para se proceder à importação do *package*, o processo foi o seguinte: Aceder à área ***Assets***, localizada no menu superior do *Unity3D*, depois ***import package*** e a opção ***custom package***.

---

<sup>109</sup> Área de *download* do ficheiro de instalação do *unity3D* <http://unity3d.com/unity/download/>, consultado a 9/10/2012.

<sup>110</sup> É necessário assegurar que mais nenhuns *drivers* ou *software* relacionado com o *Kinect*, para além do que é referido, está instalado no computador. No caso de existirem outros *softwares* é necessário proceder a uma desinstalação dos mesmos.

<sup>111</sup> Área de *download* da *framework* <http://openni.org/Downloads/OpenNIModules.aspx>

Após importação dos ficheiros da *framework*, o trabalho incidiu na construção da componente de rede do projeto. Tal como no projeto anterior, a implementação começou pela componente de rede. Esta parte já estava previamente construída, apenas se efetuou uma cópia do código dos ficheiros para este novo projeto.

Em seguida, por forma a colocar o avatar em condições de poder ser sincronizado na rede, foi necessário atribuir o componente **Network View** ao avatar. Esta atribuição do componente *Network View*, tem uma particularidade. Como se pretende que não só o avatar seja sincronizado na rede (posição X,Y,Z do avatar), mas também todos os movimentos do esqueleto se pretende que sejam visíveis para todos os outros utilizadores. Assim desta forma foi necessário que a componente *Network View* fosse atribuída a cada *joint* de forma individual. A Figura 62 mostra a lista de todos os itens que precisam da componente para os movimentos do avatar estarem corretamente sincronizados na rede. A Figura 63 mostra um dos elementos individuais do esqueleto com a componente de rede. Esta questão foi inicialmente um problema, mas após várias pesquisas e tentativas falhadas percebeu-se então que não era suficiente sincronizar o objeto como um todo, mas sim cada componente individual do objeto tinha de ter a componente de rede. Esta questão é muito importante pois permitiu que os movimentos naturais do utilizador detetados pelo sensor fossem reproduzidos no seu próprio computador, mas também no computador dos outros utilizadores ligados à rede.

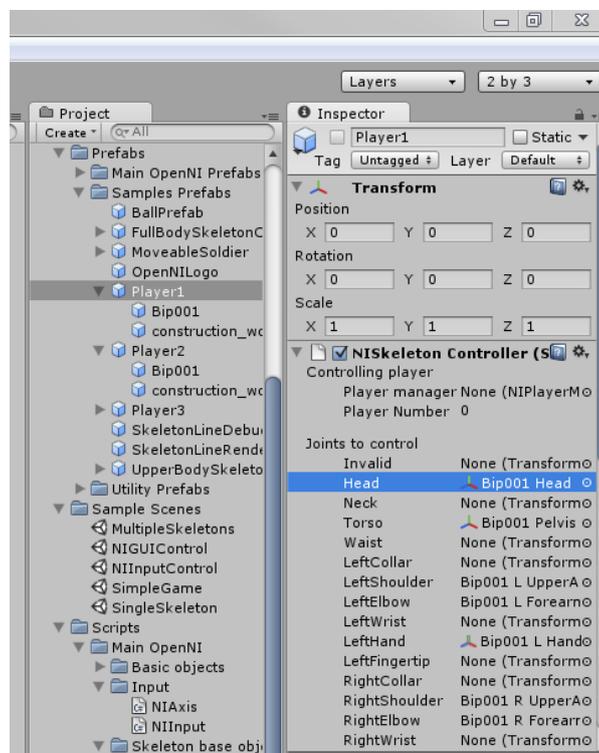


Figura 62 - Joints do esqueleto do avatar.

Esta *framework* possibilitou a sincronização do avatar em rede e simultaneamente controla-lo através dos gestos. A *framework ZigFu*, como explicado previamente, não permitiu desenvolver estas duas componentes em conjunto. A estrutura dos elementos constituintes da *framework OpenNI Wrapper* permitiu, ao contrário da *framework ZigFu*, conjugar a componente de sincronização em rede e de controlo do avatar.

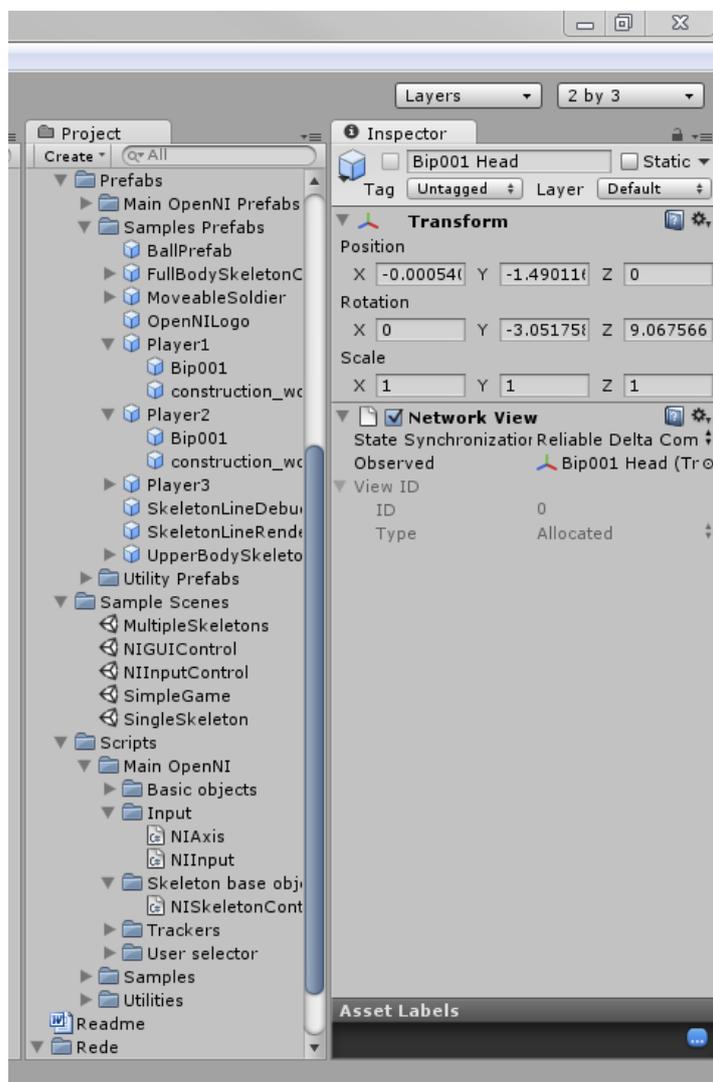


Figura 63 – Joint “head” do esqueleto do avatar com a componente *Network View*.

