Diogo Wahnon Silva Teixeira de Brito

Nº 37739

A Framework for Developing Motion-Based Games

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientadora : Teresa Romão, Prof. Auxiliar,
Universidade Nova de Lisboa

Júri:

Presidente:	Prof. Doutor João Manuel Santos Lourenço
Arguente:	Prof. Doutor Carlos Alberto Pacheco dos Anjos Duarte
Vogal:	Prof. Doutora Teresa Isabel Lopes Romão

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

September, 2013

A Framework for Developing Motion-Based Games

To Laica.

Acknowledgements

First of all, I want to express my gratitude to all of those who have helped me achieve my masters degree. To Faculdade de Ciências e Tecnologias, for allowing me to study in this area. To my supervisor Professor Teresa Romão for all the care, ideas and constructive criticism during the work in my thesis. To Pedro Centieiro, who was also preciously helpful, providing a lot of support and assistance during the development of this work. To Ahmad Akl, for the availability and help during this work. Moreover, I'd like to thank André Gil and Bliss Applications, for all the flexibility that was given me for completing my degree. Also, for all the technical support that I had from my dear colleagues.

Those who helped me during the tests in this work. Fábio Bernardo, for his indescribable help. Sara, for helping me out throughout this work and Sérgio, for his three-dimensional viewing capabilities. My friends, for walking with me along this path, while writing this thesis.

Last, but definitely not least, my family, for always given me everything I needed. My father, who has shown me the correct paths and given me his full support throughout my life. My mother and brother, for all their strength and care.

Abstract

Nowadays, whenever one intents to develop an application that allows interaction through the use of more or less complex gestures, it is necessary to go through a long process. In this process, the gesture recognition system may not obtain high accuracy results, particularly among different users.

Since the total number of applications for mobile systems, like Android and iOS, is close to a million and a half and is still increasing, it appears essential the development of a platform that abstracts developers from all the low-level gesture gathering and that streamlines the process of developing applications that make use of this kind of interaction, in a standardize way. In this case such was developed for the iOS system.

At the present time, given the existing environment issues, it is ideal to attract the attention, motivate and influence the greatest number of people into having more pro-environmental behaviors. Thus, as a proof of concept for the developed framework, an educational game was created, using persuasive technology, to influence players's behaviors and attitudes in a pro-environmental way.

Therefore, having this idea as a basis, it was also developed a game that is presented in a public ambient display and can be played by any participant close to the display who has a device with iOS mobile system.

Keywords: gesture recognition, accelerometer, public ambient displays, dynamic-time-warping, games, persuasive technology

Resumo

Actualmente, sempre que se pretende desenvolver uma aplicação que permita interacção através do uso de gestos mais ou menos complexos, é necessário passar por um longo processo. Neste, pode acabar por não se obter um reconhecimento de gestos com bons índices de precisão, particularmente entre utilizadores distintos.

Uma vez que o número total de aplicações para sistemas móveis como Android e iOS está perto do milhão e meio e continua a aumentar, revela-se fundamental o desenvolvimento de uma plataforma que abstraia os programadores de todo o funcionamento de baixo nível de captação de gestos e que agilize o processo de desenvolvimento de aplicações que recorram a este tipo de interacção, de uma forma uniformizada. Neste caso tal foi desenvolvida para o sistema iOS.

Hoje em dia, dados os problemas actualmente existentes a nível ambiental, é ideal chamar a atenção, adquirir o interesse e influenciar o maior número de pessoas a ter comportamentos mais pró-ambientais. Assim, e como prova de conceito da plataforma desenvolvida, um jogo educacional, que faz uso da tecnologia persuasiva, foi criado, de modo a influenciar as atitudes e comportamentos dos seus jogadores de uma forma pró-ambiental.

Assim, com esta ideia como base, foi também desenvolvido um jogo que é apresentado num expositor público e que pode jogado por qualquer participante perto da zona do expositor que tenha um dispositivo com o tipo de sistema móvel iOS.

Palavras-chave: reconhecimento gestos, acelerómetro, expositores públicos, dynamic-time-warping, jogos, tecnologia persuasiva

Contents

List of Figures

List of Tables

1

Introduction

This document exposes the research and work that was made in similar and relevant areas in terms of gesture recognition, environmental issues and the use of persuasive computing. In this thesis a framework for recognizing motion based gestures was built, as was a proof of concept for the framework. Both are also addressed in this document.

This thesis was conducted in the scope of the DEAP project (Developing Environmental Awareness with Persuasive Systems), which includes members of New University of Lisbon and the University of Évora. DEAP project is funded by FCT (Fundação para a Ciência e Tecnologia).

1.1 Motivation

In the past years people's use of mobile devices has spread all around the world. The main contribution for such event is that the size of its components became smaller, their connectivity capacities have been improved, their computing performance is higher and their price is affordable by a large part of the western world.

Besides, there have been major contributions in the area of touch screen mobile devices, promoting interfaces with better usability that allow users to interact with the devices in a touch-based way thus loosing the need of having hardware keyboards, which allow the display of much more information. Furthermore, components such as accelerometers have been used in order to enable the use of gestures and also to

increase the level of context-awareness of the device. A simple example in the smartphones environment is the switching from portrait to landscape mode depending on the rotation of the device, which is discoverable by its acceleration.

Having this background, companies that engineer and manufacture this kind of equipments developed their respective operative systems (OSs) in order to allow developers world-wide to build customized applications. The two most known operative systems for mobile devices that were designed with this in mind were iOS and Android, by Apple Inc. and Google, respectively.

Besides, smartphones computation power and storage capacity has been quickly growing bigger thus allowing software developers to build better and more capable applications.

After developing an application for any of these operative systems, it is possible to promote, distribute or sell it, by using their software's applications market, Apple® App Store for iOS and Google Play for Android. In the first, there is over than 900.000 applications [Cos13], while in the latter there is over a million of applications [Row13], which shows that there is in fact a large number of deployed applications for such devices.

Therefore, it is easily visible that mobile devices, particularly smartphones, have been widely used in the recent years, with millions of application downloads, showing that it is possible for enthusiastic developers to have a new idea and build a custom application that can be freely distributed in its respective store. Furthermore, it is simple for developers to access the data captured by multiple sensors that these devices contain, such as accelerometer, gyroscope or wireless.

Moreover, the framework was developed for smartphones because this kind of devices don't require an external setup for gesture recognition, as do other existing devices, such as Nintendo's Wii or Microsoft's Kinect. In this thesis the system that was chosen to target the framework was iOS. Such choice was made, not particularly for any technological matter but because this work is done in the scope of the DEAP project and applications in the same scope were built for iOS. Thus, future applications also funded by DEAP will probably be built for the iOS, increasing the probability of the framework to be used and helpful in such cases.

Regarding the data captured by the accelerometer sensor in a smartphone context, this represents the device acceleration on its multiple, usually three (x-y-z), axis. Such can be seen in Fig. 1.1. For a mobile application developer, making use of this raw-data can be done trivially for simple tasks: it has been made in the past by some applications to respond to shaking or to the rotation of the device. While the first is used for instance as an undo command the last is, by far, the most employed use of the accelerometer, as

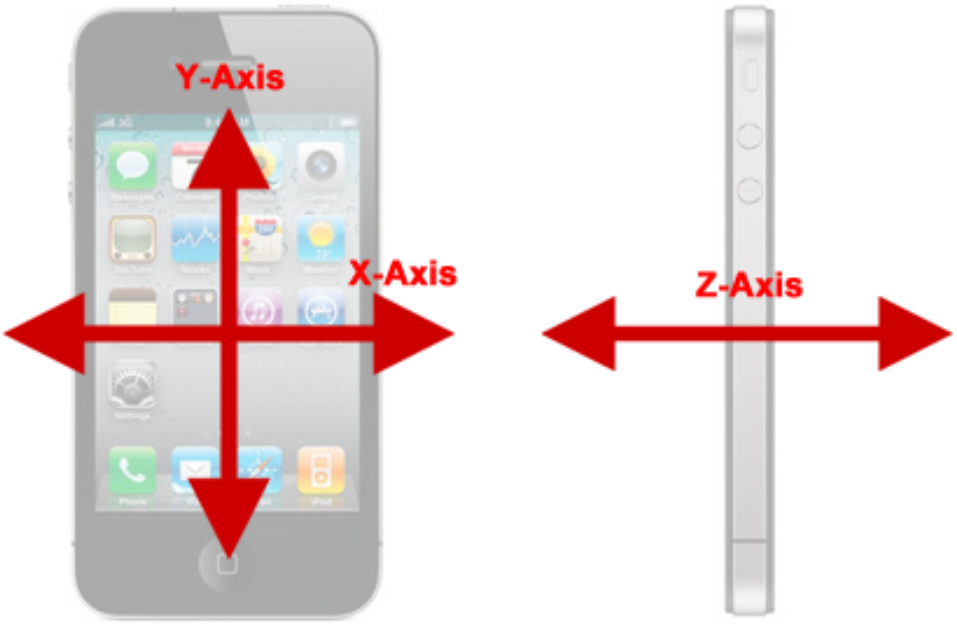


Figure 1.1: iPhone with coordinate system.

it is used for switching the device's display from portrait to landscape mode.

However, when it comes to developing applications that make a more complex use of this data, like gesture recognition, the task is far more complex. If a developer intends to build an application that makes use of gestures, he must be responsible for all the related work regarding its recognition. This, even tough it may seem a trivial question can imply multiple days of work.

Nowadays, if a developer builds a gesture recognition system from scratch there are multiple phases that must be performed: define the gestures that shall be recognized; for each one of these, various actors (participants, developers) must perform the gesture multiple times in order to calibrate the system. In the end, after a sluggish process where multiple actors, not only the developers, have spent their time, the results may not be as accurate as one would wish. Shortly speaking, it can be a painful work not only for developers but also for participants that help building the recognition system – all without guaranteed results. Thus, the main goal of this thesis is the development of a framework that allows developers to build motion-based applications without much effort nor low-level accelerometer knowledge and with a good recognition accuracy. Considering that this thesis was developed within the scope of the DEAP project, it also

has the role of making use of persuasive systems to alert users of the environmental issues that exist nowadays. Moreover, regarding the environmental scope of this work, humans are putting a huge effort upon Earth's natural resources, by the means of various sectors such like energy, agriculture, fishery and transportation. These actions are making climate changes that can already be measured like the world's temperature increase and the loss of the ozone layer. Therefore, if no action is taken in order to change this situation, irreversible damages can be caused to the environment within decades.

One way of struggling against this demand is by making world citizens aware of the problems and the primitive causes that contribute to the excessive consumption and waste of natural resources. Hence, it is possible to use persuasive technology in order to change users' behaviors towards the environment. Therefore, in this work, as a proof of concept for the framework, was developed a motion-based game that makes use of persuasive technology in order to warn and change participants behaviors towards what most contributes to world energy and resources consumption, leading to pollution and climate change.

1.2 Description and Context

Each and everyday more applications are published into the Apple® App Store, being developed by different thousand of companies. Recently, the 50 billionth application download was made from Apple® App Store [Pre13], which reveals the extreme demand for such kind of applications.

Nonetheless, it is not straightforward for a software developer to build a game or any other application that makes use of motion based gestures for the iOS system. It is a sluggish process which can take much time from the developer and other players enrolled in the creation of the application. Furthermore, in such cases, the result may not turn out as expected, having a low recognition accuracy or a high delay in the recognition process.

To elaborate the framework sketch, some related works in the gesture recognition area have been essential. It was possible to evaluate which are the existing technological possibilities to approach the issue of recognizing a movement and how some of them may be combined in order to obtain the best recognition accuracy levels.

Moreover, regarding both the fact that the framework must be tested and that this thesis is done under the DEAP project, a proof of concept is also required.

People are continuously subject to actions in which, even without their knowledge, are endangering the planet. Such can be caused by simple actions like wasting energy whilst using energy-expensive light bulbs or by not recycling their waste. It is

important to increase the awareness of people to these kind of situations, which can be achieved by showing them which actions are correct or not. Such can be reinforced by teaching the users in a amusing and funny way, as it is a game. Furthermore, if this game is played in a multiplayer environment based on scores, the user will be even more focused in practicing what the game states is correct, given the fact that results can be compared between players and friends.

1.3 Contributions

The main contributions of this thesis are:

- A framework for motion-based games built for the iOS system, that enables and facilitates the development of applications that make use of gestures;
- A game that informs users about the environmental issues that exist nowadays and stimulates them into having more pro-environmental behaviors;
- Insights regarding the use and development of gesture-based games and the interaction with public ambient displays and mobile devices;

1.4 Document Organization

The remainder of this document is organized in the following maner:

- Chapter 2 addresses the relevant work that was made in similar areas as this thesis, such as gesture recognition and persuasive technology in a gaming context.
- Chapter 3 describes the gesture recognition system, from the original system theory, to the final architecture implementation.
- Chapter 4 covers the proof of concept for the framework, as well as the user tests that were conducted to evaluate the framework and the game.
- Chapter 5 is the last, in which the conclusions about the work that was done are made. Moreover, the future work is also addressed in this section.

2

Related Work

In order to fully comprehend the content presented in this document, some relevant related areas must be studied. This chapter, besides providing a summary of the state of the art in these areas, intends to make a brief description of them.

In Section 2.1 the concepts of pervasive computing and context awareness are covered. It is explained how the development of wireless technologies and new types of embedded devices can contribute to people daily tasks. It is also explained how context can be important to these kind of applications, how it can be retrieved by them and how it can affect them.

In the next section, 2.2, gesture-controlled user interfaces, it is shown how computer interaction was enriched due to technological developments in the last decades. New forms of interaction and some practical examples are given.

Section 2.3, addresses some studies that were made in the gestures for mobile interaction and how these can be useful and a natural communication channel.

Following, in Section 2.4, the work related to the core of this thesis, different kind of possible ways of recognizing gestures are approached. A focus is made on the movement-based category, once it doesn't require an external setup and can be used in an ubiquitous environment, such as the smartphones environment.

Besides specifying the available technologies helpful in this area, a brief explanation of some related algorithms and techniques for both improving recognition accuracy and

reducing computational costs is presented.

As the movements recognized by the proposed framework must be submitted to a proof of concept, a multiplayer game which promotes interaction between mobile devices and public ambient displays was developed. Since this work was developed in the scope of project DEAP, the game focus some pro-environmental ideas. In this way, the following sections cover related topics in the areas of environmental awareness, interaction with public ambient displays and persuasive technology in a gaming context.

Section 2.5, environmental awareness, addresses the effort that is being put upon Earth's natural resources and biodiversity. Some of the key areas that most contribute to this are presented and have been taken into account when designing the proof of concept.

Interaction with public ambient displays, Section 2.6, presents multiple studies that use this kind of technology and how it can help new users to interact with the system.

Following, section 2.7, persuasive technology in gaming context, explains the basis of these technologies and how they can be employed in order to change users' behaviors. Moreover, some related works in the area, specially some pro-environmental awareness ones, are presented.

By addressing these multiple subjects the reader will know how the use of gestures in a pervasive computing can be useful; which kind of recognition technologies and other works in the gesture recognition area exist.

Finally, it is be explained how one can combine gesture recognition and user persuasion in order to achieve environmental awareness and pro-environmental user behaviors.

2.1 Pervasive Computing

Pervasive computing, also known as ubiquitous computing, has long been idealized and studied. In 1991, Weiser, who is said to be the father of ubiquitous computing, addressed a scenario where hardware and software would be used by people unconsciously to accomplish everyday tasks. This scenario would emerge somewhere over the next 20 years – quite where we are now – and is in fact not far from the truth [Wei91]. Regarding this sketch proposed by Weiser, offices or rooms, would be composed by a set of different devices from TAB's (consisting of all that can be found in nowadays smartphones, like iPhone) , PAD's (similar to today's tablets like iPad) and computers, all connected by radio frequency, like Wireless Local Area Network (WLAN). In fact, it is quite the picture that can now be found in any office or room of a developed country,

in the middle and high class society.

This kind of computing can be of much help and comfort to a person daily life once it has numerous uses, such as home temperature management, where a certain temperature is kept independently of the user actions or a refrigerator that is capable of knowing the validity of its contained food.

The development of wireless networks, together with the common use of new types of embedded and mobile devices are leading to the already current, but specially future, scenario in which cheap interconnected computing devices are capable of supporting users in a multiple set of tasks.

As said in [Sat01], a pervasive computing environment is one where there exists a truly integration between the users and the existing computing and communication capabilities, thus becoming a technology that is invisible to the user.

Applications in this kind of environment, for having the need of acting in highly dynamic environments and placing fewer demands on user's, have the need of being sensitive to the environment that surrounds them.

That made, unlike what happens with usual desktop applications, a need for context-awareness appears [HIR02]. Besides, it is also valuable to know what context, in a pervasive environment, really means once it is a term employed with a wide variety of meanings.

Moreover, pervasive computing sustains why this work is targeted to a smartphones environment, once it is easy for users to access it with no external setups or complications. Besides, the use of gestures is a natural interface for humans.

2.1.1 Context awareness

Context is a concept that has been studied for the past four decades by various areas of computer science and with multiple definitions, such as "that which surrounds, and gives meaning to something else" [SBG99].

Regarding the mobile and ubiquitous context, it is demanding the ability of discovering and reacting to changes in a given environment, providing the user with services that directly depend upon its surrounding environment.

There are two concepts that must be addressed while developing a context-aware application: contextual sensing and contextual adaptation. While the first refers to the detection of contextual information, the second refers to the capability of an application to adapt itself upon dynamic contextual information. Regarding the first point, it is possible to identify relevant information with the use of sensing components. These

components, that integrate any smartphone nowadays, can be, for instance, accelerometers or gyroscopes for motion sensing or GPS for location sensing. Then, with data from this sensors and with multiple techniques according to the type of data, context-aware behaviors can be taken.

According to [HIR02], there are some observations about the nature of contextual information in pervasive computing systems, namely: Contextual information can be static or dynamic, being the majority of information mainly dynamic. Also, given its dynamics, this information has a temporal validity. It is imperfect and it can be incorrect, inconsistent or incomplete. As the contextual information is derived from sensors, it has many alternative representations and, for the same reason, there is usually a significant gap between the data granularity at the sensor level and the one that the application is really interested in working with. Note that this point is particularly essential in this thesis, once one of its goals is to develop a framework that shall fill this gap between the low-level accelerometer data and the high-level gesture recognition.

Finally, it is also fundamental that programmers are abstracted from gathering this kind of information and so, infrastructures are needed to gather and manage this type of information to applications, in order to facilitate their development.

2.2 Gesture-controlled user interfaces

During the past decades, computer interaction has been mainly done through the use of traditional keyboard and mouse [BP09]. This way of interaction, even tough it is still widely used nowadays and, in some contexts without any other better options, can lack of usability in some other contexts. For instance, the mobile interaction is a good example where it is not suitable to work in this traditional way thus requiring other ways of interaction.

However, technological developments allowed the appearance of new ways of computer interaction that provide higher usability and with decreasing costs [BP09] [Pyl05].

It is natural for humans to use gestures as a way of communication. As young children, we learn to communicate through gestures even before being able to talk. Moreover, gestures can be used instead or in combination with verbal communication [BP09]. So, throughout our life, from youth to elderness, we don't use only gestures or verbal communication, but a combination of both in order to express ourselves in a more complete manner and, most important of all, we use it daily and without even noticing it.

This way, it is possible for humans to communicate with machines without requiring the use of mechanical devices like joysticks, keyboards or mouse. Objectively, humans can thus communicate with intelligent environments, like smart homes, by employing natural body movements or gestures.

This work, however, will be focused on a gesture control system in which a mobile hand-held device is used. By doing so, users do not need to be in a controlled environment that makes, for instance, the use of camera-based recognition but instead, to hold the device in their hands while describing a gesture.

Moreover, one must bare in mind that movements may vary from person to person, having different meanings depending on the user's culture. A good example is the way most western countries use head gesture for "yes" or "no". While, in these countries, people nod their head vertically for "yes" and horizontally for "no" this result is the exact opposite on cultures like Bulgarian and Serbo-Croatian, where the vertical nod means "no" and the horizontal is for "yes".

In terms of gesture-controlled interfaces there are multiple areas where these can be applied, such as entertainment, artificial intelligence, simulation, education and health.

In entertainment, there are examples being widely used nowadays like Nintendo Wii and Microsoft Kinect.

Nintendo Wii, released in November 2006, is a home video game console that, after one year, had already sold over 20 millions units world-wide [Lee08]. Its main feature is the use of the Wiimote, a remote control that combines a 3-axis accelerometer, a high-speed IR-camera, a vibration motor, a speaker, Bluetooth and buttons. It allows users to easily interact with gestures and pointing, being connected by the wireless Bluetooth. This remote, in 2008, had a suggested retail price of US$40, which is really cost-effective for all it contains and is capable of [Lee08].

Microsoft Kinect is a motion sensing input device that allows users to interact solely with gestures and spoken commands, without making use of keyboard, mice or other device [Hsu11].

Even though it was primarily built for the Xbox 360 home video game console and Windows PCs, it is capable of being used in multiple other applications. Only in 2011, 18 million Kinects' were sold, demonstrating how widely it has been used [Eis12]. This may also result from its relatively cheap price of US$149, which, compared to interactive whiteboards can be in the range of US$800 to US$2500 [Eis12].

Albeit both these products have an high gesture recognition accuracy and success, they require an extra setup. Thus, unlike smartphones which users carry every day, are not fit for more general contexts.

There are also applications in other areas that are based in systems like Kinect. An

example of this is presented in [CCH11]. It is a Kinect-based system for physical rehabilitation targeted to young adults with motor disabilities. It is used in a public setting and shows that the participants increased their motivation in physical rehabilitation. Based in the ability of Kinect to detect user's movements and positions, it uses this information to detect if a participant is performing the exercises correctly and works in two phases. In a first phase, the participants must execute the movement correctly and, when such is done, the system learns and saves that movement. In a second phase, the participants perform the movement that the system previously stored.

Besides, it includes an interactive interface with audio and video feedback which enhances the participant's interest and motivation, contributing to a better experience and thus a better physical rehabilitation. The participant's status are kept in the system which allow therapists to review them easily and can be useful for a more complete recovery. In addition, using this type of environment, participants don't need to be bothered with any attached user sensors that can be obtrusive in their rehabilitation.

Other examples are studied like the use of Kinect in Education Technology [Hsu11]. In this work, it is shown that the use of Kinect and this kind of interactive environments has a great potential of boosting students creativity and enhancing classroom interactions. This study shows that Kinect is quite versatile as a teaching tool in a classroom: teachers can interact with the contents via body movements, gestures or voice communication; Kinect works with multiple-users and so students in the room can interact with the system; a classroom in this context can provide a whole-class support, allowing group interaction or teacher-student one-to-one interaction. Lastly, as it collects in-depth, 3D, information, it can support different kinds of teaching activities, like dance and martial arts.

Other studies, non-related to the wide known Kinect and Wii, have also been made in this area, like [Vat12]. In this work, Vatavu states that gestures are fundamental in terms of computing for their potential for supporting natural, intuitive and fluent interaction, often supported by its user familiarity. Thus, in his study, a technique for reusing gesture commands for frequent ambient interaction is introduced. Vatavu proposes a concept to interact with remote displays in which a mix of mobile phone use and a remote gesture recognition exists. In this kind of system, the mobile phone contains the gesture set and the remote display is responsible for implementing and running the gesture recognizer. This way it is possible for a user to carry its own gestures on a device of his own, like a mobile phone, and upload them into any public ambient display that they will interact with. This solution takes advantage of strong points both in mobile devices as in gesture approaches. In the first case, it makes use of the device connectivity, that allows the connection to any ambient display and also its storage

capacity. Regarding the gesture approach, it promotes the user's preferred gestures, contributing to the system usability.

This can bring two important advantages to what happens today: users will already know how to interact with the system, because they have their own commands, and the accuracy of gesture recognition algorithms will considerably improve as the system can use user-specific training gestures for comparison, thus being user-dependent. The interconnection between the mobile phone and the remote display is possible with current mobile phones given their multiple connectivity options such as Bluetooth or WLAN. A scenario where such approach would be interesting, for instance, railway stations in different countries. A user who travels a lot through different stations and has the need of verifying in the system the next departing trains. However, an interaction is needed with the display and so the user can upload its known gestures, stored in his wireless capable mobile phone, and afterward interact with the system, that was unknown to him, with his own gestures.

The system is tested with the Microsoft Kinect. Similarly to what is done in this thesis application, the comparison between different gestures is makes use of DTW.

Regarding the translation between a gesture and a system function, its sketch can be seen in Fig. 2.1. Considering the commands, these are mainly divided in three sets: BASIC-CMDS, APP-CMDS and USER-CMDS. Regarding BASIC-CMDS, these are the most frequently used by any application like *start*, *next* or *close*. This is not an initially extensive set but one who contains basic commands that can be extended by any application. APP-CMDS is the set of commands of each application and USER-CMDS is the user specific set of commands. This last is initially a subset of BASIC-CMDS and is then extended by the user, as he interacts with different applications that use a larger set of gestures. Furthermore, to enable the user to reuse some gestures, thus not having the need of creating new ones depending on specific applications, a fourth set, AUGM-APP-CMDS acts as a synonym of gestures for APP-CMDS.

When the user uploads its gesture set to the application a matching algorithm that makes use of this infrastructure is used. In this phase the association between the USER-CMDS and AUGM-APP-CMDS is done and a possibility of results exist: perfect matching, perfect synonym-aided matching, partial sufficient matching and partial insufficient matching.

Even though in this document some camera-based recognition systems have been addressed, a motion-based recognition will be developed in this thesis, as aforementioned.

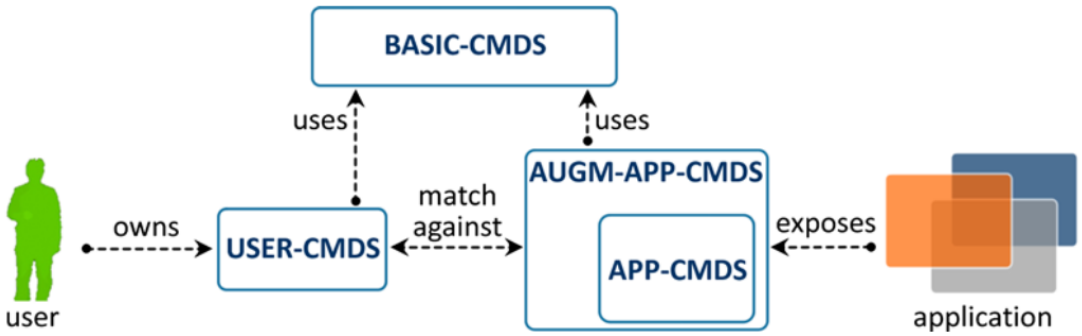


Figure 2.1: Commands matching consists in finding the best association between the commands exposed by the application (APP-CMDS) and the commands previously defined by the user (USER-CMDS). Using command synonyms (AUGM-APP-CMDS) helps this process. Taken from [Vat12].

2.3 Gestures for mobile interaction

Motion gestures in a mobile interaction environment have been studied on different manners in the past, but primarily evolving guessability studies, as in [RLL11] or [LWS+12].

Such studies are useful since smartphones combine several tasks into one component limiting their input and output, having small screens and embedded thumb-sized screen keyboards. Thus, using gestures to interact with smartphones may be seen as a more natural communication channel [Accelerometer-based gesture control for a design environment], providing additional input capabilities in a more intuitive way, that can be learned by observing others [LWS+12].

In terms of use in mobile interaction, studies in the area conclude that users are quite receptive towards using motion gestures to perform tasks. As shown in [RLL11] 82% of the participants said that they would use motion gestures at least occasionally, as only 4% would never use them at all.

In order to create the proposed framework, a dictionary of gestures based on previous studies was used. These was based on works like [AV10], [AFV11] and [KKM+05]. The main reason why a dictionary of known gestures was used is to be able to focus on gestures that users have shown some preference in the past and also to get the best recognition accuracy results out of these, instead of going through an infinite possibility of other gestures that may not reveal useful at all for the general users.

However, since it is a goal that our proposed framework enables developers to build applications that make use of gesture recognition, without the knowledge of gathering accelerometer data and processing it in order to achieve an high level of accuracy and

repeatability, it is also an objective of our work to let programmers add new gestures to be recognized.

2.4 Gesture Recognition

One of the most critical factors in the gesture based interaction is its recognition. In order to provide a good user experience and interaction, specially in a game based environment, it is important that the recognition is done almost instantaneously, unperceived by the user. Besides, it must have a high accuracy.

Regarding gesture recognition, previous works can be classified in two major categories: camera-based and movement-based.

In this document some examples were already presented for both cases. Two particularly interesting examples exist in both areas. For camera-based there is Microsoft Kinect and for movement-based the Wii Remote.

While camera-based can only be used where a fixed camera is set and established in a controlled environment the same is not true for dynamic environments, as uncontrolled public zones, for instance. In this way, movement-based recognition can be used as a better approach.

Regarding gesture recognition technology, several options can be used to capture the human gestures namely infrared beams, data glove, video recognition, accelerometers and gyroscopes. Moreover, it is also possible to combine multiple technologies as it has happened in the past [BP09].

In this work, the main goal is the accelerometer-based recognition, once a sensor for reading such values is widely available in the smartphones universe. Regarding this kind of recognition, several approaches have been tried in the past and, in recent cases, providing levels of accuracy near 100%. Albeit the goal of the developed framework in this work is not to reach an accuracy superior to the already existent (although our results reveal similar accuracy rates), it provides an implementation that promotes the abstraction to general application developers of programming such kind of gesture recognition.

Movements, even from the same person and very similar, are never exactly equal, having temporal variations and position deviations that can generate noisy data. Therefore, there must be a manner of matching different movements and, in this way, different techniques exist. The two most used techniques in gesture recognition are Hidden Markov Models (HMMs), a statistically machine learning method and Dynamic Time Warping, an algorithm that matches similarities between sequences of different durations. However, both algorithms have been widely used in other areas, such as speech

recognition.

These methods alone do not suffice in order to achieve an accuracy of 100%. As the accuracy is quite relevant in the gesture recognition, a more complex system, as proposed in [AFV11], is used in this thesis. A general overview of the system architecture is shown in Fig. 3.3. It requires the use of multiple algorithms, such as Dynamic Time Warping, Affinity Propagation, Random Projection and Compressive Sensing. A brief explanation of these are presented further in this section.

Akl's system, compared to others like uWave and Continuous or Discrete HMMs systems, achieves a superior performance, never going lower than 90% average recognition accuracy, in an interval of 8-18 gestures. These results take into consideration both user-dependent and independent gestures, as well as mixed-user ones. In terms of definitions, user-dependent recognition refers to when a system is trained with data from a subject and it is also used by this one. User-independent occurs when the recognition is made independently of the user while mixed-user recognition exists when multiple subjects train and use the recognition system.

It is noteworthy the fact that the work developed by Akl et al. [AFV11], given its context, is targeted to a user-independent gesture recognition where it is much harder to obtain high accuracy values. In this matter, while uWave is targeted to user-dependent gesture recognition achieving an accuracy of 75.4%, the latter system has higher values, comprehended in 94.60-96.84%.

2.4.1 Dynamic Time Warping

Dynamic Time Warping (DTW) [SC07] [KR05] is an algorithm that is able to find the optimal alignment between two time series and has been widely used in areas such as science, medicine, industry and finance, in subjects like data mining and gesture recognition. In terms of complexity, it has a quadratic time and space complexity, which limits its use to small time series data sets.

DTW uses a technique of stretching or shrinking a signal along its time axis in order to find a region of similarities between both signals thus finding the minimum-distance warp path, by using a dynamic programming approach. Thus, it serves as a good way to compute the degree of similarity between any two time series or sequences.

This is quite useful, particularly when compared to the Euclidean Distance, which is also a widely use technique to represent the degree of similarity between two sequences $\boldsymbol{C} = \{c_1, \ldots, c_n\}$ and $\boldsymbol{Q} = \{q_1, \ldots, q_n\}$. In the Euclidean Distance, the cost is computed

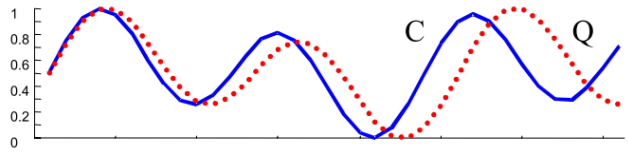


Figure 2.2: Two time sequences C and Q which are similar, but out of phase. Taken from [KR05].

as following:

$$d_{\text{Euclid}}(C,Q) = \sqrt{\sum_{i=1}^{n}(c_i - q_i)^2} \tag{2.1}$$

However, the latter has the limitation that both sequences must be of the same length, which makes it inapplicable in all the cases where two signals don't share the same length.

Assume that we have two sequences

$$\begin{aligned} \boldsymbol{C} &= \{c_1, \ldots, c_n\} \\ \boldsymbol{Q} &= \{q_1, \ldots, q_m\} \end{aligned} \tag{2.2}$$

and we wish to align both sequences using DTW(C,Q). In order to accomplish such thing, we build a $n \times m$ matrix $\mathbf{M}$ where the $(i^{\text{th}}, j^{\text{th}})$ element of $\mathbf{M}$ contains the distance $d(c_i, q_j)$ between the two points c_i and q_i and $d(c_i, q_j) = (c_i - q_j)^2$. Each element (i,j) corresponds to the alignment between the points c_i and q_j

A warping path W is a contiguous set of matrix elements that defines a mapping between C and Q. The k^{th} element of W is defined as $w_k = (i,j)_k$. So, W is defined as

$$W = \{w_1, w_2, \ldots, w_k, \ldots, w_K\} \tag{2.3}$$

where

$$\max(m,n) \leq K < m + n - 1 \tag{2.4}$$

The warping path is usually subject to multiple constraints [KR05]:

- Boundary conditions: $w_1 = (1,1)$ and $w_K = (m,n)$ and this needs the warping path to start and finish in diagonally opposite corner cells of the matrix;
- Continuity: Given $w_k = (a,b)$ then $w_{k\cdot 1} = (a',b')$ in which $a - a' \leq 1$ and $b - b' \leq 1$. This circumscribes the allowable steps in the warping path to adjacent cells,

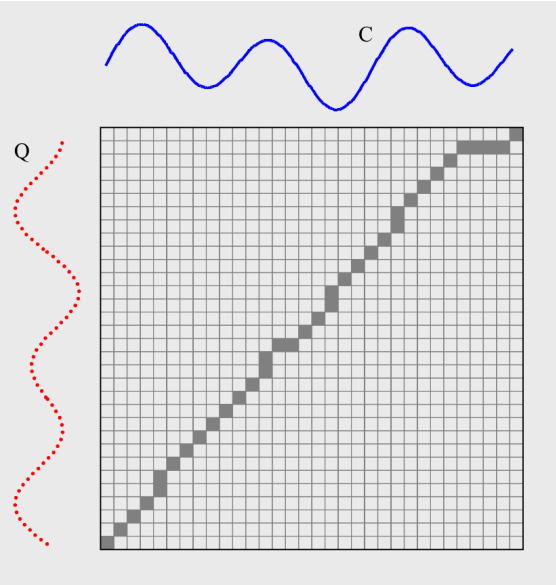


Figure 2.3: Alignment of the two sequences by finding the optimal warping path, that is shown in the solid squares. Taken from [KR05].

including diagonally adjacent cells.

- Monotonicity: Given $w_k = (a, b)$ then $w_{k\cdot1} = (a', b')$ in which $a - a' \geq 0$ and $b - b' \geq 0$. This compels that the points in W are monotonically spaced in time.

Even though there are exponentially many warping paths that satisfy the above conditions, we are only interested in those that minimize the following warping cost:

$$\text{DTW}(C, Q) = min \left\{ \sqrt{(\sum_{k=1}^{K} w_k)} \right. \tag{2.5}$$

By employing dynamic programming it is possible to find this path, by evaluating the following recurrence which defines the cumulative distance $\gamma(i, j)$ as the distance $d(i, j)$ found in the current cell and the minimum of the cumulative distances of the adjacent elements:

$$\gamma(i, j) = d(c_i, q_j) + \min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \tag{2.6}$$

The resulting matrix of the Eq. 2.6 can be seen in Fig. 2.3, in which the optimal path, that satisfies the aforementioned warping path constraints, is shown in the solid squares.

A practical example of this is shown in Section 2.4.1.1.

4	9	4	1	0
3	4	1	0	1
3	4	1	0	1
2	1	0	1	4
2	1	0	1	4
1	0	1	4	9
1	0	1	4	9
	1	2	3	4

Table 2.1: Matrix of distances between $\boldsymbol{C}$ and $\boldsymbol{Q}$.

4	19	6	1	0
3	10	2	0	1
3	6	1	0	1
2	2	0	1	5
2	1	0	1	5
1	0	1	5	14
1	0	1	5	11
	1	2	3	4

Table 2.2: Matrix of the optimal warping path

2.4.1.1 Dynamic Time Warping Example

Suppose we intent to compare two signals and evaluate the difference that exists between them, by calculating their cost of similarity. Using two simple example signals, $\boldsymbol{C}$ and $\boldsymbol{Q}$, where one is the simple compression of the other, let

$$\begin{aligned} \boldsymbol{C} &= \{1, 1, 2, 2, 3, 3, 4\} \\ \boldsymbol{Q} &= \{1, 2, 3, 4\} \end{aligned} \tag{2.7}$$

be the data in these two sequences. Here, $\boldsymbol{Q}$ is a smaller, compressed version of $\boldsymbol{C}$. To solve this problem we start by building a 7×4 matrix, since $n = 7$ and $m = 4$. Moreover, on each side of the matrix we put the two signals that will help us calculate each position of the matrix. To fill each (i^{th}, j^{th}) position of the matrix, we make the following calculation $d(c_i, q_j) = (c_i - q_j)^2$. Such can be seen in Table 2.1. After the matrix of distances is completed, it is possible to calculate the matrix with the optimal warping path, that can be seen in Table 2.2. To populate this matrix, as formulated in Eq. 2.6, each (i^{th}, j^{th}) position of the matrix is equal to the value of the same position in Table 2.1 plus the minimum value among the adjacent cells – $(i-1, j-1), (i-1, j), (i, j-1)$. Moreover, the similarity cost between the two signals is equal to the value in the top right corner in the Table 2.2. Due to the similarity among these two signals, the cost between them is 0.

2.4.1.2 Fast Dynamic Time Warping

FastDTW [SC07] is a proposed algorithm that has the same objective but achieves a linear time and space complexity, by avoiding the brute-force dynamic programming approach of the standard DTW algorithm and using a multilevel approach. Comparing results among two implemented DTW algorithm optimizations, Sakoe-Chuba band and data abstraction, FastDTW shows much better results. Looking at the average error,

	radius				
	0	*1*	*10*	*20*	*30*
FastDTW	19.2%	8.6%	1.5%	0.8%	0.6%
Abstraction	983.3%	547.9%	6.5%	2.8%	1.8%
Band	2749.2%	2385.7%	794.1%	136.8%	9.3%

Table 2.3: Average error of three algorithms at the selected radius value (errors of the 3 groups of data are averaged). Taken from [SC07].

it decreases as the radius parameter increases for all algorithms. Moreover, FastDTW converges to a error of 0% much faster than others. Therefore, in terms of accuracy, FastDTW is better than the compared optimizations, as seen in Fig. 2.3.

e

Regarding efficiency, as seen in Fig. 2.4, FastDTW shows faster results than DTW for small time series, being the radius an influential factor in the execution time. For larger time series, FastDTW runs much more quickly. The main limitation of the FastDTW algorithm is that, in some cases, it is not able to find the optimal solution, as it is an approximate algorithm, which can be an issue if the problem requires optimal warp paths [SC07].

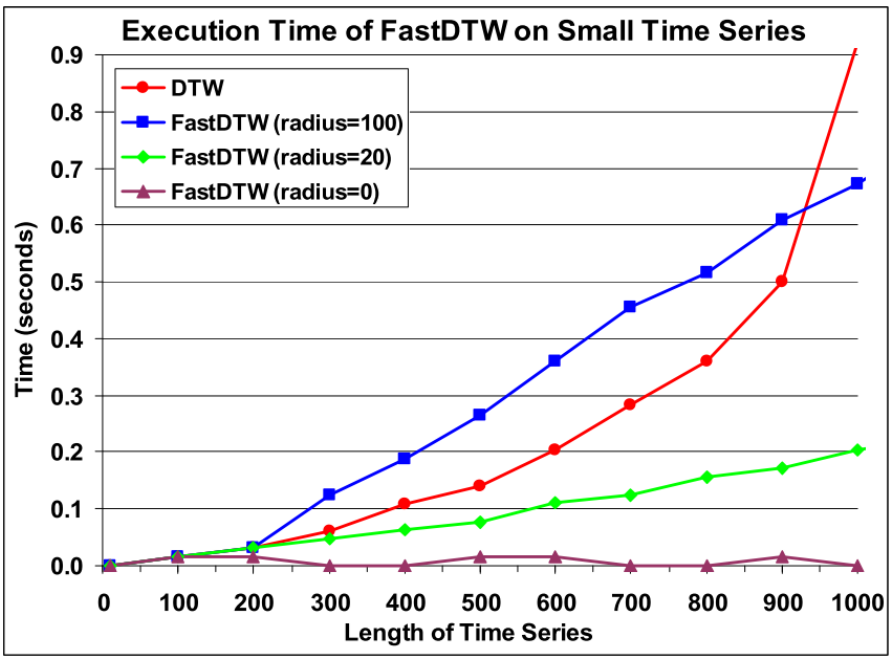


Figure 2.4: The efficiency of FastDTW and DTW on small time series. Taken from [SC07].