Para que o avatar fosse reconhecido na rede foi necessário proceder à sua instanciação na rede. A Figura 64 mostra o código da função *OnNetworkLoadedLevel*. A função *Instantiate*, pertence à classe *Network*.

```
Intantiate ▶ OnNetworkLoadedLevel()
1 var avatar0: Transform;
2
3 //OnNetworkLoadedLevel
4 function OnNetworkLoadedLevel () {
5
6     //Instanciar o avatar quando a rede é iniciada.
7     Network.Instantiate(avatar0,new Vector3(0,0,0), transform.rotation, 0);
8
9
10 }
11
12 function OnPlayerDisconnected (player: NetworkPlayer){
13
14     Network.RemoveRPCs(player, 0);
15     Network.DestroyPlayerObjects(player);
16 }
17
18
19
```

Figura 64 – Instanciar o avatar.

Para o processo de instanciação estar concluído foi necessário associar o avatar(Objeto *Player0*) à variável criada (*avatar0*). Este processo fez-se no *inspector* do objeto “Instanciar” que se encontra no projeto, como se pode ver na Figura 65.

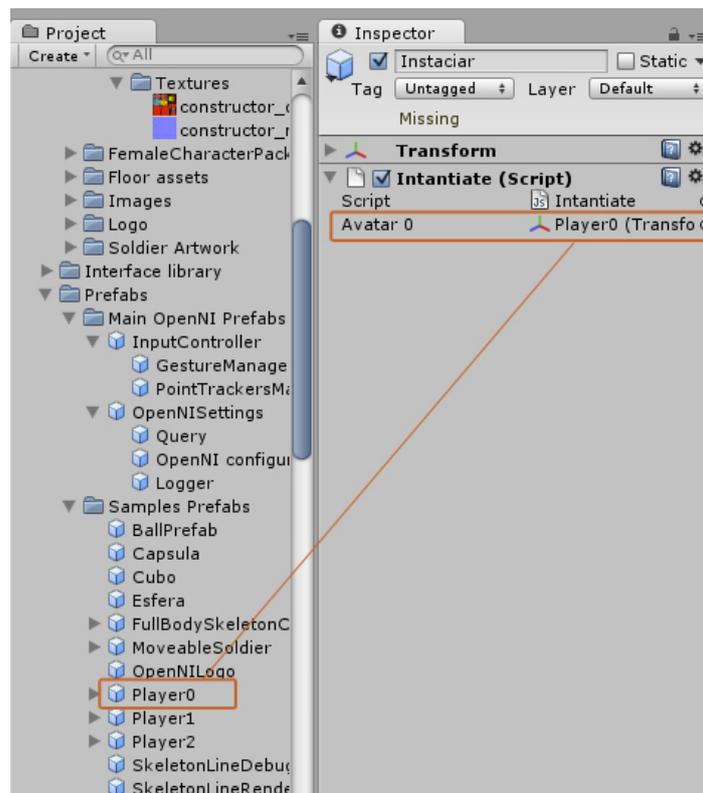


Figura 65 – Variável avatar0.

O primeiro teste efetuado com o recurso a esta *framework*, a animação do avatar na aplicação em rede foi bem-sucedida. De uma forma resumida, o utilizador controla o avatar com gestos no seu computador, e todos os outros utilizadores conectados podem ver o avatar em movimento. Em algumas circunstâncias a animação apresentou um comportamento irregular, apesar das condições de teste serem aparentemente as mesmas. Ou seja existem interferências com a animação que se vê nos outros utilizadores da rede. Este teste foi realizado tendo em conta apenas um utilizador e um *Kinect*, mas outros utilizadores a visualizar a aplicação.

Através da utilização desta *framework* (*OpenNI Wrapper*) já foi possível ultrapassar a limitação técnica (ocorrida com a *framework ZigFu*) com a sincronização do avatar em rede e o controlo através dos gestos. Neste ponto da implementação e teste do protótipo a animação e a sincronização ocorreram de forma simultânea.

A próxima etapa foi tornar o sistema verdadeiramente multiutilizador, ou seja colocar vários avatares e vários sensores em simultâneo a comunicar no protótipo. Para isso foi essencial colocar mais avatares. Para isso fez-se uma duplicação do avatar existente.

A Figura 66 mostra o código que instancia os avatares na rede. Foi necessário instanciar na rede estes novos avatares, assim o código do ficheiro *Instantiate.js* foi modificado para que sempre que algum utilizador novo se liga à rede, lhe seja atribuído um avatar. A variável responsável por essa identificação dos novos utilizadores é o *index*.

```
if(Network.isServer){
    Network.Instantiate(avatar0, transform.position, transform.rotation, 0);
    boneco = GameObject.Find('Player0(Clone)');
    boneco.transform.position = Vector3(-2, 0,0);
    //playerClone = "Player0(Clone)";
}
else if(index == "1") {
    Network.Instantiate(avatar1, transform.position, transform.rotation, 0);
    bonecol = GameObject.Find('Player1(Clone)');
    bonecol.transform.position = Vector3(0, 0,0);
}
else if(index == "2")Network.Instantiate(avatar2, transform.position, transform.rotation, 0);
```

Figura 66 – Instanciar três avatares.