DTW has been commonly used in recent works regarding accelerometer-based gesture recognition like [AV10], [AFV11] or uWave [LZWV09].

2.4.2 Hidden Markov Models

A Hidden Markov Model is a stochastic state machine that consists in a Markov chain with a finite number of states where each state is associated with a set of random functions. It has been widely applied in areas of time series recognition, like speech or gesture, that have temporal and spatial variability [KKM+05].

In this kind of model, some of the variables influenced by the states are directly observable but the states themselves are hidden. A probability distribution (typically Gaussian) is associated with every state, influencing the possible output tokens. In practical uses, HMM's takes time series of observation as input and returns the probability that the input data is generated by that model [ZCWY09].

In terms of computation complexity, it is directly proportional to the dimension and number of the feature vectors, which can be an issue if a big dictionary of gestures is used [AV10].

HMM has also been used in the past regarding gesture recognition, like in [ZBA09] or [Pyl05].

[Pyl05] is an accelerometer based gesture recognition using continuous HMMs. In their work, they found beneficial to normalize data in order to compensate the offset that is introduced by the tilt of the device. This tilt mainly occurs due to the orientation of the device given the shape of the human hand and arm. In terms of results, recognition accuracy was above 90% for samples with a sampling rate from 15 to 30 Hz, except for gestures that had duration inferior to a second; in this case, for some rates, gestures would even be unrecognizable. According to Timo [Pyl05], these results are valid with a set of at least 10 gestures, as in the tests.

2.4.3 Affinity Propagation

Clustering data based on a measure of similarity is a critical step in engineering systems. An usual approach is to use the data to learn a set of centers such that the sum of squared errors between data points and their nearest centers is small. When such centers are selected from actual data points, they are denominated exemplars. A recent technique for data clustering is Affinity Propagation (AP) algorithm [FD07].

AP is an algorithm that simultaneously considers all data points as potential exemplars and recursively exchanges real-valued messages among these until a high-quality set of exemplars and corresponding clusters emerges. Exemplar is the nomenclature

used to denominate the selected center of a cluster from the existing data points. For example, and as it is used in this thesis, the similarity function can be the negative Euclidean distance between data points; such is applied so that a maximum similarity corresponds to the closest data points.

Furthermore, Akl et al. choose AP as clustering technique, because it operates on a matrix of similarities between data points instead of operating on feature vectors or raw data [AFV11]. As the similarity costs are computed between gesture traces, clustering is based on their temporal characteristics. In their work, AP is used in a configuration that doesn't require the need for forcing all traces to be of the same length as it is done in other works.

Ultimately, when compared to other clustering techniques like the K-means, AP shows better results, mainly due to the initialization independent property [FD07].

2.4.3.1 Input and Tuning

AP takes as input a collection of real-valued similarities between data points, where the similarity $s(i, k)$ indicates how well suited data point k is to be the exemplar for data point i.

Also, instead of requiring that the number of clusters is specified, AP takes as input a set of real numbers $s(k, k)$ for each data point k, known as preference (p). The data points for whose preference (p) is larger are more likely to be chosen as exemplars. However, if all data points are initialized with the same preference constant, then all of them will have the same probability of becoming an exemplar.

In a similar way, using the same preference value (p) for all data points also directly influences the number of clusters that is given as output. By increasing the preference value used as input, the total number of clusters is also increased. However, if this value is decreased, the same happens to the number of clusters.

If this shared value is the median of the input similarities, the result will be a moderate number of clusters, and if such value is their minimum, it will result in the smallest number of clusters.

2.4.3.2 Message exchange

There are two kinds of messages that are exchanged between data points: "responsibility" and "availability". Each of them takes into account a different kind of competition. While the first, "responsibility", is used to decide which data points are exemplars, the "availability" is used to decide which cluster a data point belongs to. Such can be summarized as:

- the "responsibility" $r(i,k)$, sent from data point i to candidate exemplar k, reflects the accumulated evidence for how well-suited point k is to serve as exemplar of point i, taking into account other potential exemplars k' for data point i. The responsibilities are calculated as following:

$$r(i,k) \leftarrow (i,k) - \max_{k's.t.k' \neq k} \{a(i,k') + s(i,k')\} \tag{2.8}$$

  where $i \neq k$, $s(i,k)$ is the similarity between data point i and k and $a(i,k)$ is the availability message defined in the next item;

- the "availability" message $a(i,k)$, sent from candidate exemplar point k to data point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other data points that k should be an exemplar. The availabilities are calculated as following:

$$a(i,k) = \min \left\{ 0, r(k,k) + \sum_{i's.t.i' \notin \{i,k\}} \max\{0, r(i',k)\} \right\} \tag{2.9}$$

- both self-responsibility $r(k,k)$ and the self-availability $a(k,k)$ reflect accumulated evidence that point k is an exemplar. However, self-responsibility is based on the input preference and by the maximum availability received from surrounding data points whilst the self-availability bases the exemplar suitability on the positive responsibilities sent to candidate exemplar k from other points. Such is expressed as:

$$r(k,k) = s(k,k) - \max_{k's.t.k' \neq i} \left\{ a(k,k') + s(k,k') \right\} \tag{2.10}$$

$$a(k,k) = \sum_{i's.t.i' \neq k} \max\{0, r(i',k)\} \tag{2.11}$$

2.4.3.3 Algorithm Stabilization

In AP, for each point i, its exemplar, and the cluster exemplar itself in case $k = i$, is found the in following manner:

$$exemplar_i = arg\max\{a(i,k) + r(i,k)\} \tag{2.12}$$

Although the exemplar choosing and clustering procedure may be terminated at any iteration, it is important that such, and message-passing, is only terminated after the algorithm stabilization. Multiple options arise to pick such point that can be after a fixed number of iterations, after changes in the messages fall bellow a threshold or after the local decisions stay constant for a certain number of iterations.

2.4.4 Compressive sensing

Compressive sensing (CS) theory asserts that it is possible to recover certain signals from far fewer measurements than the traditional sampling methods [CW08]. For further example assume that a signal can be expressed as a $d \times 1$ vector $x = \Psi s$ where Ψ is a $d \times d$ matrix and s is a $d \times 1$ sparse vector that has only $l_s \ll d$ nonzero elements. The location of each nonzero element in s is unknown. This method is based on two principles: sparsity and incoherence:

- sparsity, which is relative to the signals of interest, expresses the idea that information rate of a continuous time signal may be much smaller than its bandwidth suggests. Also, it takes advantage on the fact that many natural signals are sparse or compressible, in the sense that they have concise representations when expressed in the proper Ψ;
- incoherence, which is relative to the sensing modality, extends the duality between time and frequency and denotes that objects with a sparse representation in Ψ must be spread out in the domain in which they were acquired. Also, that the sampling waveforms have an extremely dense representation in Ψ.

x is compressed using a $k \times d$ sensing matrix Φ, which yields the vector y with dimension k as following:

$$y = \Phi x = \Phi \Psi s \tag{2.13}$$

Also, it is possible to perfectly recover s if k meets the following inequality:

$$k \geq c l_s log(d/l_s) \tag{2.14}$$

in which c is a constant and l_s is the sparsity level. [CW08]

Afterward, it is possible to reconstruct the signal by solving the following ℓ_1 norm minimization problem:

$$\min_s \|s\|_1 \quad \text{subject to} \quad y = \Phi \Psi s \tag{2.15}$$

2.4.4.1 Restricted Isometry Property

As referred in [BDDW08], in the discrete compressive sensing problem, we are interested in economically recording information about a vector (signal) $x \in \mathbf{R}^d$, where d is generally a large value. We allocate a budget of k non-adaptive questions to ask about x and each of these take the form of a linear function applied to x. Therefore, the information we extract from x is given as following:

$$y = \Phi x \tag{2.16}$$

where Φ is a $k \times d$ matrix and $y \in \mathbf{R}^k$. The matrix Φ maps $\mathbf{R}^d$ into $\mathbf{R}^k$, where k is usually much smaller than d.

To extract the information y contains about x, we use a decoder Δ that maps from $\mathbf{R}^k$ back again into $\mathbf{R}^d$. The role of Δ is to provide an approximation $\bar{x} := \Delta(y) = \Delta(\Phi x)$ to x. The mapping Δ is typically non-linear.

The central question of CS is: "What are the good encoder-decoder pairs (Φ, Δ) ?"

To measure the performance of an encoder-decoder pair (Φ, Δ), we introduce a norm $\|\cdot\|_X$ in which we measure the error. Then,

$$E(x, \Phi, \Delta)_X := \|x - \Delta(\Phi x)\|_X \tag{2.17}$$

consists in the encoder-decoder error on x. More generally, if $\mathbf{K}$ is any closed and bounded set contained in $\mathbf{R}^d$, then this encoder-decoder error on $\mathbf{K}$ is given by

$$E(\mathbf{K}, \Phi, \Delta)_X := \sup_{x \in \mathbf{K}} E(x, \Phi, \Delta)_X \cdot \tag{2.18}$$

Thus, the error on set $\mathbf{K}$ is determined by the largest error itself contains. Regarding the question of what constitute good encoder-decoder pairs, we introduce $\mathcal{A}_{k,d} := \{(\Phi, \Delta) : \Phi \text{ is } k \times d\}$

The best possible performance of an encoder-decoder on $\mathbf{K}$ is given by

$$E_{k,d}(\mathbf{K})_X := \inf_{(\Phi,\Delta) \in \mathbf{A}_{k,d}} E(x, \Phi, \Delta)_X \cdot \tag{2.19}$$

This is the so-called "minimax" way of measuring optimality that is prevalent in approximation theory, information-based complexity and statistics [BDDW08].

The decoder Δ is important in practical applications of compressive sensing. Candès, Romberg and Tao [CRT06b] have shown that decoding can be accomplished by the

linear program

$$\Delta(y) := \underset{x\,:\,\Phi x = y}{\arg\min} \|x\|_{\ell_1^d}. \tag{2.20}$$

Candès and Tao [CT05] introduced the isometry condition on matrices Φ and established its important role in compressive sensing. Given a matrix Φ and any set $\mathbf{T}$ of column indices, we denote by $\Phi_{\mathbf{T}}$ the $k \times \#(\mathbf{T})$ matrix composed by these columns. Similarly, for a vector $x \in \mathbf{R}^d$, we denote by $x_{\mathbf{T}}$ the vector obtained by retaining only the entries in x corresponding to the column indices $\mathbf{T}$. We say that the matrix Φ satisfies the *Restricted Isometry Property* (RIP) of order m if there exists a $\delta_m \in (0, 1)$ such that

$$(1 - \delta_m)\|x_{\mathbf{T}}\|_{\ell_2^d}^2 \leq \|\Phi_{\mathbf{T}} x_{\mathbf{T}}\|_{\ell_2^k}^2 \leq (1 + \delta_m)\|x_{\mathbf{T}}\|_{\ell_{2d}}^2 \tag{2.21}$$

holds for all sets $\mathbf{T}$ with $\#(\mathbf{T}) \leq m$. The condition (2.21) is equivalent to requiring that Grammian matrix $\Phi_{\mathbf{T}}^T \Phi_{\mathbf{T}}$ has all of its eigenvalues in $[1 - \delta_k, 1 + \delta_k]$, being $\Phi_{\mathbf{T}}^T$ the transpose of $\Phi_{\mathbf{T}}$

The "good" matrices for compressive sensing should satisfy 2.21 for the largest possible m. In [CT05] Candès and Tao, show that whenever Φ satisfies the RIP of order $3m$ with $\delta_{3m} < 1$, then

$$\|x - \Delta(\Phi x)\|_{\ell_2^d} \leq \frac{C_2 \sigma_m(x)_{\ell_1^d}}{\sqrt(m)} \tag{2.22}$$

where $\sigma_m(x)_{\ell_1^d}$ denotes the ℓ_1 error of the best m-term approximation and where the constant C_2 depends only on δ_{3m}.

The proof of 2.22 particular formulation is available in [CDD07] while the original proof is accessible in [CRT06a].

The question before us now is: "How can we construct matrices Φ that satisfy the RIP for the largest possible range of m?"

Beforehand, it is important to note that verifying the RIP may be a difficult task. First, it requires bounded condition number for all sub-matrices built by selecting $m := \#(\mathbf{T})$ and second, the spectral norm of a matrix is not generally easy to compute.

Nevertheless, the most outstanding example that answers this question is the $k \times d$ random matrices Φ whose entries $\phi_{i,j}$ are independent realizations of Gaussian random variables [BDDW08]

$$\phi_{i,j} \sim \mathcal{N}(0, \frac{1}{n}). \tag{2.23}$$

Likewise, one can also use matrices where the entries are independent realizations of ± 1 Bernoulli random variables, such as

$$\phi_{i,j} := \begin{cases} +1/\sqrt{n} & \text{with probability } \frac{1}{2}, \\ -1/\sqrt{n} & \text{with probability } \frac{1}{2}, \end{cases} \tag{2.24}$$

or related distributions, such as

$$\phi_{i,j} := \begin{cases} +\sqrt{\frac{3}{n}} & \text{with probability } \frac{1}{6}, \\ 0 & \text{with probability } \frac{2}{3}, \\ -\sqrt{\frac{3}{n}} & \text{with probability } \frac{1}{6}, \end{cases} \tag{2.25}$$

2.4.5 Random Projection

Random projections (RP) has emerged as a powerful technique employed for dimensionality reduction [BM01] [LG03]. This kind of technique helps searching and exploring vast amounts of data while being computationally and time-efficient. Moreover, it helps reducing the computational cost of large amounts of data, by reducing their dimensionality and allowing to compute a smaller set.

An ideal dimensionality technique has the capability of efficiently reducing a d-dimensional data into a k-dimensional subspace, where $k \ll d$, while still preserving the original data. This is done by using a $k \times d$ matrix $\mathbf{A}$ whose columns have unit lengths. Using matrix notation, where $\mathbf{X}_{d\times n}$ is the original set o n d-dimensional observations, then

$$\mathbf{X}^{RP}_{k\times n} = \mathbf{R}_{k\times d}\mathbf{X}_{d\times n} \tag{2.26}$$

where $\mathbf{X}^{RP}_{k\times n}$ is the projection of the data into a lower k-dimensional subspace. The concept of random projection is inspired by the Johnson-Lindenstrau lemma [JL84], which states that if points in a vector space are projected into a randomly selected subspace of high dimension, then the distances between the points are approximately preserved, as proven in [FM88] and [DG99].

Strictly speaking, the formulation in 2.26 is not a projection because $\mathbf{A}$ is not usually orthogonal and a linear mapping in such cases can cause considerable distortions in the data set. One can resort to the orthogonalization of $\mathbf{A}$, but such is computationally expensive. However, according to a result by Hecht-Nielsen et. al [HN94], in high-dimensional spaces, there exists a much larger number of almost orthogonal directions than the actual number of orthogonal directions. Therefore, the random vectors may

be close enough to orthogonal to provide a reasonable approximation to the original vectors.

Regarding RP, $\mathbf{A}$ is the general case of Φ, in CS (see Section 2.4.4) and also RIP (see Section 2.4.4.1). Moreover, $\mathbf{A}$ is a sampling operator for $\mathbf{X}$ and is invertible if each $\boldsymbol{x} \in \mathbf{X}$ is uniquely determined by its projected data $\mathbf{A}\boldsymbol{x}$, which can be formulated as in Eq. 2.27. Actually, matrix $\mathbf{A}$ is a one-to-one mapping between $\mathbf{X}^{RP}$ and $\mathbf{X}$, which allows the identification for each $\boldsymbol{x} \in \mathbf{X}$ from $\mathbf{A}\boldsymbol{x}$.

$$\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbf{X}, \;\; \mathbf{A}\boldsymbol{x}_1 = \mathbf{A}\boldsymbol{x}_2 \rightarrow \boldsymbol{x}_1 = \boldsymbol{x}_2 \tag{2.27}$$

Nonetheless, we want that a small change in $\boldsymbol{x}$ produces a small change in its projected data $\mathbf{A}\boldsymbol{x}$. So, a stricter condition is given, as expressed in Eq. 2.28, where $\alpha > 0$ and $\beta < \infty$ and are both constants. $\mathcal{H}$ is an Hilbert space, and $\psi_n \in \mathcal{H}$ is a sampling vector [LD08].

$$\alpha\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_{\mathcal{H}}^2 \leq \|\mathbf{A}\boldsymbol{x}_1 - \mathbf{A}\boldsymbol{x}_2\|_{l_2}^2 = \sum_n |\langle \boldsymbol{x}_1 - \boldsymbol{x}_2, \psi_n \rangle|^2 \leq \beta\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_{\mathcal{H}}^2 \tag{2.28}$$

The sampling condition expressed in Eq. 2.28 on $\mathbf{A}$ is related to the concept of RIP and is the same as RIP if $\mathbf{X}$ has sparse columns and the columns come from the same subspace [LD08] [CT06]. Moreover, as mentioned in Section 2.4.4.1, any distribution of zero mean and unit variance satisfies the condition expressed in Eq. 2.28.

2.5 Environmental Awareness

Nowadays, a huge effort demand is being put upon Earth's natural resources and biodiversity by different sectors. However, this demand is not predicted to decrease but instead to get only higher in the next decades. So, it is of much importance that citizens overall have a more crucial role in promoting and conserving the environment.

Also, as stated in OECD Environmental Outlook to 2030 [OEC08], "Without more ambitious policies, increasing pressures on the environment could cause irreversible damage within the next few decades".

There are key sectors that need special attention, like Energy, Transportation, Agriculture and Capture fisheries; OECD highlights priority actions in these sectors [OEC08].

Overall, the principal greenhouse gas that causes climate changes, carbon dioxide, is expected to increase in the next decades. Energy sector's related carbon dioxide emissions, without any added policies, are estimated to grow by 52% to 2030. Transport

related carbon emissions are projected to increase by 58% to 2030.

As also stated in OECD Environmental Outlook to 2030, "Many environmental challenges cannot be solved by environment ministries alone" and that most environmental problems can only be solved, among other measures, with the co-operation with business and civil society."

Regarding energy consumption, its increasing consumption is a problem that affects the environment worldwide [McC01]. Even though governments throughout the world are promoting the development of renewable energy sources and multiple international agreements aim to restrain the emissions such as CO^2, people's personal energy consumption behavior still remains relatively unchallenged [BTK06]. Moreover, in the past 30 years, the electricity usage has grown about 30% [Age05] and the potential of consumption reduction in a domestic environment exists [Dar06].

Considering these factors, it is important that every citizen is fully aware of the main causes that are endangering the environment and what they can change in their behaviors to help contributing to reduce Earth's consumption of natural resources and pollution.

In the scope of the DEAP project it is intended to make people aware of the environmental situation we are currently living in and also to persuade them into having pro-environmental behaviors, particularly to make them know the particular key sectors that most represent a danger to the world's environmental resources and biodiversity. As this thesis was developed in the scope of the DEAP project, we address this matter in the proof of concept.

2.6 Interaction with Public Ambient Displays

In the matters of interaction with public ambient displays, individuals direct interaction activities can be classified in two types: overt or covert [KFL08]. Overt interaction occurs when an individual uses designated input devices like keyboard or explicit physical movement, which may result in a learning by watching and also encourage surrounding people to interact as well. On the other side, it can cause social embarrassment, discouraging the interaction with the display. The latter, covert interaction, exists when there is interaction between a phone or implicit physical movement. This type of interaction allows a greater level of privacy and it may increase the user confidence levels. However, not as much information is passed to others reducing the acquisition of knowledge that can be achieved with the display interaction.

In their study, Kaviani et al [KFL08], show that covert interaction reduces considerably the chance of learning by watching and that this learning happens mainly in

groups where members know each other. They also conclude that, in order to enable both covert interaction and learning, more emphasis on the information part should be given, since learning by watching among strangers is not supported in the same way as, for example, touch screen applications. Therefore, instructions on how to interact were given almost 50% of the display space in the applications. Besides, font sizes would be chosen accordingly to the detail level of information, where the most detailed would have the smallest font. In this way, those who were more interested in the information would get closer to the display. For people who were just passing in the street, the process of starting to read large and less detailed information could lead to getting close to the display and even result on their interaction as actors. Results show that a 95% success rate in the learning ability was achieved [KFL08].

Previous works in this area have been done like MAID [Sal12] or Gaea [Cen11], both further addressed in this document. Besides, works like CityWall [PK08], are also relevant in this area. CityWall is a large multi-touch display that was installed during eight days in the city center of Helsinki, Finland. In these eight days, 1199 people, in various social configurations, interacted with the system. Peltonen et al. [PK08] show how a public ambient display can become a place that triggers strangers to come in contact and interact between them.