Apesar desta instanciação de forma individual do utilizador, os testes efetuados demonstraram que a *framework* não está preparada para interpretar os dados recebidos de mais do que um *Kinect*. Quando mais do que um sensor está ligado à aplicação em rede, o avatar controlado é sempre o mesmo, pois a *framework* atribui sempre o mesmo *Kinect* a esse avatar (apesar de existirem vários em simultâneo). Esta dificuldade da *framework* em interpretar os dados de mais do que um sensor impediu que mais do que um avatar fosse controlado por diferentes sensores.

Assim, a *OpenNI Wrapper* permite controlar um avatar, quando os dados são recolhidos por um sensor. No caso de se incluir mais do que um sensor para o controlo dos movimentos de um segundo avatar, a *framework* não consegue identificar os sensores como dispositivos individuais e assume que são apenas um, assim, independentemente do *Kinect*, o avatar que é controlado é sempre o mesmo.

Foram estabelecidos contactos com outros investigadores na área, mas não resolveu o problema. Foram também desenvolvidas várias pesquisas no sentido de perceber qual a origem do problema e eventuais soluções para este, mas sem sucesso. Da análise das várias classes da *framework* não foram encontradas evidências de como identificar individualmente os sensores e assim atribuir um avatar a cada sensor. Esta dificuldade impediu a implementação na totalidade da componente multiutilizador do projeto.

Para se verificar que o problema residia na componente de interação (*framework* e sensor) foram feitos testes de comunicação em rede e de controlo de elementos na cena através das teclas. Tal permitiria validar a componente de rede sem a componente de interação via sensor de movimento. As experiências foram feitas através do controlo (de um cubo e de uma esfera) com teclas (alteração da posição X e Y dos elementos). Através das teclas do computador foi possível deslocar os objetos e sincronizar no outro utilizador. Estes testes foram essenciais para perceber que o problema da comunicação em rede anteriormente verificado não existiu quando se retirou a componente de interação via sensor de movimento.

Para a implementação dos testes descritos anteriormente foi desenvolvido o seguinte código A Figura 67 para instanciar os novos objetos na rede (cubo e esfera). Estes também têm a componente (*Network View*) associado e podem ser controlados com as teclas.

```
//TESTE INSTANCIAR CUBO E ESFERA

Network.Instantiate(cubo, transform.position, transform.rotation, 0);
ncubo = GameObject.Find('Cubo(Clone)');
ncubo.transform.position = Vector3(2, 0,0);

Network.Instantiate(esfera, transform.position, transform.rotation, 0);
nesfera = GameObject.Find('Esfera(Clone)');
nesfera.transform.position = Vector3(-2, 0,0);
```

Figura 67 – Instanciar o cubo e esfera.

O aspeto diferente entre os vários testes é a entrada de dados, neste caso via teclas, e na situação dos avatares a interação é processada via sensor *Kinect* e a *framework OpenNI Wrapper*.

A Figura 68 mostra a deslocação dos objetos (cubo e esfera) através do controlo por teclas na rede. Um utilizador num computador controla o cubo e o outro utilizador noutra computador controla a esfera. O processo de instanciação dos objetos na rede e de atribuição da componente de rede é similar ao que foi efetuado nos testes com os avatares.

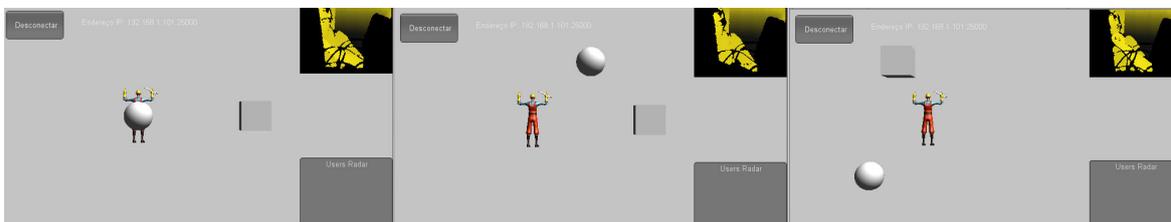


Figura 68 – Movimentos do cubo e da esfera através do controlo por teclas.

#### 4.4.5.3 Design de interação - *Framework OpenNI Wrapper*

Para o desenvolvimento da componente de interação gestual no protótipo, a *framework OpenNI Wrapper* foi essencial para este desenvolvimento.

Apresenta-se em seguida as principais funções essenciais para o funcionamento da interação gestual com a aplicação. Estas são as principais funções e que permitem atribuir ao avatar os movimentos do utilizador.

1. ***OpenNISettings Prefab*** – Este *prefab* é responsável pelas principais opções de configuração da *Framework OpenNI*. É responsável pela inicialização da *framework* no ambiente do *Unity3D*. É constituída por três *Game Objects*: *Logger*, *Query* e *OpenNI Configuration*. Neste *prefab* é possível definir opções como por exemplo: Acesso à câmara RGB e *User Generator* (caso seja necessário aceder às propriedades de criação do esqueleto).
2. ***OpenNI Skeleton Control*** - As funções associadas a este pacote permitem aceder à informação do *skeleton*, e utilizar essa informação para o controlo dos modelos 3D. Nestas funções são reconhecidas os principais pontos de referência do esqueleto.
3. ***Input Management*** – Função associada à possibilidade de interação natural com gestos, através da deteção de pontos específicos (como a mão) e a gestão dos gestos e as ações correspondentes.

Estas são as principais funções responsáveis pela interação utilizador-avatar, acedidas através da *framework OpenNI Unity Toolkit*. As quais processam os dados recolhidos pelo sensor e permitem controlar os avatares.