These different approaches and related works can be useful when designing the game proposed by this thesis, in order to attract different groups of people near the display and, most of all, to motivate people to play it.

2.6.1 Technology as human social relationship capacitor

As stated in [Aga05], humans have a basic need for contact with other humans, even pointing out that the lack of this social relationship can constitute a risk factor for health that can even exceed those caused by cigarette smoking or lack of physical activity. Stefan Agamanolis states that new technologies truly affect human behavior in relationships as well, being able to act as mediator or even a catalyst.

That made, the game developed in our work as a proof of concept for the framework, enables users to play against each other. It is played in a one-to-one basis and, to make it more competitive the users shall be awarded points each time they execute a correct action. Thus, participant's who answer correctly to the given problems, which is done by the executing gestures, will be awarded more points. Hence, a better entertainment experience with the application, and thus internalization of the concepts that are intended to be passed to the users, is expected to occur.

2.7 Persuasive Technology in Gaming context

Persuasive technology is defined as the group of technologies that intent to apply changes on user's attitudes and behaviors without the use of coercion, which is the use of force or threats.

B.J. Fogg coined the term Captology as a new work that describes the study and use of computers as persuasive technologies. Particularly, it explores how people attitudes and behaviors change when interacting with computers itself rather than using them as communication mediators. In terms of intent, there is an universe of three computer intent types: endogenous, when a designer or producer creates a technology with the intent to persuade others in some way; exogenous, when one provides another with a computer technology with the intent of changing its attitudes; and lastly, autogenous, when a person chooses to use a technology in order to change his own behaviors. Captology focus in the first one, endogenous. In terms of persuasion scale, there can be considered two magnitudes: macro, when a product is designed with the pure intent of persuasion and micro, when a product includes persuading elements but its main objective is not persuasion [Fog98] [Fog02].

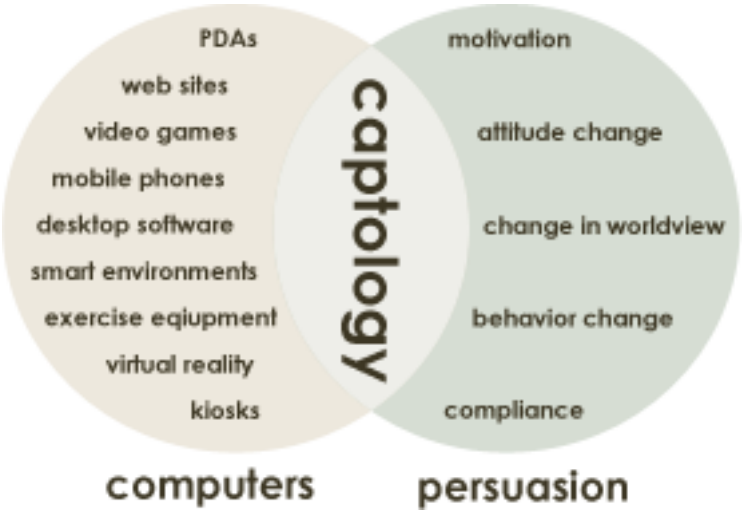


Figure 2.5: Captology: How computing and persuasion overlap. Taken from [Cer11].

Persuasive technology can be applied in the field of Human-Computer Interaction (HCI) once that people respond socially to computer products as they would in different social situations [LHS08] [Fog02]. Also, there have been previous work applying these technologies in various fields such as Management, Health [OCO10], Education or Commerce. Regarding environment, multiple works have been done in the past, as

[FDK+09], eVision [San12] [SRaD+12], MAID [Sal12] or Gaea [Cen11].

PowerHouse [BTK06] is a computer game designed to influence behaviors associated with energy use and promote an energy-aware lifestyle among teenagers. With the foundation that the world's energetic consumption in the past 30 years has increased about 50% and that the domestic electric consumption can be cut significantly, it aims to explore new design and approaches that can be used to raise awareness of energy-related issues. Moreover, Bang et al. [BTK06] state that people are increasingly playing computer games and that these games seem to be particular powerful to influence decision-making and user behavior. Their pre-game studies concluded that teenagers' knowledge on how much energy is being used by different domestic activities is generally quite low, with a rate of 50% correct answers. In order to appeal the participants, the game format consists in a TV reality show (docu soap) where people live together in a closed house, similar to the popular game Sims [Spo13]. This game uses persuasion by modeling energy consumption of different activities in a home. For instance, if a player is taking a hot shower the energy meter will show this immediately and the money meter will drop.

eVision [San12] is an application with the goal of informing users about environmental threats in their surroundings as well as their corresponding consequences. Besides, it has also the objective of encouraging users to change their attitude towards the environment.

In order to accomplish this goal it uses augmented reality, detecting cars, planes and bicycles. Also, it offers digital rewards for completing pro-environmental tasks, by means of giving the user an in-app currency – green leafs – that allows the in-app buying of items to customize the game character. These are rewarded when a user finishes a virtual cleaning task, for example. Furthermore, it creates a bond between the user and the virtual character in order to make users more receptive to suggestions, by allowing the user to customize the game character.

MAID [Sal12] uses a persuasive game interface that is based on gesture recognition technology. For this recognition, Salvador has chosen a camera-based environment using the already addressed Microsoft Kinect. Its main objective is to keep users alert about simple ways of saving energy at home by either teaching them procedures to do so, or by simply showing how these can have a large effect summing the overall's savings. Moreover, MAID's is based in the idea that small individual savings in the overall picture can have a great impact on the environment and also that the most common source of energy are fossil fuels, which leads to the emission of green-house gases. Hence, by reducing energy consumption at home, either by using it less or in a more efficient way, people are able to save world resources and reduce the green-house gas

emissions.

Gaea [Cen11] is a location-based multiplayer mobile game that prompts users to recycle virtual objects within a specified spatial context. Its main goal is to persuade users to recycle their wastes in a correct manner, promoting entertainment and social engagement, combining mobile devices with a public display. To do so, players need to collect virtual garbage that is shown in maps on their mobile phones. The objects are linked to specific geographic coordinates and users need to go there, pick them up and then go back to the public display and drop the item into the right recycling garbage collector. When all the garbage has been collected a quiz starts in the public display. The questions are related to information that was presented when dropping garbage and users can answer them in their mobile phones. If they answer correctly they will earn extra points. This way, the users are motivated to pay attention to the provided information in order to win the game.

Even though the game built in this thesis is developed merely as the proof of concept for the framework, it was the author's intention to transfer a pro-environmental message to the players. Hence, such was addressed in key sectors that most contribute to the worldwide environment degradation, such as energy consumption and overall resource waste.

2.8 Smartphone games with interaction by the use of accelerometer based gestures

There is a wide range of games in the Apple® App Store for iOS that make use of the accelerometer to receive actions from the user. Most of them require that the device is tilted to perform a game character trajectory change. The following list addresses a few:

- Doodle Jump is a traditional platform game in which the game character continuously keeps moving up and must not miss a platform or will fall. To control the character position the user must tilt the device to the desired orientation.
- Labyrinth 2 is a game in which a maze must be crossed with a sphere. During this, multiple obstacles such as walls and holes harden the level. The user is able to control the level gravity by moving the device, thus affecting the sphere position that is subject to this gravity.
- Temple Run is an endless running game in which different curves and objects constantly appear. If the character does not turn into the right direction or hits an

object he loses. In this case the user must tilt the device in order to move right and left along the path, for picking coins. Moreover, this game combines the use of screen swipes in order to make the correct turns and jumps.

The developed work in this thesis takes the concept of reading the accelerometer data even further, once programmers are easily able to define gestures (not only device tilting) and recognize them during game-play.

3

Gesture Recognition System

The gesture recognition system was built having in mind that other developers would use it as a framework to develop their own applications. Thus, the main goals set while it was designed were: the developer abstraction of what is happening inside the framework; a fast and accurate gesture recognition; easy to integrate as an external library and, most of all, easy to use.

By using this system any iOS developer is able to easily combine his software with gesture recognition. Moreover, being capable of using a predefined dictionary of gestures that comes with the library, any other dictionary that is loaded into the framework and further, the possibility to add new gestures to the system that can be stored and migrated to other systems.

The gesture acquisition is done by using the iPhone, iPod or iPad embedded three-axis accelerometer. Each axis is represented by an array, thus a gesture trace is composed by a three-dimensional matrix.

The framework was developed for iOS 6.1.

3.1 Context

There are multiple ways of interacting with a system, particularly a smartphone: through voice, touch or even by tilting the users head. For the last case, we have the example of the most recent smartphones when such action can be used for swiping through screen

menu options. However, the gesture recognition system used in this thesis focus and makes use of the interaction based in gesture movements.

These type of movements are obtained when a user moves his hand while also holding the device. This is possible by reading the acceleration of the hand (which is holding the device) at each moment, making use of the device accelerometer. The accelerometer is responsible for continuously collecting the device acceleration, in 3 axis (x,y,z). Such gestures can have different complexities, durations and different associated actions. Regarding its complexity and duration, a gesture can range between a simple hand tilt, from left to right, or a gesture representing a letter or symbol. Moreover, by identifying distinct gestures, it is possible to associate different actions to each one.

The system, presented by Akl et al. [AFV11], is capable of receiving and recognizing gestures, including custom ones that a developer wishes to have. It is composed by N gestures, each of them is represented by a set of traces. For example, a gesture **G** is executed M times by one or multiple users. Each execution is called a trace and this set of different traces all represent the same gesture **G**. Such can be formulated as follows,

$$\begin{array}{c} G_1=\{\boldsymbol{T}_{1,1},\boldsymbol{T}_{1,2},...,\boldsymbol{T}_{1,M}\}, \\ G_2=\{\boldsymbol{T}_{2,1},\boldsymbol{T}_{2,2},...,\boldsymbol{T}_{2,M}\}, \\ \vdots \\ G_N=\{\boldsymbol{T}_{N,1},\boldsymbol{T}_{N,2},...,\boldsymbol{T}_{N,M}\}. \end{array} \tag{3.1}$$

Regarding a trace, it is composed by a set of arrays, each one corresponding to an axis (x,y,z). Inside of each array is the data collected by the device's accelerometer regarding that axis. The number of elements in each of these arrays, to the same trace, is denominated the trace length. Note that for the same trace this length must be equal among all the three arrays, but between different traces this length may differ.

Some traces, according to their characteristics, are nominated to represent a gesture. Moreover, a gesture can have one or more of these. They are used for a faster gesture recognition, as will further be explained, and are described as exemplars. Regarding the remaining traces, they are important to increase the system recognition accuracy.

In order to obtain an higher gesture recognition accuracy, there are quite a few challenges that must be taken into account. These can be related to the sensor calibration or noise, to the fact that different users describe gestures differently and even that the same user, also given to hand trembling, probably won't ever describe the same gesture exactly the same way twice. All these facts imply that a simple gesture brute force matching isn't possible, because the accelerometer values, although may be similar, won't ever be an exact match, besides the temporal and computational complexity associated. So, in order to reduce the above mentioned challenges impact, some procedures are applied. Among these are data smoothing, the use of algorithms like dynamic

time warping and also compressive sensing.

Regarding the system itself, it is composed by two main phases: the training and the recognition.

In the first phase, the system is taught the gestures that it shall recognize in the second phase. A set of gesture traces are used as input and the system is responsible for tuning the data and choosing which of them will be used as exemplars for the correspondent gestures.

The second phase is where a gesture trace is used as input and the system is responsible for finding the correspondent gesture.

Therefore, the first phase is that in which the developer is working on the application, creating the gestures it should be able to recognize, whilst the second phase is for use in a running application environment.

3.2 Procedure

To construct a gesture recognition system it is required that a dictionary of gestures is defined beforehand, so the system can have some information regarding what it shall recognize. Therefore, in this work, for each gesture **G** defined in that dictionary, M traces are collected, in order to create a database that will allow the system the comparison of data. As stated before, a gesture trace data consists in the acceleration of the movement of an hand, which is achievable by a 3-axis accelerometer. Such data is acquired by having a subject moving a device that has an accelerometer and allows the collection of its data.

However, traces can have different durations, once hand gestures suffer from temporal variations, which discloses a challenge in the design of the system. Taking this into account, dynamic time warping algorithm is used to compare different gesture traces. Afterward, affinity propagation works with the results of this comparison as input, separating the different traces among several clusters, each of them represented by an exemplar. This is the core of the first phase of the system. Regarding the exemplars, these are used further in the system, in the recognition phase.

In order to recognize a gesture, similarly to the creation of the system database, a subject moves a device describing the gesture in the air and the system has the goal of unveiling which gesture was performed. That is done by comparing the input trace with the stored exemplars and filtering a subset of traces that may possible match. Supported by the fact that these acquired gesture traces are sparse, the filtered traces and the input are projected into the same subspace, so it is possible to overcome the challenge of the traces with different durations. The projection is done through the use

of two projection matrices: a matrix with sparse data – that has information regarding the gestures – and a matrix with independent Gaussian random variables. Such is only possible because the projection matrices satisfy the restricted isometry property that is required for recovering the original information. After the data is projected, a recognition problem is formulated as an ℓ_1-minimization problem whose solution, in the best case scenario, is the correspondent gesture trace executed by the subject.

3.3 Training phase

It is trivial that, to enable gesture recognition, a system must be able to learn them first. In this case, the system receives as input a set of traces and the correspondent associated gesture. This input is collected with the device accelerometer.

As aforementioned, some procedures need to be applied to increase the gesture accuracy. Among them, the first to be applied is the smoothing of the gesture traces data. So, for every recorded trace, there is a corresponding smoothing phase where noise from hand-shaking, accelerometer deviations or from any other source are attenuated. The proposed approach is to use a moving average filter in each of the three arrays. A result of what would happen to one of these arrays data can be seen in Fig. 3.1.

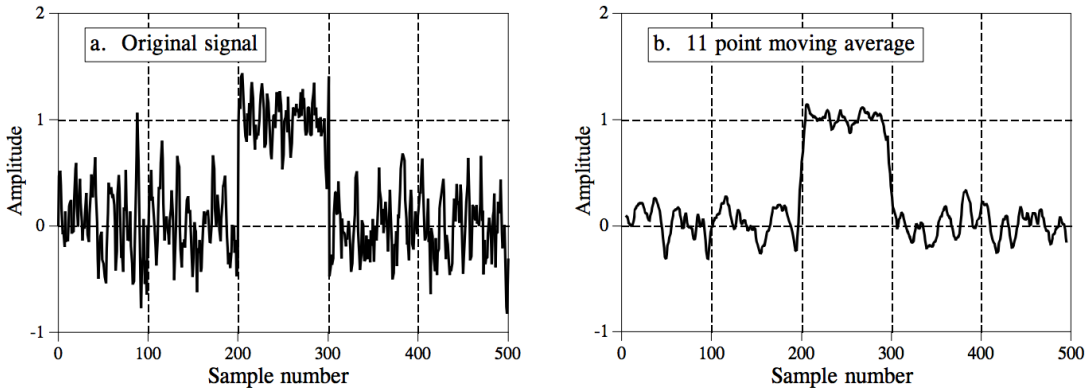


Figure 3.1: Moving average result. Taken from [Smi03].

After the smoothing phase has been accomplished, the next stage goal is to is nominate the traces that will be the gesture exemplars. The main reason why exemplars are used is because it helps reducing the associated computational cost in the latter phase, once a lot of gestures can initially be excluded just by comparison with their exemplars.

This stage is divided into two main tasks, the dynamic time warping and the affinity propagation (please see Sections 2.4.1 and Section 2.4.3 for details).

In the first task, the level of similarity between different gesture traces is computed. Here, DTW is used because it is an easy and fast method of comparing and finding the similarity between two different traces, that may differ in length.

Given two traces $\mathbf{T}_i$ and $\mathbf{T}_j$, of length n and m, the Eq. 3.2 describes how the level of similarity is computed.

$$\text{DTW}(\mathbf{T}_i, \mathbf{T}_j) = \sqrt{D^2_{n,m}(x) + D^2_{n,m}(y) + D^2_{n,m}(z)} \tag{3.2}$$

$D_{n,m}(x)$, $D_{n,m}(y)$ and $D_{n,m}(z)$ are the DTW similarity costs computed between the traces, for each of its axis.

After computing all the costs between the different traces, the results are used as input for the second task.

AP, through the input of the similarity costs, organizes the different traces in clusters, nominating which trace better represents its cluster.

As stated in Section 2.4.3, which regards to the AP clustering algorithm, there are two types of exchanged messages between data points: "responsibility" and "availability". Taking these definition and applying them to our particular case, the first type supports the decision of which traces are exemplars whilst the second type helps deciding to which cluster does a trace belong to. Applied to the gesture recognition problem, the "responsibility" message is given in Eq. 3.3, where $i \neq j$.

$$r(i,j) \leftarrow (i,j) - \max_{j' s.t. j' \neq j} \{a(i,j') + s(i,j')\} \tag{3.3}$$

Defining L as the total number of gesture traces, $s(i,j)$ represents how well the trace $\mathbf{T}_i$ is appropriate to be the exemplar of $\mathbf{T}_j$, and such is expressed in Eq. 3.4.

$$s(i,j) = \text{DTW}(\mathbf{T}_i, \mathbf{T}_j) \;\; \forall i,j \in \{1,2,\ldots,L\} \tag{3.4}$$

Considering the "availability" message, it is calculated as expressed in Eq. 3.5.

$$a(i,j) = \min \left\{ 0, r(j,j) + \sum_{i' s.t. i' \notin \{i,j\}} \max\{0, r(i',j)\} \right\} \tag{3.5}$$

Moreover, besides the measure of similarity, AP can also take as input a set of real numbers known as preference (p), for each gesture trace. Hence, a trace with a larger value of p is more likely to be chosen as exemplar. In this system, this value is proportional to the median of the input similarities, as expressed in Eq. 3.6. In this equation, β is a constant value that controls the number of clusters that will be generated, in an

inversely proportional manner. Thus, if β decreases more clusters are generated. Such adaptation to the original formulation is taken from [AFV11].

$$p = \beta \times \text{median}\{s(i,j) \;\; \forall i,j \in \{1,2,\ldots,L\}\} \tag{3.6}$$

AP is used in this system because, in comparison to other similar techniques like *K*-means, generates better clusters, once it has an initialization independent property [FD07], as expressed in Section 2.4.3.

Note that although AP is a good clustering algorithm, it is not optimal, which means that multiple clusters can be created from a set of traces that correspond to the same gesture. This may also be related to the fact that these traces, as stated before, can be much different from one another. Moreover, the set of exemplars that is taken from the AP outputs is expressed in Eq. 3.7, where $H \geq N$. Note that ξ stands for all the exemplars in the system.

$$\xi = \{\, \mathbf{E}_1, \mathbf{E}_2, \ldots, \mathbf{E}_H \,\} \tag{3.7}$$

This said, the output of AP is a set of clusters and, for each cluster, its exemplars. All these representatives are added to the set of exemplars of that gesture. In our system particular implementation, both the traces, exemplars and clusters are saved and associated with the gesture, once such is useful in the gesture recognition phase.

After this task, the system will have already stored a set of exemplars for each gesture. As will be seen further, this exemplars have a major roll in the recognition phase.

3.4 Recognition phase

When it comes to the recognition of a newly performed gesture trace, as mentioned earlier, it is not feasible the direct comparison of its data with all the existing exemplars as it would compromise the system accuracy and performance.

This phase receives as input a gesture trace, further named as $\mathbf{Y}$, and tries to find any correspondent gesture stored in the system. If such is found, the matching gesture is returned.

Remember that ξ defines all the existing gesture exemplars in the system and let $\mathcal{R}$ define all the traces that are similar to $\mathbf{Y}$. To reach this set of traces the following approach is taken: firstly, in a method similar to the testing phase, $\mathbf{Y}$ is matched against ξ, through DTW. Afterwards, all those traces that are not a close match are discarded in the process, reducing the overall computing cost in the future. Such can be expressed

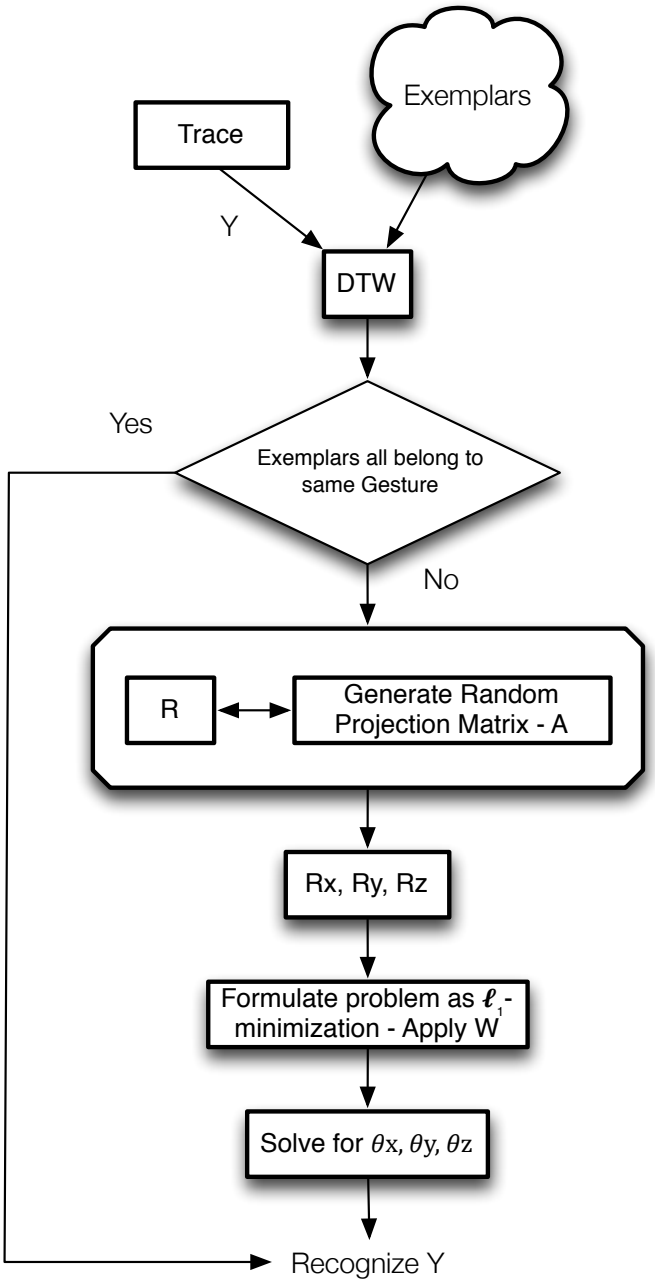


Figure 3.2: Overall diagram block for gesture recognition. Original in [AFV11].

as

$$\mathcal{R} = \{\mathrm{M}_i \,|\, \forall i \;:\; \mathrm{E}_i \in \xi \; and \; \mathrm{DTW}(\mathrm{E}_i, \mathrm{Y}) \; < \alpha\} \tag{3.8}$$

where $\mathbf{M}_i$ is any member of the ith cluster with the exemplar $\mathbf{E}_i$. Also, let κ consist in the union of $\mathcal{R}$ and $\mathbf{Y}$, for future references. Note that a trace may only be the exemplar of one cluster.

As stated by Akl et al. in [AFV11], the threshold shown in Eq. 3.9 has been found to provide the best results. Therefore, such has been used.

$$\alpha = 2 \cdot min\{\text{DTW}(\text{E}_i, \text{Y}), \forall \ \text{E}_i \in \xi\} \tag{3.9}$$

After discarding the traces which don't resemble, and before the recognition process moves to the next part, a validation is made in order to achieve better results: if all the existing traces in $\mathcal{R}$ correspond to the same gesture, that one is returned as output, since the next phase output would be the same gesture. However, if that doesn't happen, the system moves forward.

Afterward, every trace in κ is grouped and projected into the same subspace. Employing this technique helps reducing the overall computation cost of the problem but it also solves the issue with traces that have different durations. Such projection allows the reduction of data while preserving the original data.

Such thing is only possible because the hand gestures appear to be sparse since a smooth trajectory is outlined by the hand, that is holding the device which contains the accelerometer, while a gesture is performed. Thus, it is possible to represent a gesture trace with less data, as shown by the theory of compressive sensing, which is more detailed in Section 2.4.4.

Random projection (RP) has shown to be a good technique for dimensionality reduction and therefore is the employed technique for projecting κ into the same subspace.

However, RP requires that all the traces must be in the same space before the reduction. Such is achieved by determining the largest length l_{max} defined as

$$l_{max} = max\{l_Y, l_1, ..., l_L\} \tag{3.10}$$

where L is the total number of traces in $\mathcal{R}$ and $l_1, \ldots, l_L$ represents their lengths. Such length is equal to the number of elements collected by the accelerometer for that trace in any given axis. After computing l_{max}, all the traces which have a shorter length than l_{max}, are filled with zeros in all axis, until they length reaches l_{max}. Thus, every traces will be in the same space.

The random projection matrix $\mathbf{A}$ satisfies the RIP condition and it makes no difference which of its columns are chosen to compress a trace [CT05]. So, the padded zeros can be placed in the beginning of a trace data, in between or at the end of it. In this

thesis, similarly to the original proposal [AFV11], the zeros are padded to the end of a trace data, for simplicity. This action can be explained by the mathematical formulations expressed in Eq. 3.11 and Eq. 3.12.

$$\mathbf{R}_x = \left[\boldsymbol{r}_1^x, \boldsymbol{r}_2^x, \cdots, \boldsymbol{r}_L^x\right] = \begin{bmatrix} \boldsymbol{r}_{1,1}^x & \boldsymbol{r}_{2,1}^x & \cdots & \boldsymbol{r}_{L,1}^x \\ \boldsymbol{r}_{1,2}^x & \boldsymbol{r}_{2,2}^x & \cdots & \boldsymbol{r}_{L,2}^x \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{r}_{1,l_1}^x & \boldsymbol{r}_{2,l_2}^x & \cdots & \boldsymbol{r}_{L,l_L}^x \\ 0_1 & 0_2 & \cdots & 0_L \end{bmatrix} \tag{3.11}$$

$$\boldsymbol{y}_x = \begin{bmatrix} y_1^x \\ y_2^x \\ \vdots \\ y_{l_y}^x \\ 0_y \end{bmatrix} \tag{3.12}$$

In these formulations, Eq. 3.11 and Eq. 3.12, $\mathbf{R}_x$ is the matrix whose columns represent the x-component of the padded traces and $\boldsymbol{y}_x$ is the matrix whose columns represent the x-component of the padded trace that is being recognized. Regarding $0_1, 0_2, \ldots, 0_L$ and 0_y, they are zero vectors in which the latter has a length of $(l_{max} - l_y)$ while the others have a length of $(l_{max} - l_i)$. Furthermore, $\mathbf{R}_y, \mathbf{R}_z, \boldsymbol{y}_y$ and $\boldsymbol{y}_z$ are processed in the same way, regarding the y and z components.

As stated in Section 3.2, two matrices are needed in order to project the data into a lower subspace. Regarding the matrix with random independent variables $\mathbf{A}$, it has the size of $l_c \times l_{max}$ where l_c is the new common lower subspace. In order to obtain l_c, the sparsity level of each trace in κ must be computed. As stated before, gesture traces data are in fact waveforms composed by smooth curves. To achieve a sparse representation of such kind of data a Fourier transform is a well suited transformation and is therefore used.

Thus, let a sparse sequence be defined as $\boldsymbol{r}$, $\widetilde{\boldsymbol{r}}$ as its Fourier transform, k_r as its sparsity level and r_m as the maximum magnitude in $\widetilde{\boldsymbol{r}}$.

The sparsity level k_r of r is defined in the following inequality:

$$k_r \geq K \times B_\upsilon \tag{3.13}$$

where K is a constant and B_υ is the existing number of samples in $\widetilde{\boldsymbol{r}}$ that are greater than the threshold υ defined the next equation:

$$\upsilon = c \times r_m \tag{3.14}$$

where $c \in [0, 1]$ to keep only the significant values. According to Candès et al., the sparsity level of k_r can be three or four times B_υ, so K is equal to 3 or 4 [CW08].

Moreover, the Fourier transform of a gesture trace $\mathbf{R}$ is defined as

$$\widetilde{\mathbf{R}} = \begin{bmatrix} \widetilde{\boldsymbol{r}}_x & \widetilde{\boldsymbol{r}}_y & \widetilde{\boldsymbol{r}}_z \end{bmatrix} \tag{3.15}$$

and its sparsity as

$$k_{\mathbf{R}} = max\{k_{\boldsymbol{r}_x}, k_{\boldsymbol{r}_y}, k_{\boldsymbol{r}_z}\} \tag{3.16}$$

Defining $\mathbf{L}$ as the total number of elements in κ, the sparsity level $k_{\boldsymbol{R}}$ is computed for all its traces, as in the following expression

$$l_k = max\{k_{\boldsymbol{R}_i}; \forall i \in L\} \tag{3.17}$$

Therefore, after having constructed the random projection matrix $\mathbf{A}$ and the sparse data matrix $\widetilde{\mathbf{R}}$, it's possible to project the data as formulated, for the x-axis:

$$\overline{\mathbf{R}}_x = \mathbf{A}\mathbf{R}_x = \begin{bmatrix} \mathbf{A}\boldsymbol{r}_1^x, \mathbf{A}\boldsymbol{r}_2^x, \cdots, \mathbf{A}\boldsymbol{r}_L^x \end{bmatrix} \tag{3.18}$$

and

$$\overline{\boldsymbol{y}}_x = \mathbf{A}\boldsymbol{y}_x \tag{3.19}$$

where $\mathbf{R}_x$ consists in the projected data in the x-axis into the subspace and $\overline{\boldsymbol{y}}_x$ consists in the projected input gesture trace.

Their relationship can be expressed as

$$\overline{\boldsymbol{y}}_x = \mathbf{R}_x\boldsymbol{\theta}_x \tag{3.20}$$

where $\boldsymbol{v}_x$ is theoretically a $L \times 1$ zeros vector, except $\boldsymbol{\theta}_x(n) = 1$, such that $\boldsymbol{r}_n^x$ is the trace position that better matches $\boldsymbol{y}_x$.

$$\boldsymbol{\theta}_x = \begin{bmatrix} 0, \cdots, 0, 1, 0, \cdots, 0 \end{bmatrix}^T \tag{3.21}$$

where T stands for transposition. Nonetheless, as referred before, gesture traces

have temporal deviations thus it is impossible to have a perfect match and a error must be associated, forcing us to reformulate the problem in the following manner

$$\overline{\boldsymbol{y}}_x = \mathbf{R}_x \boldsymbol{\theta}_x + \varepsilon_x \tag{3.22}$$

where the associated noise is ε_x.

Adopting the same formulation as [FAVT09], the preprocessor $\mathbf{W}$ is introduced, defined as:

$$\mathbf{W}_x = \mathbf{Q}_x \overline{\mathbf{R}}_x^{\dagger} \tag{3.23}$$

where $\mathbf{Q}_x = \mathrm{orth}(\overline{\mathbf{R}}_x^T)^T$, $\mathrm{orth}(\overline{\mathbf{R}}_x)$ is an orthogonal basis for the range of $\overline{\mathbf{R}}_x$ and $\overline{\mathbf{R}}_x^{\dagger}$ is the pseudo-inverse of the matrix $\overline{\mathbf{R}}_x$.

The gesture recognition problem moves to a new formulation

$$\boldsymbol{h}_x = \mathbf{W}_x \overline{\boldsymbol{y}}_x = \mathbf{Q}_x \boldsymbol{\theta}_x + \varepsilon'_x \tag{3.24}$$

where $\varepsilon'_x = \mathbf{W}_x \varepsilon_x$. $\boldsymbol{\theta}_x$ can be well recovered from $\boldsymbol{h}_x$ with a high probability through the following ℓ_1-minimization program:

$$\hat{\boldsymbol{\theta}}_x = \arg\min \|\boldsymbol{\theta}_x\|_1, \;\; s.t. \;\; \boldsymbol{h}_x = \mathbf{Q}_x \boldsymbol{\theta}_x + \varepsilon'_x \tag{3.25}$$

The exposed previous formulations represent the recognition of the input trace based on the x-axis data only. $\hat{\boldsymbol{\theta}}_y$ and $\hat{\boldsymbol{\theta}}_z$ are solved in the same way.

In order to recognize the input gesture trace making use of all the three axis values, the $\hat{\boldsymbol{\theta}}_x$, $\hat{\boldsymbol{\theta}}_y$, $\hat{\boldsymbol{\theta}}_z$ vectors are combined together as expressed:

$$\hat{\boldsymbol{\theta}}_{eq} = \hat{\boldsymbol{\theta}}_x^2 + \hat{\boldsymbol{\theta}}_y^2 + \hat{\boldsymbol{\theta}}_z^2 \tag{3.26}$$

The input gesture is finally recognized by association with the gesture the trace $\mathbf{R}_i$ belongs such that $\hat{\boldsymbol{\theta}}_{eq}(i)$ is maximum.

3.5 Architecture and Challenges

The gesture recognizer system developed in the scope of this thesis is, as previously mentioned, much based in the work by Akl et al. [AFV11] and it is targeted for the iOS system. Nonetheless, besides the primary goal of gesture recognition, it was built with the thought that it is a project which can be improved in the future by other developers. Thus, every component in the framework is well decoupled, in order to easily allow

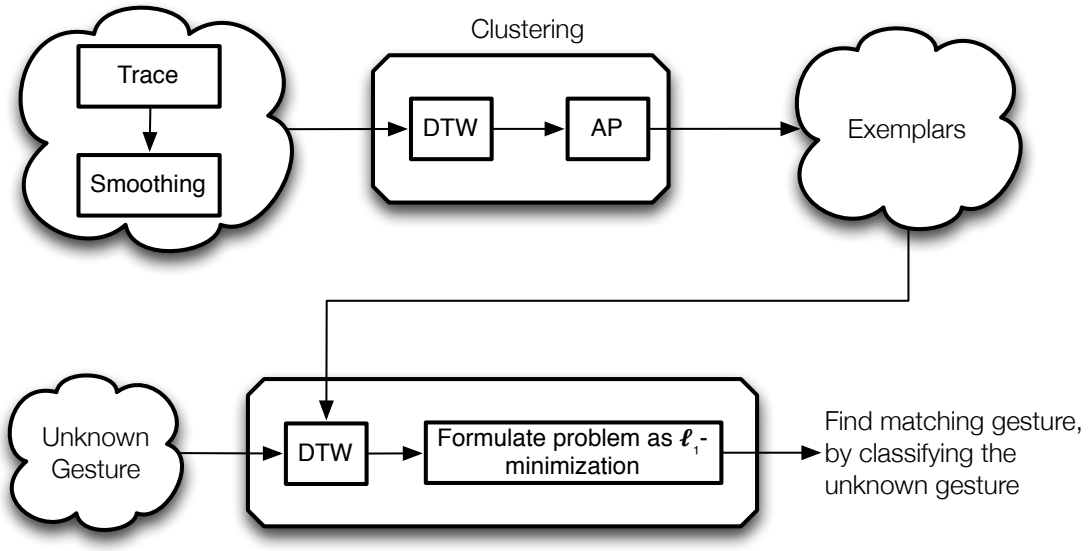


Figure 3.3: Overall gesture recognizing system architecture. Original in [AFV11].

any modifications on the algorithms and technologies used.

Regarding the gesture data acquisition, all this logic is concentrated in the `AccelerometerUtils` class. This class is responsible for reading a gesture into a structure, and if such setting is enabled, to smooth the data it gathered.

In terms of accessing the data throughout the framework, such is possible by the `DataPersistenceMediator` class. Moreover, this class is responsible for data persistence and management. By having this abstraction it is easily possible for any developer to store its dictionary or gestures or even, for example, by downloading the gestures from the web and loading them into the framework to enable its use. Regarding the persistence of data in the device, such is done by making use of the `NSKeyedArchiver` class.

Regarding the use of the framework for gesture recognition as a library, such is quite easy, as it was initially intended. The developer only needs to link the project into its own and import the class `MotionGestureRecognizer`. Such class is all the programmer needs to use and can be both empty initialized as it can be initialized with a dictionary of gestures, that needs to be stored in the application as a resource. In the latter case, the programmer just needs to access this resource and pass it to the framework. Regarding the recognition of gestures itself, this class provides three methods: `startReadingTrace`, `stopReadingTrace` and, lastly, `recognizeTrace` which returns the most similar gesture to the trace that was recorded between the two first calls.

The main issues in the construction of this framework were related with the fact that

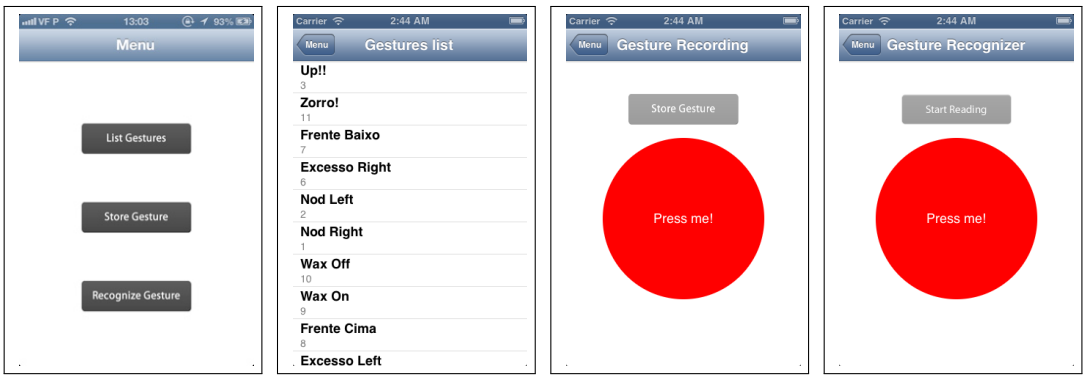
much of the technologies and algorithms used by the gesture recognition system are implemented to MathWorks® MATLAB®. Such happens because MATLAB® is a signal processing, engineering oriented programing language. So, code which is easy to use in such environment, like producing the orthonormal of matrices, which in the end, enable an easy implementation of technologies like random projection or ℓ_1-minimization, among others, is not so straightforward in a mobile environment, such as iOS. Therefore, the use of external libraries, such as `OpenCV` and `KL1p`, the use of auxiliary classes such as the `MatrixUtils` and `TraceRecognitionerMediator`, were needed for abstracting the recognition system of matters like matrix related computations.

A practical example of this challenge is the trace clustering. To cluster the traces, the approach was not to take a local solving method, for two reasons. First, the affinity propagation clustering method that is proposed to be used [FD07] is not implemented in a language that would run in the system for which this thesis is targeted (C, C++, Objective-C). Its development in a native language for iOS would be somewhat out of the scope of this thesis and would take much time from our work. Second, even if such was to be implemented, these kind of algorithms are computationally costly, thus not being particular indicated to execute in devices such as smartphones. Therefore, a server-side approach was used, even though the disadvantages that are inherited from such choice. Thus, when the traces are to be clustered and after the DTW process has been executed, the clustering data that is required by the algorithm is translated to a client-server common language and sent through the web. The algorithm executes in a remote server-side and, afterward, the clustering results are returned to the device, which is acting as a client.

Furthermore, unlike it was initially pretended, the use of FastDTW (see Section 2.4.1.2) was not possible since its implementation wasn't available for the programmig languages used in this thesis. Therefore, DTW algorithm was used. Nonetheless, the results wouldn't be much affected once the average length of the gesture traces time series is not greater than 200, as shown further in the framework tests. Therefore, according to Fig. 2.3, the execution time for computing the cost between two different signals of that magnitude is almost the same.

3.6 Developer Environment

To help the development and testing of the framework an interface, that may also be used by other developers, was built. These screens, displayed in Fig. 3.4, represent the user interface that may be used to see which gestures are stored in the system, to store new gestures and to recognize a new one.



(a) Initial Menu (b) Gestures List (c) Gesture Recording (d) Gesture Recognizing

Figure 3.4: The multiple existing screens in the developer environment.

Fig. 3.4(a), which is the application home screen, allows a user to move to three other screens. Fig. 3.4(b) is simple list that shows the existing gestures in the system, by name and id.

To store a new gesture, the Fig. 3.4(c) shows the screen where such is performed. This area consists in two simple buttons. The user must press the red button to start reading a new gesture trace and hold it until the gesture is performed. Such action can be performed M times, each for a new trace. Furthermore, the gesture trace data is smoothed after its recording has finished. Note that the *Store Gesture* is disabled until the user performs at least 3 gesture traces. When finally the user intents to store that new gesture, and if the *Store Gesture* button is enabled, the user must press such button and all the traces that were previously recorded are clustered and the exemplars are chosen, as explained in Section 3.3.

On the other hand, to recognize a newly performed gesture, Fig. 3.4(d) shows the correspondent screen. This area, also with only two buttons is similar to the area represented in the Fig. 3.4(c). To perform the gesture that will be recognized, the user must press and hold the red button to record a trace. After that, the *Start Recognizing* button will be enabled and the user can press it in order to recognize the trace that was input. In the end, if any gesture was recognized, the system outputs the gesture id and name.

Regarding the screens for gesture recording and recognizing, in a previous version, instead of the red button, the user needed to press a button twice (the first time for start recording a trace and a second time for stopping). However, such wasn't viable for many gestures. For example, if the user wished to perform a gesture in which, by the end of the recording, the device screen was not facing the user, the second button press would be very difficult for the user. Thus, by just releasing the button, the task of

signaling the system to stop recording a gesture would be much easier.