#### 4.4.6 Desenvolvimento da Interface

A Figura 69 mostra a interface do menu da aplicação. É constituída por um menu, com três botões onde o utilizador clicar. Através da interação gestual, o utilizador pode selecionar a área que pretende. No canto superior direito está a imagem em profundidade do utilizador. Esta imagem permite dar uma indicação visual do posicionamento do utilizador em relação ao sensor. Após o utilizador ser detetado e reconhecido, ele pode selecionar a opção desejada. O quadrado cinza (perto do título) é o que corresponde à mão do utilizador, ou seja, quando o utilizador interage com a mão direita (no exemplo da Figura 69) o quadrado desloca-se de acordo com os movimentos do utilizador. No desenvolvimento é possível definir qual é o ponto (*joint*) que se pretende detetar e com o qual se controlará os elementos da interface. Assim é possível alterar as opções no projeto de desenvolvimento para funcionar com a mão esquerda. No canto inferior direito está uma área onde aparecem os utilizadores que estão a ser detetados sob a forma de quadrado. Este quadrado pode ter 4 cores distintas dependendo da fase do processo de deteção e identificação utilizador pelo sensor.



Figura 69 – Menu Aplicação.

Em seguida apresenta-se a lista de cores e respetivos estados associados à cor do quadrado, representativo do utilizador.

- Vermelho – quando o utilizador é detetado, mas não está calibrado.
- Cor de laranja – esta cor indica que o utilizador está em processo de calibração.
- Amarelo – Quando o utilizador está calibrado.

- Verde – O utilizador está calibrado e a ser detetado e é o utilizador principal, ou seja o que controla a aplicação (no caso de estarem mais pessoas a serem detetadas pelo sensor).

A calibração é o processo que permite a *framework* identificar uma forma humana no campo de visão do sensor, e após ser detectada essa forma é construído um “esqueleto”. Através deste esqueleto gerado pode-se obter os dados das coordenadas de cada articulação (*joint*) e atribuir ao modelo 3D.

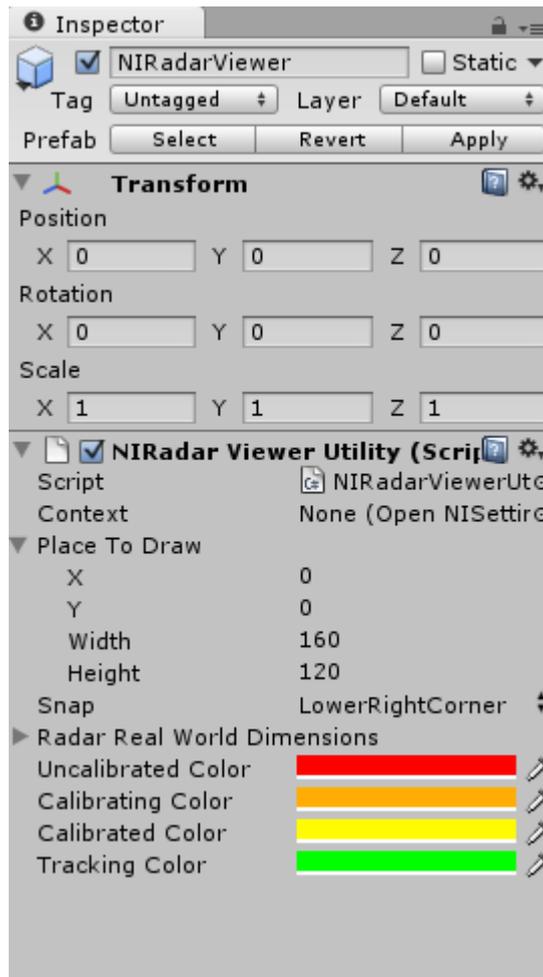


Figura 70 – Opções do quadrado de representação da deteção de utilizadores.

Quando o utilizador escolhe uma opção do menu, o retângulo apresenta uma barra de progresso que indica o tempo que o utilizador deve estar parado sobre o objeto para o selecionar. A Figura 71 mostra o utilizador a selecionar um botão e tem uma indicação visual. Para o utilizador selecionar o item do menu ele deve estar parado durante 1 segundo. No menu, os botões tem um estado *hover*, quando o utilizador passa por cima do botão, o botão fica sem contorno e o texto do botão passa a azul. Este é um indicador visual para o utilizador e que dá indicação de que o botão pode ser selecionado.

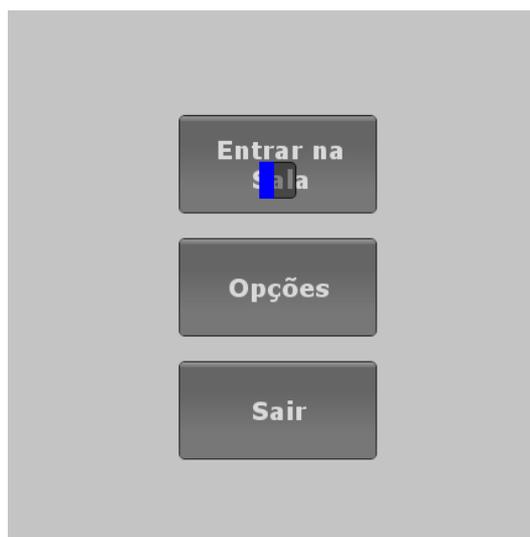


Figura 71 – Seleção do elemento.

A implementação deste protótipo permitiu testar tecnologias que possibilitam o desenvolvimento de aplicações com interação gestual. Foi um processo de pesquisa, investigação e implementação de soluções para os problemas enfrentados. Sempre que possível as recomendações teóricas referidas na bibliografia foram consideradas, mas a implementação prática por vezes tornou-se complexa. A construção de uma aplicação desta natureza implicou considerar as características (vantagens e limitações) das tecnologias utilizadas, pois influenciaram a construção do protótipo.

## 5 Conclusões

Este capítulo pretende apresentar e discutir as conclusões sobre a investigação desenvolvida nesta dissertação. Uma discussão tendo em conta a questão de investigação e os objetivos do projeto, e também sobre o desenvolvimento prático do protótipo. Em seguida apresentam-se igualmente as limitações ao estudo e perspetivas de trabalho futuro.

O presente estudo foi orientado pela questão de investigação: **Qual o potencial e limitações das atuais tecnologias de suporte à interação controlada por gestos na promoção de novas formas de interação em ambientes 3D distribuídos?**

Para se responder a esta questão foi desenvolvida uma investigação teórica através da definição dos principais conteúdos inerentes a ela, um estudo baseado em referenciais teóricos sólidos e relevantes nas áreas envolvidas. Foram apresentados conceitos, tecnologias e exemplos de projetos existentes relacionados com a temática. Foi também desenvolvida uma investigação prática, que permitiu executar as orientações teóricas mencionadas, mas também experimentar e testar as novas tecnologias.

Esta investigação teve como objetivo estudar as tecnologias disponíveis para a realização de um ambiente virtual 3D multiutilizador baseado no paradigma de interação natural. Para além de explorar o *hardware* e *software* existente, foi desenvolvido um protótipo para os testar.

O protótipo desenvolvido permitiu perceber as limitações tecnológicas, mas também as suas potencialidades no âmbito de uma aplicação multimédia 3D controlada por gestos. As **limitações**, tendo em conta o desenvolvimento prático do protótipo são as seguintes:

- Necessidade de normalizar as tecnologias disponíveis para o desenvolvimento de ambientes 3D com interação gestual.
- Simplificar o processo de instalação e configuração do *software* necessário para a criação de um ambiente 3D colaborativo.
- A falta de documentação específica relacionada com interação e comunicação em rede em ambientes multiutilizador 3D.

As **potencialidades**, resultantes da investigação e teste do protótipo:

- Possibilidade de controlo do avatar de uma forma completamente desprovida de acessórios tecnológicos.
- Forma de promover a comunicação e interação *online* em espaços físicos distintos.
- O controlo de um avatar 3D com os gestos promove uma utilização divertida.

- Existem diferentes contextos possíveis para a utilização de uma aplicação com estas características, nomeadamente atividades de expressão corporal (aulas de dança, formações onde a presença do corpo é importante, contextos associados à saúde e reabilitação física).

Para a construção do protótipo foi reunido um conjunto de tecnologias, que combinadas permitem a construção parcial de um ambiente virtual 3D controlado por gestos. De uma forma resumida, são necessárias duas grandes componentes: um sistema de reconhecimento e deteção de movimento do utilizador quando interage com a aplicação, que essencialmente é o *input* do sistema. Outra grande componente é o ambiente onde decorre a ação e que é visível para o utilizador, o ambiente virtual 3D, onde se encontra o avatar, que na sua essência corresponde ao *output* da aplicação. A combinação destes componentes permitiu desenvolver determinadas funcionalidades do protótipo. Para este conjunto de tecnologias poder funcionar na sua plenitude tem de ser alvo de modificações e adaptações, especialmente na vertente multiutilizador.

As tecnologias disponíveis para o desenvolvimento de um protótipo desta natureza, especialmente o *software*, apresentam alguma instabilidade. Por exemplo o *OpenNI* está em desenvolvimento e frequentemente existem novos *updates* disponíveis para *download*, com modificações e novas funcionalidades. Estas novas atualizações podem trazer incompatibilidades com outras versões de *software* (especificamente o *Unity3D*) e vice-versa. No caso de haverem novas alterações do *Unity3D*, isso pode afetar a compatibilidade com o *software* de reconhecimento e deteção das imagens do sensor *Kinect*.

Quando o desenvolvimento deste protótipo se iniciou, o sensor *Kinect* para *Windows* apenas estava anunciado, não se encontrando disponível para compra, o que fez com que os testes fossem efectuados com o sensor *Kinect* para a *Xbox*.

Com o desenvolvimento prático do projeto, foi possível constatar que existem diversas limitações técnicas em relação às tecnologias adoptadas, especificamente no que diz respeito às *framework* de deteção e reconhecimento do movimento. A primeira *framework* adotada (*ZigFu*) não permitiu desenvolver num mesmo projeto a componente de controlo do modelo 3D com a componente de rede. Este facto deve-se, essencialmente, à estrutura adotada pela *framework* dentro do *Unity3D*. Possivelmente, a utilização da *framework* com os módulos de rede do *Unity3D* não foi equacionada pelos criadores. A segunda *framework* adotada, apesar de mais estruturada e de permitir conjugar a componente de interação natural com as funcionalidades de comunicação em rede no *Unity3D*, não tem uma base que permita utilizar mais do que um sensor no mesmo projeto. Assim, a interação através da rede com mais do que um utilizador, e consequentemente mais do que um *Kinect*, não se verificou possível. Os testes efetuados com mais do que um *Kinect* demonstraram que a *framework* não consegue individualizar os sensores, ou seja,

independentemente do utilizador, o sensor é sempre o mesmo. Este facto impediu que mais do que um utilizador pudesse controlar o seu avatar através dos gestos.

Vários testes foram feitos na tentativa de forçar a *framework* e ultrapassar este problema, mas a estrutura da *framework* não permitiu tais modificações. A possibilidade de saber o ID de um *Kinect* não está prevista, pelo que não foi possível atribuir outros sensores a utilizadores distintos.

Este projeto de investigação propôs um conceito inovador e pouco explorado. As potencialidades de uma aplicação com estas características são variadas, desde áreas artísticas a áreas mais técnicas, como simulação e treino militar.

Os objetivos principais propostos para esta investigação foram conseguidos. Foram definidos e analisados os conceitos de Interação Humano-Computador, interação e interação gestual. Foram identificadas as principais características associadas a uma aplicação controlada por gestos. Foi referida de que forma a interação natural influencia a experiência do utilizador (UX) de uma aplicação multimédia 3D. Foi feito um levantamento bibliográfico dos principais problemas de usabilidade associados à interação controlada por gestos. Do ponto de vista das tecnologias necessárias para o desenvolvimento de uma aplicação 3D controlada por gestos, foram identificadas as principais componentes (*hardware* e *software*) necessárias para a criação da aplicação. No que diz respeito ao *hardware*, foram igualmente referidos os sensores e as suas principais características, as tecnologias de reconhecimento e deteção do movimento e os ambientes 3D.

De um ponto de vista prático, foram apontados e listados os requisitos funcionais de um protótipo que permita a interação gestual com um avatar num ambiente 3D distribuído. Foi também criada a componente de comunicação em rede (criação de um servidor). Implementou-se ainda a interação utilizador-avatar através dos gestos.