3.7 Gestures Dictionary

In order to develop the framework, test it and also to test the proof of concept, a dictionary of 12 gestures has been created, as shown in Fig. 3.5. The defined gestures range from simple device nods to right or left, to more complex gestures, such as the letter Z. Regarding average gesture trace lengths, it ranges approximately between 60 and 150 units of measurement. Although in the figure it is shown, for each gesture, it's ideal trajectory, for each trace it may be subject to changes depending on the user performance. These gestures are span two planes, ZY and XY. They were chosen as gestures that would be easy to learn by the users and to execute. Moreover, part of these gestures are also used in the game developed as a proof of concept, associated with different actions, particularly the gestures with id 1, 2, 3 and 4, displayed in Fig. 3.5.

3.8 Implementation

Most of the framework is implemented in Objective-C language, using Automatic Reference Counting (ARC). In this work some algorithms code, like DTW and Fourier Transform, were not implemented but yet they were – under GNU License – downloaded from the internet or were already part of the iOS system. All the implementation details won't be described in this section since such would be unsuitable for this document. However, it is necessary that the most relevant components are presented and explained here, since the focus of this thesis is the framework and one of its main goals is that it is easily usable by other developers.

As largely in the C environment, much of the classes are split among two files: an header (ClassName.h) which contains the method declarations, protocols and global variables, and an implementation (ClassName.m or ClassName.mm in the case this class does not contain only Objective-C code). The latter contains the method implementations.

Table 3.1 summarizes the most important classes, being stated what is their use in the framework. The description of the classes start by those which act as models, for representing elements in the system like a gesture. All these model classes implement the `NSCoding` protocol, that is needed to encode and decode classes. Such is used so the gestures, traces and exemplars can be stored persistently in the device. Also, in Table 3.1, this classes are proceeded by the suffix **Model**. Afterward, classes that are used as mediator (for algorithms, data saving, among other things) are described and

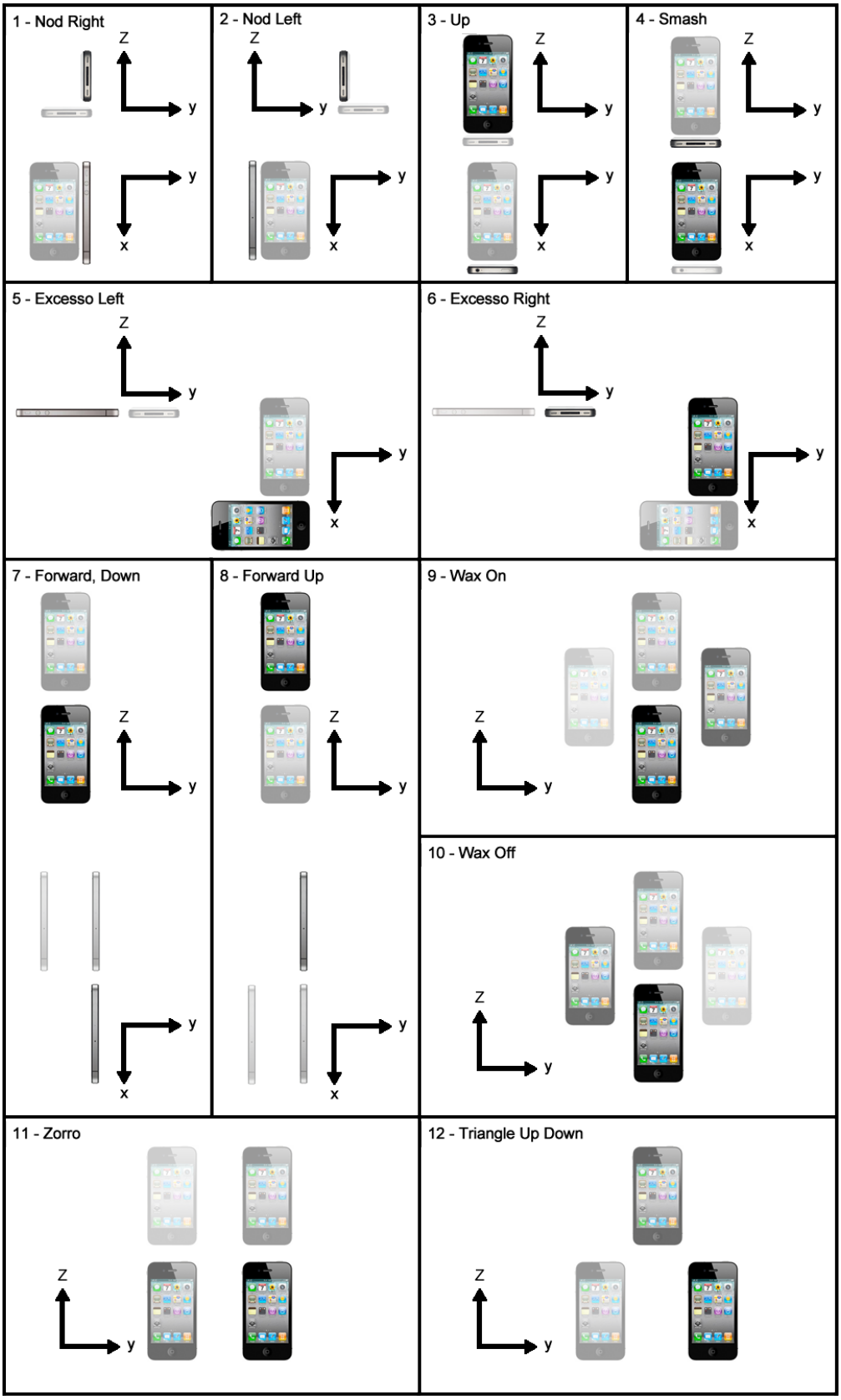


Figure 3.5: Framework gestures dictionary.

all of them are proceeded by the suffix **Mediator**. Last, but not least, the classes that are used to save and recognize gestures are addressed.

Besides the use of custom classes as described in Table 3.1, other frameworks have been used, both from the iOS system as well from third-party developers. From iOS system, among others, the `Accelerate` framework is used for the Fourier Transform and the `CoreMotion` framework is used to read the device's acceleration. External frameworks like `OpenCV`, to work with matrices, and `KL1p`, for the ℓ_1-minimization, were used and have been of great value in this work.

Regarding the server-side environment, responsible for the implementation of the affinity propagation algorithm, a ASP .NET MVC3 project using C# was built. This project, consisting alone in one class, provides a wrapper to the MATLAB algorithm implementation, that is called when handling a client request.

3.9 Implementation Results

In order to compare the implementation made in this work with other existing system it needs to be tested. Moreover, to check whether this system in viable in a real-world, gaming, context, the gesture dictionary previously mentioned (please see Section 3.7) was used.

3.9.1 Methodology

To record the gesture traces, an iPhone 4S built-in 3-axis accelerometer was used. Tests took place in a controlled room and all the 6 participants (3 males, 3 females) were already familiar with the use of smartphones. The tests consisted in 12 iterations, one for each gesture defined in the dictionary and consisting in adding the gesture into the system's dictionary and then trying to recognize it.

Regarding the training phase, before inserting a new gesture into the system, all the participants were taught how to perform it. Then, 5 of them repeated the gesture 3 times, resulting in a total of 15 traces per gesture and a total of 180 stored in the system, by the end of the tests.

Regarding the recognition phase, all the participants were required to perform the gesture 2 times. Moreover, for each iteration, participants needed to perform the already tested gestures again. Thus, there was a total of 936 tested traces.

Class	Description
GestureModel	This class is responsible for representing a Gesture. It contains a `gestureId`, that is used to identify different gestures; a `name`, which is helpful for the developer/user to know which gesture is and, lastly, a set of clusters.
ClusterModel	This class represents a Cluster. Is contains a `clusterId`, a set of traces and also the cluster exemplar.
TraceModel	This class is used for representing a gesture trace. Moreover, it is possible to know to which gesture or cluster it belongs once it contains the corresponding id's.
RTraceModel	This class is used as an auxiliary model to represent a trace that will be subject to a Fourier transform. It contains the Fourier transform of the trace and also its sparsity level.
AccelerometerUtils	This class is responsible for collecting the device's accelerometer data. Moreover, if a trace data is to be smoothed, such is also made here, calling the `MovingAverage` class.
MovingAverage	This class has the capability of smoothing the values of a sequence.
Utils	This class is a wrapper for utility and general methods or algorithms, such as generating an uniformly distributed value.
MatrixUtils	This is an utility class for manipulating and working with matrices.
DataPersistanceMediator	This class is used for both persistence of data in the device, such as creating, loading and saving, but also for neatly accessing the available data, like different clusters or traces.
TraceRecognitionerMediator	This mediator is to abstract the `DataTesting` class of working much with matrices. It is responsible for much of the random projection related computations, such as generating the random projection matrix **A** or the projected matrices.
ClusteringMediator	This class is the interface between the `DataTraining` class and the server-side where the clustering, through affinity propagation, is done. It is responsible for translating the traces into a client-server known language, requesting the server to execute the algorithm with the input data and to returned the result back again to `DataTraining` class.
DTWMediator	This class is the interface between the DTW algorithm and the framework. It receives two traces as input and computes it similarity cost, as is formulated in Eq. 3.2.
DataTraining	This class is responsible for receiving a set of traces and clustering them, generating a gesture.
DataTesting	This class is responsible for finding the most similar gesture to a trace that is given as input
Constants	In this class, that only consists in an header and not an implementation, is defined all the data of the framework, such as the IP of the server which is running the clustering algorithm as other algorithms parameters.

Table 3.1: Framework's implementation classes.

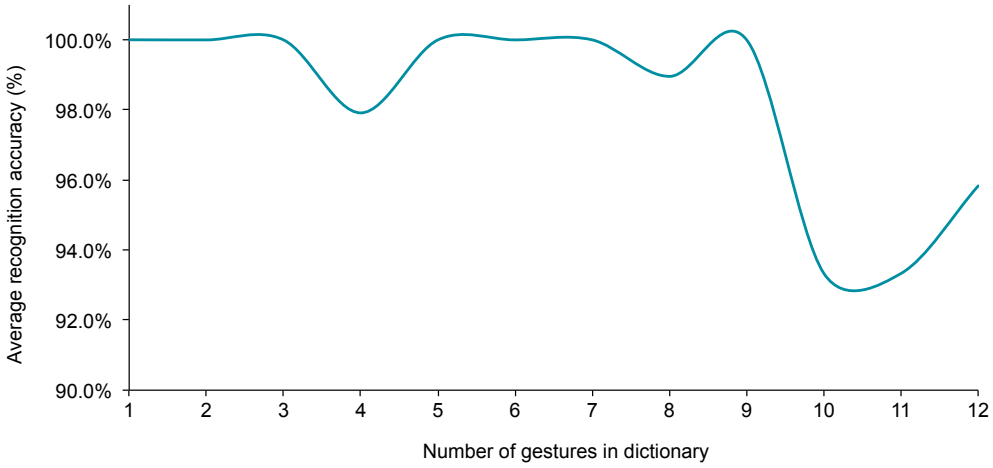


Figure 3.6: Average gesture recognition accuracy for the number of stored gestures.

3.9.2 Results

Regarding the results, they can largely suffer the impact of a simple device tilting, since the values collected by the accelerometer are different given such action. In Fig. 3.6, it is possible to see that the gesture recognition accuracy was affected by such event, once the average value is smaller with 4 gestures in the dictionary than with 6. Nonetheless, it's possible to observe that with the increase of the total number of gestures, the recognition accuracy was affected, although is was never lower than 93.3%.

In terms of gesture recognition average time, as shown in Fig. 3.7, the results show that this value was much affected as the number of gestures in the system's dictionary increased. For example, for the gesture with the id 1, this value ranged between 0.08s with one gesture in the dictionary up to 1.07s, with 12 gestures in the dictionary. Unfortunately, this figure doesn't allow to check how fast would the last added gestures take to be recognized, once they were already affected by the existing gestures in the system. Tests should have been done taking such event into account. However, besides the individual time taken for each gesture, the average value is seen to increase also, influenced by the fact that the last gestures were longer than the first. Comparing with the system developed by Akl et. al [AFV11], this system is slower. However, such is probably related with the fact that the tests conducted in their work were done with a computer, whilst ours have been made with an iPhone, which does not have such a competitive computing capacity.

Fig. 3.8 shows that a connection exists between the gesture lengths and the time

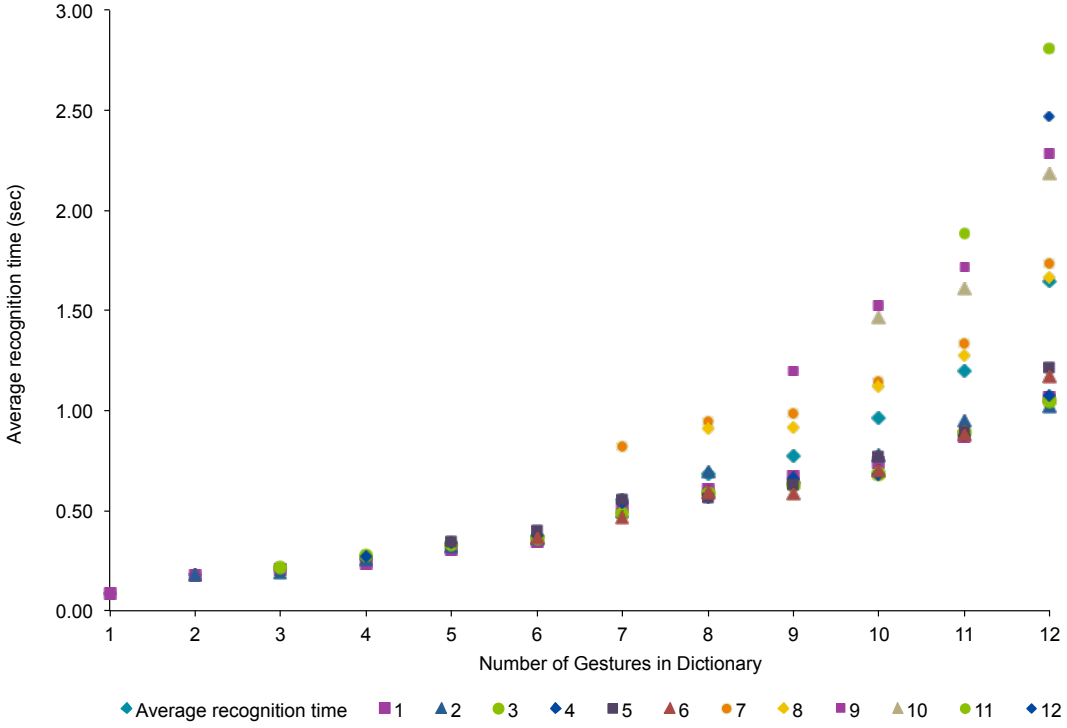


Figure 3.7: Average recognition time for gesture in each iteration.

Figure 3.8: Average gesture recognition time and average gesture traces length.

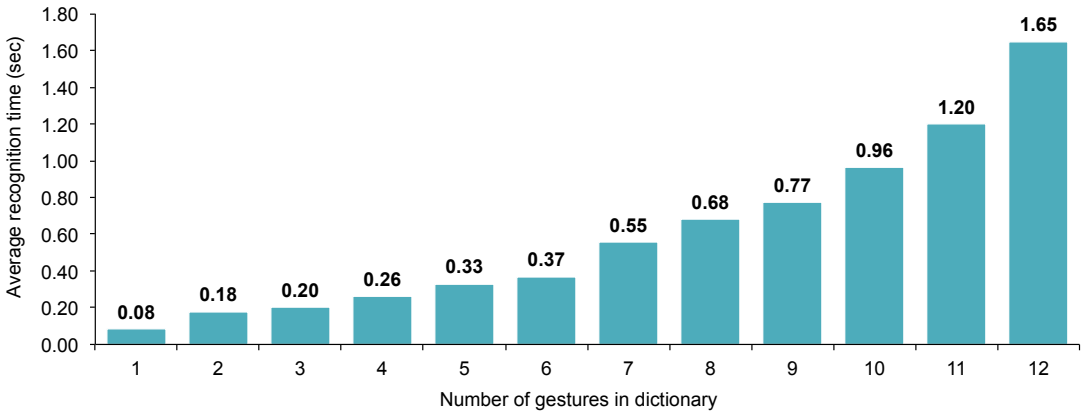


Figure 3.9: Average gesture recognition time for the number of stored gestures.

Technique	no. of gestures	Gesture Recognition Accuracy (%)		
		User-Dependent	Mixed-User	User-Independent
Implemented System	1 - 12	100 - 95.83	100 - 95.8	100 - 90.9
Akl et al.'s System	8 - 18	100 - 99.81	99.98 - 98.71	96.84 - 94.60
HMMs	8 - 18	99.97 - 99.54	99.61 - 98.11	75.96 - 71.50
uWave	8	98.6	-	75.4
System in [SPHB08]	5	89.7	89.7	-

Table 3.2: Comparison of Performance of the implemented system, the original system propoed by Akl et al., HMMs, uWave and system in [SPHB08].

it takes to recognize them. Moreover, the same figures sustains the aforementioned fact that the tested traces are not ever longer than 200 and so, the system is not much affected by not using FastDTW. Moreover, it's possible to see that the average recognition time is influenced by the fact that more gestures exist in the system.

Fig. 3.9 demonstrates once again that the results were affected by the insertion of more gestures into the system dictionary, by showing the average recognition time in each interaction for all the stored gestures.

Table 3.2 shows the comparison between the system that was implemented in this thesis and other similar systems. Noteworthy is the analysis with the original system built by Akl et al. [AFV11]: the results show that our implementation has an overall worst gesture recognition accuracy, which suggest that the tests may have been compromised or the implementation is not correct to its full extent. Nonetheless, the collected values are still quite competitive, in comparison to similar systems.

The overall picture shows that, for the implemented system to be used in a gaming

context, no more than 7 gestures would be viable, considering these gestures average length would be in the interval of 50 up to 150. Although the system is capable of correctly identifying such gestures with a high accuracy recognition, such is not made in time for a comfortable game experience, once taking more than 0.5s to convert a user action into a game feedback wouldn't be suitable.

4

Proof of Concept

In order to test the developed framework for gesture recognition, a proof of concept was required. Moreover, this thesis is in the scope of DEAP project and the developed framework may be used in the future to help building educational games that make users aware of environmental sustainability problems. Such games also explore the interaction between users making use of mobile devices and public ambient displays, in order to persuade them into adopting pro-environmental behaviors.

Therefore, the best way to achieve real results was by developing a similar game, in this particular environment. Accordingly, the chosen proof of concept for this thesis was an educational game that focus environmental sustainability problems and explores the interaction between mobile devices and ambient displays, also having in mind the persuasion of the user towards the adoption of pro-environmental behaviors. Multiple environmental areas were addressed in making the mockups of the game, like energy, recycling and pollution.

4.1 Proposed Solution

Regarding the game that was developed as a proof of concept for the framework, it is a two-player game which promotes the interaction between mobile devices and public ambient displays and that mainly addresses environmental questions.

Since the game is played in a public environment, where strangers can gather around and play for a short time, it must not take long to be finished.

Therefore, it is based upon the concept *on-rails*, in which the game guides the player, allowing him to make some choices but never deviating from the expected path, much like a railway guides a train. This way the participants only makes some actions when it is required, facilitating the development of the game, where the motion interaction is focused.

It consists in three stages, each one with different challenges. In every stage, some sectors that most contribute to excessive consumption and waste of natural resources are focused and the user must choose the correct options in order to better preserve the environment.

For example, consider a stage that consists in a set of moving lighting bulbs, in which some are compact fluorescent and others are incandescent. So, each time an incandescent light bulb is in the middle of the participant's public ambient display zone, he must make a gesture with his iPhone that represents the smashing of the bulb. If this movement is made in the exact time he will be awarded some points. This is the example of the first stage, but multiple others can be designed with a similar mind-set.

After the players have been through all stages, the one who has more points will win the game.

Also, and because the participants won't probably be familiar with the game, each stage is preceded by as introductory text explaining what gestures the participant must perform and in which cases.

As the framework is deployed for iOS system, the game was also developed for the same system.

The framework was used in the development of the game, abstracting the recognition of the gestures that are required. Regarding this subject, the proof of concept only contains code for a generic button to flag the framework to start recording a trace and stop recording it and also for associating a gesture id to an action. Besides this code, the gestures dictionary must be included as a file resource in the game and such was created through the use of the developer interface (please see Section 3.6).

For the game development the framework Cocos2d for iPhone was used. This framework allows building 2D games and graphical applications in a more simple way than working directly with OpenGL, making it easier to develop the proof of concept.

Furthermore, the game is shown in a public ambient display that is connected to the Apple TV. Also, the screen is splitted in two, as most of the two-player games, in order that both participants can play it in parallel.

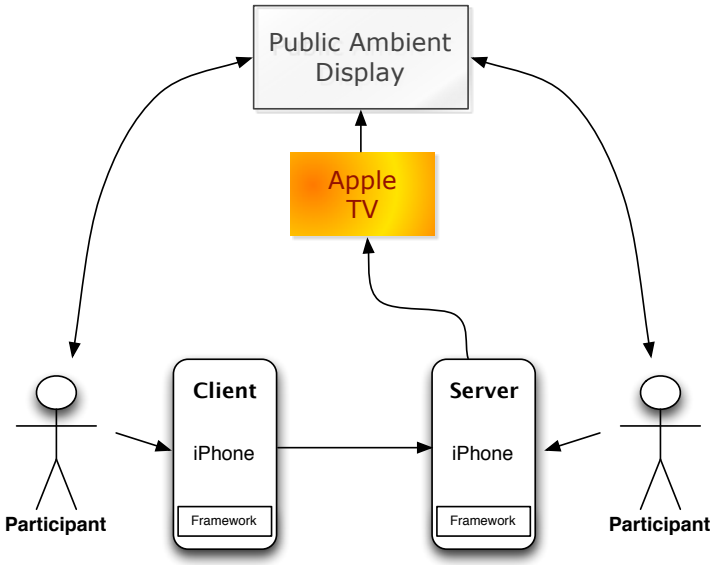


Figure 4.1: Game architecture overview.

4.2 Architecture

As the proof of concept must be integrated with the framework, it is developed and targeted for the same system. Thus, it was developed for iPhone devices.

To facilitate the whole proof of concept development, since this is not the main focus of this thesis, multiple choices aiming its construction simplicity were made, relying on already existing frameworks and simple game effects and interactions.

The overall architecture can be seen in Fig. 4.1, in which is shown the required setup: two iPhone devices, once the game is played in a one-to-one basis, an Apple TV and a Public Ambient Display.

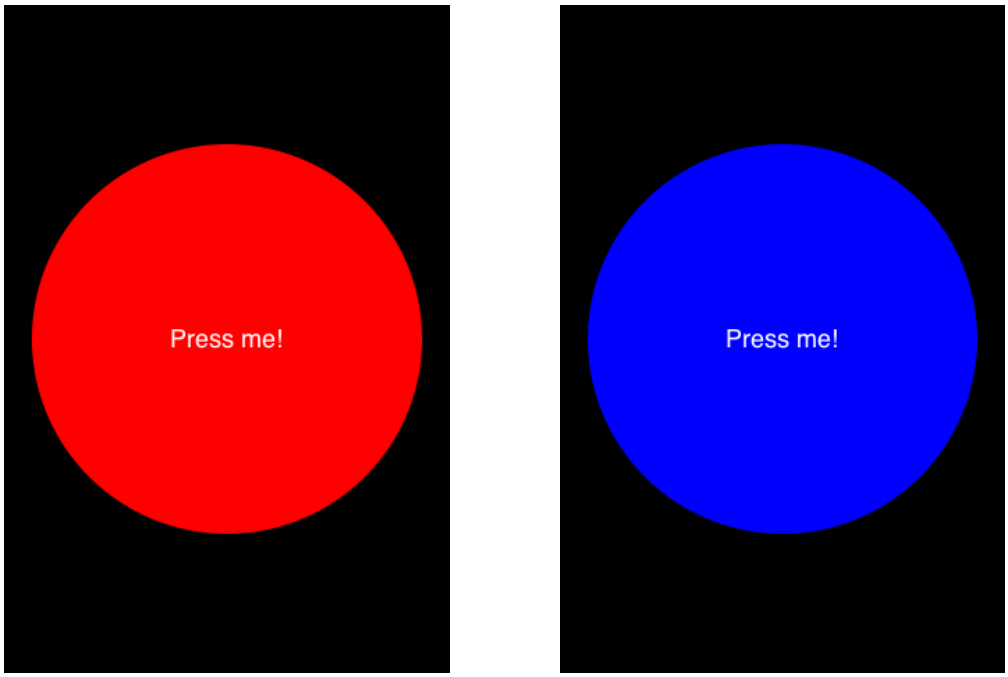
In order for the participants to play, both devices must contain the developed game. Moreover, the application allows the connection between the two players where it can act both as a client or as a server, being that the participant who creates a new game room will act as a server and the other participant who joins an already existing game will act as a client. Therefore, it is also required that both the devices and the Apple TV are in the same local network.

When a participant creates a new game room, the application, acting as a server, will advertise this room in the local network to which all the devices are connected. Afterward it waits for any other participant to join. Only when another participant joins the room, the game will take place; until that moment the participant, acting as a server, only sees a loading screen.

In order to check for existing game rooms, for creating new ones and also for the client/server interaction, the Apple Bonjour protocol was used. Moreover, this protocol reduces existing synchronization problems, allowing a faster development for this type of architecture, once it abstracts the developer from low-network issues.

After a participant joins a room and the game starts, both the server and client devices will show a black screen with a button, in order that the participant perceives that the device is to be used in a similar way to a remote. As seen in Fig. 4.2, this button is red, if the participant created the game, or blue, if the participant joined an already existing game that is waiting for participants.

A disadvantage in this kind of architecture is that the participant who acts as a client has a minor delay comparatively to the participant who is acting as server, since it must send the gesture it was interpreted in his side to the server, through the local network, with all its associated computational issues. In order to reduce such delay, the minimal required information is sent through the network.



(a) Player 1 screen (server) (b) Player 2 screen (client)

Figure 4.2: Device screen during the game.

For the participant to execute gestures during the game, all the device screen is designed taking into account which areas are more reachable by the users thumb. So, whenever the participant intents to perform a new gesture he presses and holds the

only existing big button in the screen, until the gesture is fully performed. As soon as the user moves the thumb away from the button, releasing it, the system tries to recognize the gesture and sends the result to the server, that performs any action on the client screen zone according to the recognized gesture and the game level. Such screen is different between the player 1 and player 2 so it is easy for the players to identify which is the correspondent game area, as shown in Fig 4.2. Moreover, in each game area in the game interface, there is a similar filled circle with the same color that is in the user device, as visible in Fig. 4.3(b).

In order to read and recognize a gesture trace, the game uses methods available in the gesture recognition system framework, that is used as a library in the game project. For the game developer, only three calls to the framework are necessary in order to read and recognize a gesture trace.

While the client application only has the task of reading, interpreting and sending the gesture to the server, the latter must both act as a client and as a server. This happens because, besides reading and interpreting gestures, it must receive the executed gestures by the client and further, it must execute the two-player game and stream it to the external display. Thus, even though the game is displayed in an external public display, so that both participants and also the general public may watch it, the game itself runs in the server device, being streamed to the screen. Such is only possible in iPhone's 4s and more recent iPhone devices, by mirroring the screen to an Apple TV. Moreover, to enable such feature, the server device is connected to an Apple TV, that must be in the same local network.

4.3 Gameplay

Although the main idea of the proof of concept is to be a practical test to the developed framework, it also serves the purpose of acting as an educational game with the goal of persuading its players and the surrounding audience towards the adoption of pro-environmental behaviors. Therefore, all of the game levels were built with this premise in mind, besides making the use of gestures. Moreover, the game interaction between different levels is also done making use of a gesture interface, with the users having to make a gesture to proceed.

Even though the participants may learn how the game works by watching fellow participants playing, this may not suffice. Thus, to help participants understand how to play the game, the first screen explains what is the game and what the player is required to do. Moreover, each level is preceded by an introductory text where it is explained what the participant must do in order to maximize his points. Here, it's explained what

kind of gestures the participant must execute and when is the correct time for its execution. Whenever the participants wish to move away from the instructions and start playing the game level, one of the players can execute the *Smash* (id 4 in 3.5) gesture. Furthermore, all the gestures are of simple execution once the players won't have much time for practicing them.

To increase the participants interest in the game and also to motivate them into executing the correct gestures on the correct time, thus learning which pro-environmental behaviors are correct, a score system is used. This way, the player who performed the most correct actions is rewarded with more points. This results in a competition between both players in which the most valuable player wins the game.

The different game levels focus on multiple environmental issues and all of them have an approximate duration of 45 seconds.

In the first level, the user must be able to know different kind of light bulbs and which of them are energy expensive and which are not. Also, those that are not energy efficient must be destroyed, thus associating a bad stigma to this type of bulbs. In this level, multiple objects fall from the top of the screen, one by one, in random x positions. There are two type of objects that can randomly fall: incandescent light bulbs, which are energy inefficient and fluorescent light bulbs, which are energy efficient. Every time an energy inefficient appear in the player screen, the player must destroy it by using the gesture *Smash*, which is similar to smashing something with the device. However, if this gesture is performed while an efficient light bulb is on the screen, the player will lose points. Any other gesture that the user performs will not have influence in the bulbs or the level score.

In the second level, the user must be able to sort different recycle objects among the three different recycling bins: paper, plastic and glass. Similarly to the first level, there are also multiple random objects falling from the top of the player screen. However, they always appears horizontally centered in the screen. Depending on the type of object, which can be paper, plastic or glass, the user must choose the trajectory of the falling item, by changing its x position, so that it falls on the correct recycling bin position. It is possible to change it by performing two simple gestures, *Nod Left* (id 2 in 3.5) and *Nod Right* (id 1 in 3.5). Moreover, the player must perform the *Smash* gesture if he pretends that the falling object goes inside the recycling bin placed in the middle of the screen. Every time the user inserts an object in the correct recycling bin, he his awarded with a positive score. On the other side, if an object falls in the wrong bin, the user will receive negative points.

In the third and last level, the user is suppose to distinguish between biodegradable and non-biodegradable objects. In this level the user watches a moving cartoon sprite,

simulating a character man who is walking in a street, and multiple random objects than can be biodegradable and non-biodegradable appear from the direction the cartoon is moving to, in random y positions. Also, only one object appears in the screen player at each time. Whenever a non-biodegradable object appears near the walking cartoon, the user must perform the *Up!* (id 3 in 3.5) gesture, simulating that he is picking up an object from the ground. Each time the user picks up a biodegradable object he will receive negative points whereas if he picks a non-biodegradable object, positive points will be awarded.

In the end of the third level, the game moves to a final screen where the individual level scores are shown for each player and also their total points. Moreover, it is presented which player was the winner of the game.

After the participants watch who was the winner, the game life-cycle is over.

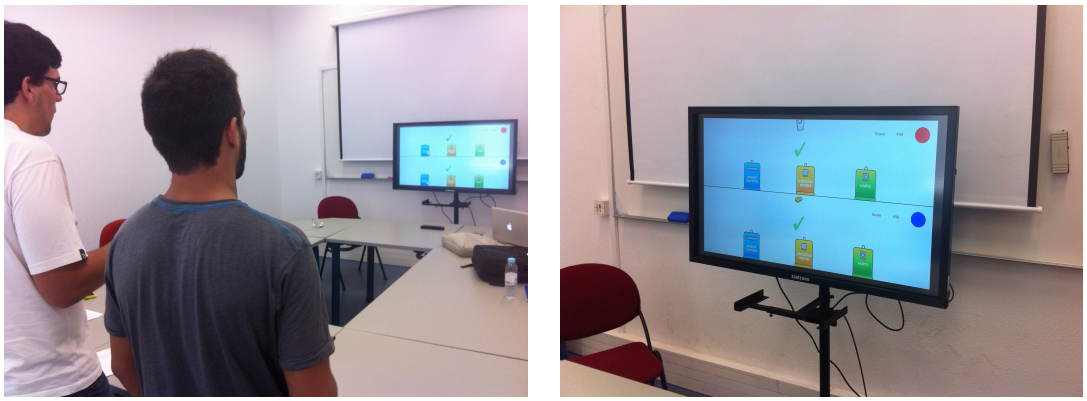
4.4 Usability Testing

Usability tests are of much importance when it comes to evaluate a product by letting users perform specific actions or scenarios. The goal consists in receiving direct input on how users interact with the system, by observing them and their actions. At the same time, discover errors and areas of improvement that haven't been detected during the development stage, such as controlled environment tests. The results of these tests ought to be treated as a baseline for all the succeeding, in order to use them as a comparison model, evaluating the evolution of the system.

4.4.1 Methodology

The tests took place at the IT department, in the FCT-UNL campus, where all the required equipment and structure needed to execute the game was available. The projection screen was a 42-inch LED TV and the game information displayed on it was fair at approximately 3-5 meters. Although the game is intended to create a honey-pot effect when displayed in a public environment, the tests were performed in a closed environment, in which every new participants would enter the room in pairs, while the others would be outside the room. Figures 4.3 shows the environment of these tests.

Each time a new pair would start the game, it was explained to them how the game worked and how would their interaction with the system be. Moreover, players were taught how to use the device to perform a gesture and how to perform each of the different gestures, focusing the fact that they should press and hold the button until gesture was complete. Along with this personal explanation, the game itself had screens telling



(a) Player perspective of the display (b) Detail of the game interface and feedback

Figure 4.3: Participants playing the game.

what the user was suppose to do in the game and, specifically, in each level, as previously mentioned. Moreover, at the beginning of each game, in the general explanation screen, each participant had the opportunity to tests 5 gestures and only then the level 1 introduction would be shown.

The devices used in the tests were an iPhone 4S, which played the server role and was used by player 1, and an iPhone 4, which acted as the client and was used by player 2.

To allow the communication between the different devices, a local network was created in order to perform the tests, connecting all the devices and the Apple TV.

At the end of each session, participants were asked to answer a questionnaire with the purpose of not only evaluate the game usability but also to evaluate the use of a device a gesture controller. Moreover, it was also intended to evaluate the users opinion about environmental issues, such as recycling and energy consumption.

The questionnaire used for the usability tests, shown in Appendix A, is based on an article by Lund [Lun01], which presents a tool, the USE questionnaire, that helps to evaluate if an interface is well designed and which problems should take priority. USE stands for Usefulness, Satisfaction and Ease of use. The idea is to ask users to rate their agreement with different statements. These agreements range from strongly disagree to strongly agree and they are constructed as seven-point Likert-type scales. In this work, participants were asked to rate statements, using a five-point Likert-type scale ranging from strongly disagree to strongly agree.

Its purpose was not only to evaluate the game usability but also to evaluate the use of a device a gesture controller.

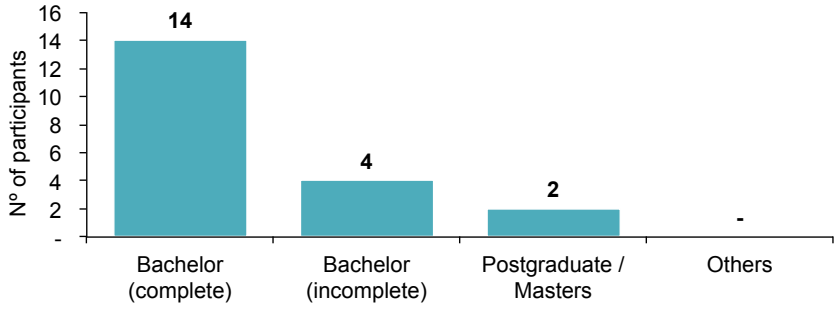


Figure 4.4: Academic Degree.

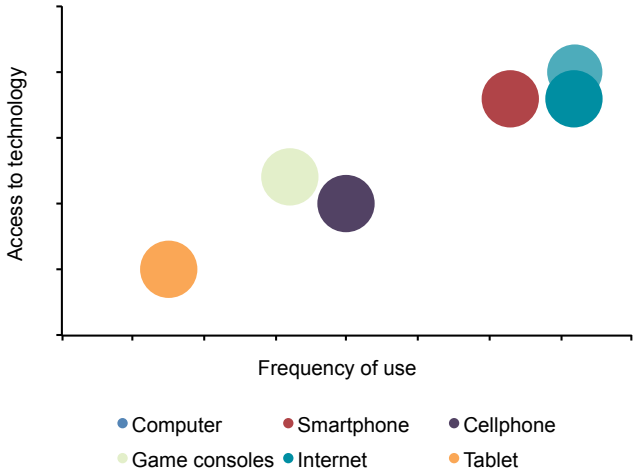


Figure 4.5: Participants' use of new technologies.

The first section of the questionnaire focuses on the user information and the second in the user background regarding new technologies and environmental issues. The third section focuses on the game usability and on the use of gestures as an interaction mechanism.

4.4.2 Discussion and Results

A set of 20 participants tested the game and answered the questionnaire with ages ranging between 20-65 years old, with an average of 24.75. Moreover, participants were from both genders, being 80% of the male gender.

Most of the participants, fourteen, had already completed their Bachelor, while four

were still completing it. Two users had already completed their Masters, as seen in Fig. 4.4. Regarding their familiarization with new technologies, which is shown in Fig. 4.5, most of them have computers, smartphones and internet, and use them in a daily basis. The least accessed and least used technology is the Tablet, while Game Consoles and Cellphones show an average use and access. It is also possible to see that those technologies that most of the users have are also those who are most frequently used, while the opposite also happens.

Shown in Fig. Fig. 4.6, in terms of users agreeing to being aware of environmental issues, most of them agree or strongly agree that are aware. Based on these results, it is possible to conclude that most of the users were already aware of environmental issues before they played the game.

In terms of questions that address the game itself, Figures 4.7, 4.8 and 4.9 show the average rate of agreements but also the average rate conditioned by the fact that the user won or lost the game. Such conditionality allows to compare the different users answers considering the fact that they succeeded better or worst than their opponent.

Fig. 4.7 shows that most of the participants consider that the game was easy to play and that this result is raised for the participants who won the game. Moreover, almost all participants strongly agree that it was easy to learn how to play the game and that they liked to play the game.

Regarding the game usability, Fig. 4.8, the results show that almost every users forgot to release the button for performing the gesture. However, we strongly believe that these results were affected by the fact that this answer requires a scale swap, once the best results would be close to 0. Such may have confused the participants, which in informal chats during the tests stated that they didn't have much problems of forgetting to release the button. Moreover, during the sessions, it was observed that most of the users didn't forget to release the button but, instead, they released it sooner than supposed, before the gesture was fully performed. Users, particularly those who won, consider that the game feedback when performing a gesture was useful. Also, there is an agreement that sometimes they don't know if the gesture they used was the one that was supposed to. In this point, the difference between the users than won the game and those who lost it is clearly visible, showing that this is probably one of the points that differentiates winners. Regarding gesture recognition, users show that an high accuracy has been achieved by the framework. Moreover, results show that the system is not as fast as the users would desire and there is a clear difference between those who won and those who lost. Regarding the game reaction, users who won the game state that it is worse than those who lost. Overall, the users concur that it is easy to use the device as a gesture controller and that the gestures were somewhat easy to

perform. Almost every user agrees that it was easy to learn the gestures dictionary and all of them strongly agree that it was easy to identify which player they were. This last point has shown a great advance comparatively to the game development stage, once it was very difficult for players unfamiliar with the game to recognize which was their corresponding screen.

This last point has shown a great advance comparatively the development stage, once it was one of the first issues to be detected.

Addressing the game results, most of the users strongly agree that their overall opinion about the game was good. In terms of reinforcing environmental ideas they previously had, participants agree that the game accomplished this point. However, players who won the game didn't agreed as much as the players who lost the game, regarding the fact that the game reinforced the ideas they previously had. Moreover, participants strongly agree that they understood the main ideas of the game.

Considering the overall picture, the participants feedback was quite good, showing that the users enjoyed to play the game and that it was easy not only to control it using gestures but that these were easy to perform and learn. However, it would be preferable that the system was faster in terms of recognition. Also, during the tests, it was uncovered that the gesture recognition and the game itself was directly affected by the device battery. While testing, the device that was acting as a server started running out of battery and the game became slower. Moreover, from the users feedback, the critics about the framework taking some time to recognize an object were accentuated in this period. After putting the device to charge, the game became faster as did the gesture recognition.

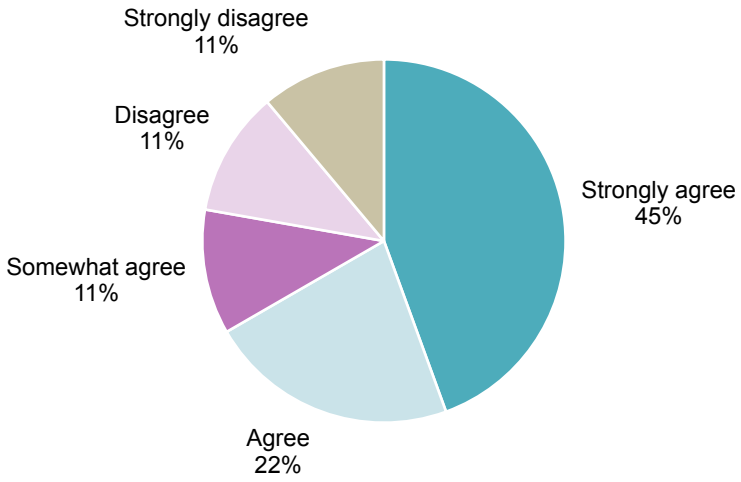


Figure 4.6: Participants agreements toward being aware of environmental issues.