Em suma, esta dissertação explorou algumas das tecnologias multimédia interativas disponíveis que permitem uma interação gestual em ambientes 3D distribuídos. Foram identificadas as principais tecnologias necessárias para a criação de uma aplicação deste tipo, assim como as limitações e potencialidades associadas aos novos paradigmas de interação em ambientes 3D. Os resultados do desenvolvimento deste protótipo demonstram que existem ainda muitas limitações técnicas que impedem um pleno desenvolvimento de funcionalidades multiutilizador e que revelam a necessidade de maturação e simplificação destas tecnologias. As potencialidades estão associadas às vantagens que a interação natural pode oferecer ao utilizador, ao proporcionar uma forma de interagir com o sistema de uma forma mais natural e intuitiva. Mas, para tal, é necessário ter em conta um conjunto de sugestões (identificadas no referencial teórico) para implementar uma aplicação controlada por gestos.

O desenvolvimento deste projeto foi um desafio. Um desafio pelas tecnologias adotadas e o fator novidade, um caminho de aprendizagem durante toda a investigação.

### **5.1 Limitações do estudo**

No decorrer do desenvolvimento desta investigação houve alguns elementos, essencialmente ao nível do *software* e *hardware*, com versões instáveis e/ou em constante desenvolvimento e modificação. Desde o sensor *Microsoft Kinect* aos *drivers* e ambiente de desenvolvimento, tudo isto cria um pacote interdependente, ou seja, uma atualização ou modificação num dos elementos influencia o funcionamento dos restantes elementos. Este foi um dos aspetos que, em determinadas alturas, afetou o desenvolvimento do protótipo. Algumas atualizações de *software* não foram exclusivamente prejudiciais, mas trouxeram também benefícios ao projeto (como a simplificação do processo de calibração do utilizador).

A área das interfaces controladas por gestos, especificamente no controlo de um avatar num ambiente 3D com integração do sensor *Kinect* para a *Xbox*, é uma área de investigação recente. A documentação por vezes é frágil e muito limitada, o que dificultou o desenvolvimento. Em muitas ocasiões, suporte foi adquirido junto de fóruns e grupos *online*, mas nem sempre se revelou suficiente para as necessidades deste projeto. O facto de envolver um ambiente virtual 3D em rede, torna este projeto ainda mais particular, o que se revelou nos pedidos de ajuda para problemas específicos a esta peculiaridade, em que a investigadora não obteve resposta.

As *frameworks* utilizadas para a comunicação com o sensor foram desenvolvidas em *C#*, uma linguagem de programação que a investigadora não dominava. Este aspeto revelou-se muito importante aquando da necessidade de alteração de parâmetros diretamente na *framework*. Esta dificuldade impediu o desenvolvimento de alguns aspetos do protótipo, como a personalização do “cursor”, ou o desenvolvimento de outras funcionalidades.

O facto de o protótipo não se encontrar com as funcionalidades pretendidas completamente implementadas dificultou o teste junto dos utilizadores, e assim criou um obstáculo à obtenção de *feedback* relativo às vantagens e desvantagens da utilização de uma aplicação deste género.

No início do desenvolvimento do projeto, esteve previsto o envolvimento de uma pessoa com conhecimentos e experiência prévia em *Unity3D* e integração com o *Kinect*. A saída inesperada da pessoa com esse cargo influenciou também o desenvolvimento inicial do protótipo, uma vez que, desaparecendo o apoio dessa parte, a investigadora acabou por ter de dedicar mais tempo à adaptação ao *software* do que seria expectável.

## 5.2 Proposta de desenvolvimentos futuros

No contexto da interação controlada por gestos em ambientes colaborativos 3D, seria interessante ter uma melhoria das *frameworks* existentes, na perspectiva da integração de opções de comunicação em rede. Estas funcionalidades poderiam permitir o desenvolvimento de uma aplicação multiutilizador e, conseqüentemente, a realização de testes com vários utilizadores. Estes facilitariam a exploração de vantagens e restrições de uma aplicação desta natureza, num contexto de utilização específico. A realização de testes de usabilidade ajudaria a apontar falhas mas também potencialidades na utilização da interação gestual em ambientes 3D colaborativos.

Existem contextos de utilização possíveis de uma aplicação com estas características, cenários onde a expressão corporal e atividades físicas sejam executadas, como por exemplo o contexto da dança em grupo. Este é um contexto de possível utilização de uma aplicação com estas características pois permite uma interação gestual sem necessidade do utilizador segurar sensores ou comandos. Tornar-se-ia uma forma mais interessante e intuitiva de controlar o avatar através da expressão corporal e com a prática de exercício simultaneamente. Existem outros estudos na área da interação humano-computador que conjugam a dança e a interação através dos gestos (Alaoui, Caramiaux, & Serrano, 2011).

Em relação às funcionalidades cuja implementação numa aplicação deste tipo poderia ser interessante, é possível referir a inclusão de mais gestos e eventos. Opções de personalização do ambiente virtual para o utilizador, tal como a personalização do avatar, poderiam ser mais-valias importantes.



## Bibliografia

- Alaoui, S. F., Caramiaux, B., & Serrano, M. (2011). From dance to touch: movement qualities for interaction design. *Proceedings of the 2011*, 1465–1470. Acedido em 9/11/2011, de <http://portal.acm.org/citation.cfm?id=1979792>
- Alben, L. (1996). Quality of experience. *Magazine interactions*, 11–15. doi:10.1145/235008.235010
- Asus Xtion Pro Live. (2012). Acedido em 17/10/2012, de [http://www.asus.com/Multimedia/Motion\\_Sensor/Xtion\\_PRO\\_LIVE/](http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/)
- Bartle, R. (2004). *Designing Virtual Worlds*. (R. A. Reiser & J. V Dempsey, Eds.) *New Riders Publishing In* (Vol. p, p. 741). New Riders Games. Acedido de <http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&path=ASIN/0131018167>
- Bell, B. M. W. (2008). Virtual Worlds Research: Past, Present & Future. *World*, 1(1), 1–5.
- Besada, J., Bernardos, A., & Portillo, J. (2011). Low cost 3D Gesture based interface use for engineering lecturing. *Engineering Education*, 1–6.
- Beurden, M. Van, & IJsselsteijn, W. (2011). User experience of gesture-based interfaces: A comparison with traditional interaction methods on pragmatic and hedonic qualities. In *access.uoa.gr* (pp. 121–124). Athens, Greece. Acedido em 2/5/2012, de [http://access.uoa.gr/gw2011/proceedingsFiles/GW2011\\_32.pdf](http://access.uoa.gr/gw2011/proceedingsFiles/GW2011_32.pdf)
- Book, B. (2004). Moving beyond the game: social virtual worlds. *State of Play*, 2(October 2004), 6–8. Acedido em 4/1/2012, de [http://deby.net/FILES/3d/ARTICLES/moving\\_beyond\\_the\\_game - social\\_virtual\\_worlds.pdf](http://deby.net/FILES/3d/ARTICLES/moving_beyond_the_game_social_virtual_worlds.pdf)
- Borenstein, G. (2011). *Making Things See. Image (Rochester, N.Y.)*. O'Reilly Media, Inc, USA.
- Bowman, D., Kruijff, E., LaViola, J. J., & Poupyrev, I. (2005). *3D user interfaces. Theory and Practice*.
- Carmo, H., & Ferreira, M. M. (1998). *Metodologia da Investigação*.
- Carroll, J. M. (2011). Human-computer interaction. *interaction design*. Acedido em 10/12/2011, de [www.interaction-design.org/printerfriendly/encyclopedia/human\\_computer\\_interaction\\_hci.html](http://www.interaction-design.org/printerfriendly/encyclopedia/human_computer_interaction_hci.html) 1/10

- Developkinect -Interactive Puppet. (2011). Acedido em 5/1/2012, de <http://developkinect.com/news/interactive-display/interactive-puppet-display-using-openframeworks-and-unity-3d>
- Dix, A., Finlay, J., Abowd, G. D., & Beale, R. (2003). *Human computer interaction* (Third Edit., p. 834). Prentice Hall.
- Galitz, W. O. (2007). *The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques* (3rd ed., p. 888). Indianapolis, Indiana: Wiley Publishing, Inc.
- Giles, J. (2010). Inside the race to hack the Kinect. *The New Scientist*, 208(2789), 22–23. doi:10.1016/S0262-4079(10)62989-2
- Hardeman, N., & Licherdell, L. (2011). Follow-This. Acedido de <http://nickhardeman.com/460/follow-this/>
- Hassenzahl, M., & Tractinsky, N. (2006). User experience - a research agenda. *Behaviour & Information Technology*, 25(2), 91–97. doi:10.1080/01449290500330331
- Hewett, Baecker, Card, Carey, Gasen, Mantei, Perlman, Strong, V. (1996). Curricula for Human-Computer Interaction. Acedido em 5/1/2012, de [http://old.sigchi.org/cdg/cdg2.html#2\\_1](http://old.sigchi.org/cdg/cdg2.html#2_1)
- ISO. (1998). ISO 9241-11: Guidance on Usability. Acedido em 12/12/2012, de [http://www.usabilitynet.org/tools/r\\_international.htm#9241-11](http://www.usabilitynet.org/tools/r_international.htm#9241-11)
- Katz, N., Cook, T., & Smart, R. (2011). Extending Web Browsers with a Unity 3D-Based Virtual Worlds Viewer. *Ieee Internet Computing*.
- Keane, S., Hall, J., & Perry, P. (2012). *Meet the Kinect - An introduction to programming natural user interfaces* (p. 150). aPress.
- Kelway, J. (n.d.). Six Circles.
- Kinect interactive visuals. (2011). Acedido em 14/11/2011, de <http://www.kinecthacks.com/kinect-interactive-visuals/>
- Koster, R. (2004). A virtual world by any other name? Acedido em 2/1/2012, de [http://terranova.blogs.com/terra\\_nova/2004/06/a\\_virtual\\_world.html](http://terranova.blogs.com/terra_nova/2004/06/a_virtual_world.html)

- Krueger, M., & Gionfriddo, T. (1985). Videoplace - An artificial reality. *ACM SIGCHI Bulletin*, (April), 35–40. Acedido em 6/12/2011, de <http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract>
- Krueger, M. W. (1993). Environmental Technology: making the real world virtual. *Communications of the ACM*, 36(7), 36–37. Acedido em 6/12/2011, de <http://dl.acm.org/citation.cfm?id=159563>
- LaViola, J. J., & Keefe, D. F. (2011). 3D spatial interaction: applications for art, design, and science. In *ACM SIGGRAPH 2011 Courses* (p. 1). ACM. Acedido em 10/12/2011, de <http://dl.acm.org/citation.cfm?id=2037637>
- Liu, W. (2011). Natural User Interface- Next Mainstream Product User Interface. In *Computer-Aided Industrial Design & Conceptual Design (CAIDCD), 2010 IEEE 11th International Conference* (pp. 203–205). doi:10.1109/CAIDCD.2010.5681374
- Microsoft. (2011a). Microsoft Kinect. Acedido em 15/12/2011, de <http://www.xbox.com/en-US/kinect>
- Microsoft. (2011b). Programming guide - Getting Started with the Kinect for Windows SDK Beta from Microsoft Research. Acedido em 9/1/2012, de <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Programming+Guide#3>
- Microsoft. (2012). Kinect for Windows Human Interface Guidelines.
- Moggridge, B. (2007). *Designing Interactions [Hardcover]* (p. 766). The MIT Press; 1 edition. Acedido em 7/1/2012, de <http://www.amazon.com/Designing-Interactions-Bill-Moggridge/dp/0262134748/theartofinnovat>
- NI Mate. (2012). Acedido em 19/3/2012, de <http://www.ni-mate.com/>
- Nielsen, J. (2010). Kinect Gestural UI: First Impressions. Acedido em 22/12/2011, de <http://www.useit.com/alertbox/kinect-gesture-ux.html>
- Nielsen, M., Störring, M., Moeslund, T., & Granum, E. (2004). A procedure for developing intuitive and ergonomic gesture interfaces for HCI. *Gesture-Based Communication in Human-Computer Interaction*, 105–106. Acedido em 3/1/2012, de <http://www.springerlink.com/index/wmp0f5ned2mtf26.pdf>

- Oliveira, L. R. (2006). Metodologia do desenvolvimento : um estudo de criação de um ambiente de e- learning para o ensino presencial universitário Developmental methodology : a study about university teaching, *10*(April 2005), 69–77.
- Open NI User Guide. (n.d.).
- OpenKinect. (2011). Main Page - OpenKinect. Acedido em 9/1/2012, de [http://openkinect.org/wiki/Main\\_Page](http://openkinect.org/wiki/Main_Page)
- OpenNI. (n.d.). Acedido em 9/1/2012, de <http://www.openni.org/>
- OpenNI. (2012). NI Apps Guidelines.
- Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(7), 677–695. Acedido em 13/12/2011, de [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=598226](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=598226)
- Peter in Conversation with Don Norman About UX & Innovation. (2007). Acedido de <http://www.adaptivepath.com/ideas/e000862>
- Quivy, R., & Campenhoudt, L. (1995). Manual de investigação em ciências sociais e humanas. *GradivaPublicações*.
- Richey, R. C., & Klein, J. D. (2005). Developmental research methods: Creating knowledge from instructional design and development practice. *Journal of Computing in Higher Education*, *16*(2), 23–38. doi:10.1007/BF02961473
- Richey, R., Klein, J., & Nelson, W. (2004). Developmental research: Studies of instructional design and development. *Handbook of research for educational ....* Acedido em 26/11/2012, de <http://www.aect.org/edtech/ed1/41.pdf>
- Saffer, D. (2007). *Designing for interaction* (2nd Editio., p. 240). Berkeley, CA/US: New Riders Publishing.
- Saffer, D. (2009). *Designing Gestural Interfaces*. (M. Treseler, Ed.) (p. 247). O'Reilly Media, Inc, USA.

- Sanchez, J. (2009). A Social History of Virtual Worlds. *Library Technology Reports*, (figure 1).  
Acedido em 4/1/2012, de <http://www.mendeley.com/research/a-social-history-of-virtual-worlds/>
- Schreiber, M., Von Wilamowitz-Moellendorff, M., & Bruder, R. (2009). New interaction concepts by using the wii remote. *Human-Computer Interaction. Novel Interaction Methods and Techniques*, 261–270. Acedido em 29/12/2011, de <http://www.springerlink.com/index/BG3J773821357165.pdf>
- Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design: Beyond Human-Computer Interaction* (second edi., p. 800). John Wiley & Sons Ltd. Acedido em 9/1/2012, de <http://www.bookdepository.co.uk/Interaction-Design-Helen-Sharp/9780470018668>
- Sidik, M. K. B. M., Sunar, M. S. Bin, Ismail, I. Bin, Mokhtar, M. K. Bin, & Jusoh, N. B. M. (2011). A Study on Natural Interaction for Human Body Motion Using Depth Image Data. *2011 Workshop on Digital Media and Digital Content Management*, 2, 97–102. doi:10.1109/DMDCM.2011.26
- Snibbeinteractive. (2011). SocialStage. Acedido em 15/11/2011, de <http://www.snibbeinteractive.com/platforms/socialstage/products/socialstage>
- Sung, K. (2011). Console Technologies. *Computer*, (February), 91–93.
- Svedström, T. (2010). *Gesture interfaces. Interfaces*.
- Tidwell, J. (2011). *Designing Interfaces*. (M. Treseler, Ed.) (Second Edi., p. 576). O'Reilly Media, Inc, USA. Acedido de <http://designinginterfaces.com/>
- Unity3D. (2012). Unity3D Network. Acedido em 10/8/2012, de <http://docs.unity3d.com/Documentation/Components/net-HighLevelOverview.html>
- Usability Body of Knowledge - Glossary. (n.d.). Acedido em 26/4/2012, de <http://www.usabilitybok.org/glossary>
- Valli, A. (2008). The design of natural interaction. *Multimedia Tools and Applications*, 38(3), 295–305. doi:10.1007/s11042-007-0190-z
- Vera, L., Gimeno, J., Coma, I., & Fernández, M. (2011). Augmented Mirror: Interactive Augmented Reality System Based on Kinect. In *Ifip International Federation For Information Processing* (pp. 483–486).

- Villaroman, N., Rowe, D., Ph, D., & Swan, B. (2011). Teaching Natural User Interaction Using OpenNI and the Microsoft Kinect Sensor.
- Von Hardenberg, C., & Bérard, F. (2001). Bare-hand human-computer interaction. In *Proceedings of the 2001 workshop on Perceptive user interfaces* (pp. 1–8). ACM. Acedido em 3/1/2012, de <http://dl.acm.org/citation.cfm?id=971513>
- Wellner, P. (1991). The DigitalDesk calculator: tangible manipulation on a desk top display. *Proceedings of the 4th annual ACM symposium on ...*, 27–33. Acedido em 15/12/2012, de <http://dl.acm.org/citation.cfm?id=120785>
- Wi-GO. (2012). Acedido em 19/11/2012, de <http://www.is2you.eu/PT/index.html>
- wigdor, D., & Wixon, D. (2011). *Brave NUI world: designing natural user interfaces for touch and gesture*. *Annals of Physics* (p. 264). Morgan Kaufmann Publishers In.
- Wilson, A. (2008). Sensor and recognition based input for interaction. In A. Sears & J. A. Jacko (Eds.), *The Human Computer Interaction Handbook* (Second edi., pp. 177–199). Lawrence Erlbaum Associates. Acedido em 7/1/2012, de <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Sensor+and+recognition+based+input+for+interaction#1>

## Anexos

# Anexo1 - Principais componentes do protótipo