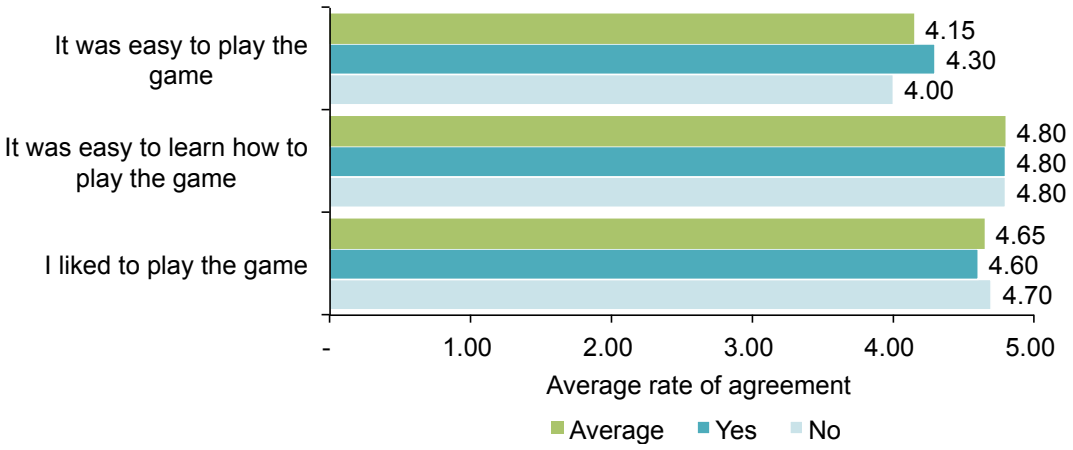


Figure 4.7: Rate of agreement towards the game experience, conditioned by if the user won the game or not.

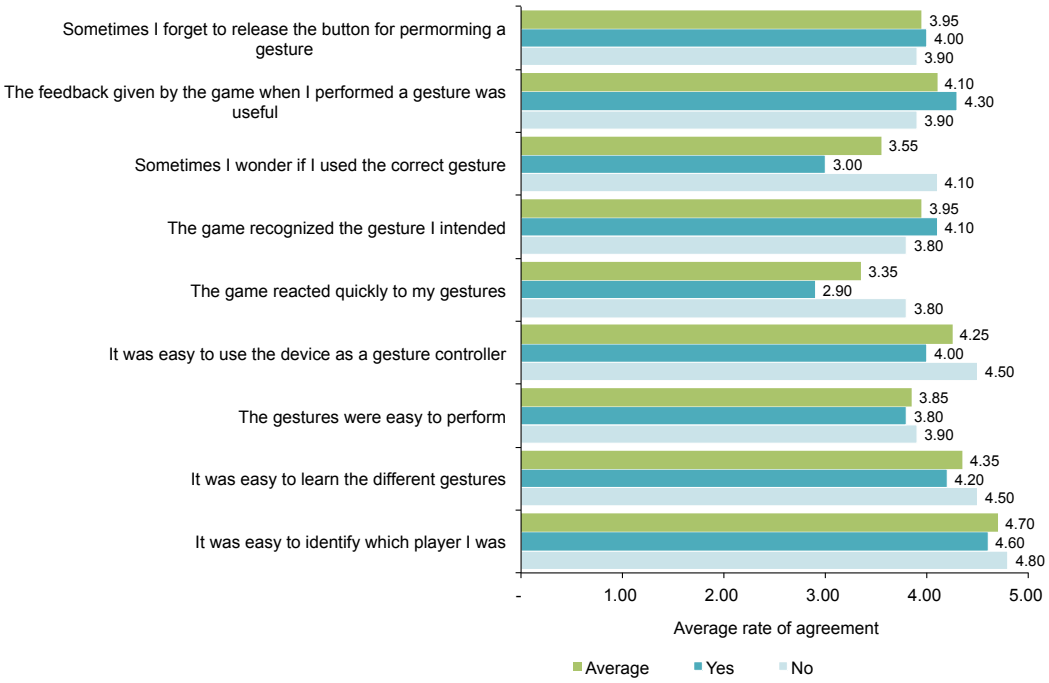


Figure 4.8: Rate of agreement towards game usability, conditioned by if the user won the game or not.

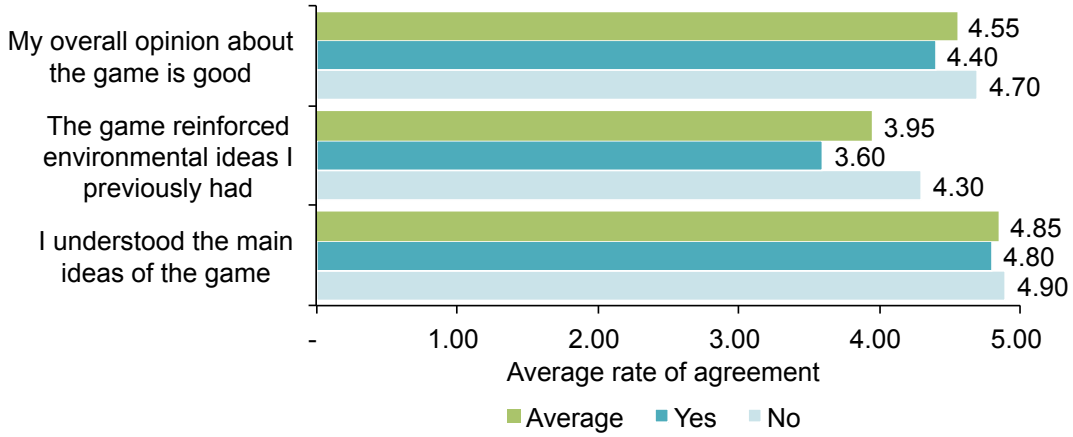


Figure 4.9: Rate of agreement towards the game results, conditioned by if the user won the game or not.

5

Conclusion and Future Work

This document presented the work done in the implementation of a system that is able to record and recognize user gestures based on motion from hand movement. Such system was developed with the end of being used as a framework by developers who wish to create motion-based applications in a swift and efficient way. Furthermore, the implemented system is targeted to the iOS system, which includes the iPhone, iPad and iPod devices. The system implementation was based in the design of a previously presented system by Akl et al. [AFV11], once it is is merely based in the data of a single 3-axis accelerometer that is built-in on such devices. Regarding its behavior, the system acts in two phases: the training phase, in which gestures are added to the system by performing multiple repetitions and the recognition phase, in which a given gesture trace is sent as input to the system and the latter tries to find any correspondent gesture stored in the system. If such is found, the matching gesture is returned. This system makes use of dynamic time warping and affinity propagation for an efficient training.

Moreover, a proof of concept for the presented framework was designed and implemented and consists on a game that makes users aware of environmental sustainability problems. Besides, it also explores the interaction between users making use of mobile devices and public ambient displays.

The system was tested with a dictionary of up to 12 gestures, containing up to 180 stored gesture traces and having tested 936 input traces. The system performance was

evaluated and compared with the original system, as well as with other similar systems. Although the comparison with the original system revealed that the achieved accuracy was inferior, these values are still very competitive, outperforming the system in [SPHB08] as well as the uWave system, even with a superior number of gestures. Moreover, the tests have shown that this specific implementation in a real-time gaming context isn't suited with a dictionary of over 7 gestures, once the average recognition time takes more than 0.55s. It is also important to mention that this value is also affected by the gesture trace lengths stored in the system, once that traces with larger lengths take more time to be recognized. However, having more than this number of gestures requires a larger memory effort from the user. In the vast majority of games it is preferable to have just a few number of different gestures, in which case the developed framework is perfectly capable of working with.

Regarding the analysis of the game that has been developed as a proof of concept for the framework, 20 participants have played it and have after answered a questionnaire. The results have shown that the users felt comfortable and enthusiastic by playing the game. Overall, users stated that the game was fast to react to their gesture and that it was fairly easy to use the device as a gesture controller. Regarding the gestures used in the game (that also make part of the framework tests dictionary), users agreed on the fact that they were easy to learn. Moreover, in terms of reinforcing environmental ideas they already had, participants agreed that the game accomplished such point.

Furthermore, the proof of concept development was quite abstracted from the gesture recognition and its gesture recognition accuracy was almost 100% in an adequate time, meaning that the framework reached its main goals.

5.1 Future Work

Future work involves having a way of incorporating gesture spotting in the framework. In this implementation, it is required signaling to start and stop recording a gesture trace. Such is done, in both the developer's interface as well as in the game controller by holding a button to signal the start and by releasing it to signal the stop. However, even though users were commonly comfortable with such procedure, it is not an ideal manner. Preferably, the system should be capable of detecting relevant gesture traces from a stream of hand movements and to recognize them accordingly. Also, in the training phase, the gesture traces clustering was done in a remote server-side, which may not be suitable if a developer does not have access to an external server running the clustering algorithm. Ideally, the system should have this algorithm implemented in order to facilitate the developers work, by integrating everything in the framework

application. Moreover, the tests shown that the system is affected by the device tilting, so, both the participants that tested the framework as those who have tested the proof of concept, were asked to avoid tilting the device. It would be ideal to transform the system in order that such event is diminished.

Bibliography

[AFV11] Ahmad Akl, Chen Feng, and Shahrokh Valaee. A novel accelerometer-based gesture recognition system. *IEEE Transactions on Signal Processing*, 59(12):6197–6205, December 2011.

[Aga05] S Agamanolis. New technologies for human connectedness. *ACM Interactions*, 12(4):33–37, 2005.

[Age05] International Energy Agency. *Key World Energy Statistics*. IEA Books, Paris, France, 2005.

[AV10] Ahmad Akl and Shahrokh Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In *IEEE International Con- terence on Acoustics Speech and Signal Processing (ICASSP 2010)*, pages 2270–2273, 2010.

[BDDW08] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.

[BM01] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[BP09] Moniruzzaman Bhuiyan and Rich Picking. Gesture-controlled user interfaces, what have we done and what's next ? In *Proceedings of the Fifth Collaborative Research Symposium on Security, E-Learning, Internet and Networking (SEIN 2009), Darmstadt, Germany*, pages 26–27, 2009.

[BTK06] Magnus Bang, Carin Torstensson, and Cecilia Katzeff. The powerhouse: A persuasive computer game designed to raise awareness of domestic energy consumption. In *Persuasive Technology*, pages 123–132. Springer, 2006.

[CCH11] Yao-Jen Chang, Shu-Fang Chen, and Jun-Da Huang. A Kinect-based system for physical rehabilitation: a pilot study for young adults with motor disabilities. *Research in developmental disabilities*, 32(6):2566–70, 2011.

[CDD07] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. *Journal of the American Mathematical Society*, 22(1):211–231, 2007.

[Cen11] Pedro Centieiro. Mobile Persuasive Interfaces for Public Ambient Displays. Master's thesis, Faculty of Science and Technology (FCT), New University of Lisbon, 2011.

[Cer11] Cerebra. Captology venn diagram. http://cerebra.co.za/image/captology-venn-diagram, 2011. Accessed: 04/01/2013.

[Cos13] Sam Costello. How Many Apps Are in the iPhone App Store. http://ipod.about.com/od/iphonesoftwareterms/qt/apps-in-app-store.htm, 2013. Accessed: 16/09/2013.

[CRT06a] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on*, 52(2):489–509, 2006.

[CRT06b] Emmanuel J Candès, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

[CT05] Emmanuel J Candès and Terence Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.

[CT06] Emmanuel J Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[CW08] Emmanuel J Candès and Michael B Wakin. An introduction to compressive sampling. *Signal Processing Magazine, IEEE*, 25(2):21–30, 2008.

[Dar06] Sarah Darby. The effectiveness of feedback on energy consumption. *A Review for DEFRA of the Literature on Metering, Billing and direct Displays*, 486:2006, 2006.

[DG99] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of the johnson-lindenstrauss lemma. *International Computer Science Institute, Technical Report*, pages 99–006, 1999.

[Eis12] Craig Eisler. Starting February 1, 2012: Use the Power of Kinect for Windows to Change the World. http://blogs.msdn.com/b/kinectforwindows/archive/2012/01/09/kinect-for-windows-commercial-program-announced.aspx, 2012. Accessed: 07/01/2013.

[FAVT09] Chen Feng, Wain Au, Shahrokh Valaee, and Zhenhui Tan. Orientation-aware indoor localization using affinity propagation and compressive sensing. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2009 3rd IEEE International Workshop on*, pages 261–264. IEEE, 2009.

[FD07] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science (New York, N.Y.)*, 315(5814):972–6, February 2007.

[FDK+09] Jon Froehlich, Tawanna Dillahunt, Predrag Klasnja, Jennifer Mankoff, Sunny Consolvo, Beverly Harrison, and James A Landay. UbiGreen: investigating a mobile tool for tracking and supporting green transportation habits. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 1043–1052, 2009.

[FM88] Peter Frankl and Hiroshi Maehara. The johnson-lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.

[Fog98] B.J. Fogg. Persuasive computers: perspectives and research directions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, number April, pages 225–232, 1998.

[Fog02] B. J. Fogg. Persuasive technology: using computers to change what we think and do. *Magazine Ubiquity*, 2002:89–120, 2002.

[HIR02] Karen Henricksen, Jadwiga Indulska, and Andry Rakotonirainy. Modeling context information in pervasive computing systems. *Pervasive Computing*, 2414:167–180, 2002.

[HN94] Robert Hecht-Nielsen. Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational intelligence: Imitating life*, pages 43–56, 1994.

[Hsu11] HJ Hsu. The Potential of Kinect as Interactive Educational Technology. *2nd International Conference on Education and Management Technology*, 13:334–338, 2011.

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[KFL08] Nima Kaviani, Matthias Finke, and Rodger Lea. Encouraging crowd interaction with large displays using handheld devices. *Computing Systems*, pages 1–3, 2008.

[KKM+05] Juha Kela, Panu Korpipää, Jani Mäntyjärvi, Sanna Kallio, Giuseppe Savino, Luca Jozzo, and Sergio Di Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, 10(5):285–299, August 2005.

[KR05] Eamonn Keogh and CA Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[LD08] Yue M Lu and Minh N Do. Sampling signals from a union of subspaces. *Signal Processing Magazine, IEEE*, 25(2):41–47, 2008.

[Lee08] JC Lee. Hacking the nintendo wii remote. *Pervasive Computing, IEEE*, 7(3):39–45, 2008.

[LG03] Jessica Lin and Dimitrios Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining*, 2003.

[LHS08] Dan Lockton, David Harrison, and Neville Stanton. Design with Intent : Persuasive Technology in a Wider Context 2 Perspectives on Design with Intent. *Persuasive Technology*, 5033:274–278, 2008.

[Lun01] Arnold M Lund. Measuring usability with the use questionnaire. *Usability interface*, 8(2):3–6, 2001.

[LWS+12] Hai-Ning Liang, Cary Williams, Myron Semegen, Wolfgang Stuerzlinger, and Pourang Irani. User-defined surface+motion gestures for 3d manipulation of objects at a distance through a mobile device. *Proceedings of the*

10th asia pacific conference on Computer human interaction - APCHI '12, (c):299, 2012.

[LZWV09] Jiayang Liu, Lin Zhong, Jehan Wickramasuriya, and Venu Vasudevan. uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, 5(6):657–675, December 2009.

[McC01] James J McCarthy. *Climate change 2001: impacts, adaptation, and vulnerability: contribution of Working Group II to the third assessment report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, 2001.

[OCO10] Rodrigo De Oliveira, Mauro Cherubini, and Nuria Oliver. MoviPill: improving medication compliance for elders using a mobile persuasive social game. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 251–260, 2010.

[OEC08] OECD. OECD Environmental Outlook to 2030. http://www.oecd.org/environment/indicators-modelling-outlooks/40200582.pdf, 2008. Accessed: 10/01/2013.

[PK08] Peter Peltonen and Esko Kurvinen. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1285–1294, 2008.

[Pre13] Apple Press. Apple's App Store Marks Historic 50 Billionth Download. http://www.apple.com/pr/library/2013/05/16Apples-App-Store-Marks-Historic-50-Billionth-Download.html, 2013. Accessed: 27/08/2013.

[Pyl05] T Pylvänäinen. Accelerometer based gesture recognition using continuous HMMs. *Pattern Recognition and Image Analysis*, 3522:639–646, 2005.

[RLL11] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *CHI '11 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 197–206, New York, New York, USA, 2011. ACM Press.

[Row13] Dan Rowinsky. Google Play Hits One Million Android Apps. http://readwrite.com/2013/07/24/google-play-hits-one-million-android-apps, 2013. Accessed: 16/09/2013.

[Sal12] Ricardo Salvador. Gesture Based Persuasive Interfaces for Public Ambient Displays. Master's thesis, Faculty of Science and Technology (FCT), New University of Lisbon, 2012.

[San12] Bruno Santos. Changing Environmental Behaviors through Smartphone-Based Augmented Experiences. Master's thesis, Faculty of Science and Technology (FCT), New University of Lisbon, 2012.

[Sat01] M Satyanarayanan. Pervasive computing: Vision and challenges. *Personal Communications, IEEE*, 8(August):10–17, 2001.

[SBG99] Albrecht Schmidt, Michael Beigl, and HW Gellersen. There is more to context than location. *Computers & Graphics*, 23(6):893–901, December 1999.

[SC07] Stan Salvador and Philip Chan. FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[Smi03] Steven W Smith. *Digital signal processing: a practical guide for engineers and scientists*. Access Online via Elsevier, 2003.

[SPHB08] Thomas Schlömer, Benjamin Poppinga, Niels Henze, and Susanne Boll. Gesture recognition with a wii controller. In *Proceedings of the 2nd international conference on Tangible and embedded interaction*, pages 11–14. ACM, 2008.

[Spo13] EA Sports. The sims. http://thesims.com, 2013. Accessed: 30/08/2013.

[SRaD+12] Bruno Santos, Teresa Romão, A. Eduardo Dias, Pedro Centieiro, and Bárbara Teixeira. Changing environmental behaviors through smartphone-based augmented experiences. In *Proceedings of the 9th international conference on Advances in Computer Entertainment*, ACE'12, pages 553–556, Berlin, Heidelberg, 2012. Springer-Verlag.

[Vat12] RD Vatavu. Nomadic gestures: A technique for reusing gesture commands for frequent ambient interactions. *Journal of Ambient Intelligence and Smart Environments*, 4(2):79–93, 2012.

[Wei91] M Weiser. The computer for the 21st century. *Scientific American*, 256:94–104, 1991.

[ZBA09] X Zabulis, H Baltzakis, and A Argyros. Vision-based hand gesture recognition for human-computer interaction. *The Universal Access Handbook, Human Factors and Ergonomics, pages*, pages 1–56, 2009.

[ZCWY09] X Zhang, X Chen, W Wang, and J Yang. Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 401–405, 2009.

Appendix: Proof of Concept Questionnaire

In this chapter, the document that was used to conduct the usability conclusions about the users experience while playing the game that was used as a proof of concept for the gesture recognition system is exposed.

Questionnaire

In the following pages you will find several questions about your experience as a user of the game that is used as a proof of concept for the gesture recognition system. We hope that you answer them all, always sincerely.

Note that all your answers and data is anonymous and confidential and will only be used in the purpose of this study.

User

1. **Age**

   _____ years

2. **Gender**
   - □ Male
   - □ Female

3. **Academic Degree**
   - □ Doctor/Postdoctoral
   - □ Postgraduate/Masters
   - □ Bachelor (Complete)
   - □ Bachelor (Incomplete)
   - □ High School
   - □ Primary School
   - □ None

Background

4. **Are you familiar with new technologies?**
   - □ Yes
   - □ No

5. **Which kind of technologies do you use? How Frequently?**
   - □ Computer Rarely □—□—□—□ Daily
   - □ Smartphone Rarely □—□—□—□ Daily

□ Cellphone Rarely □—□—□—□ Daily

□ Game Consoles Rarely □—□—□—□ Daily

□ Internet Rarely □—□—□—□ Daily

□ Other _____ Rarely □—□—□—□ Daily

6. **I usually recycle at home.** Strongly Disagree □—□—□—□—□ Strongly Agree
7. **I am aware of the world's environmental issues.** Strongly Disagree □—□—□—□—□ Strongly Agree

The Game

General

8. **I liked to play the game.** Strongly Disagree □—□—□—□—□ Strongly Agree
9. **It was easy to learn how to play the game.** Strongly Disagree □—□—□—□—□ Strongly Agree
10. **It was easy to play the game.** Strongly Disagree □—□—□—□—□ Strongly Agree

Usability

11. **It was easy to identify which player I was.** Strongly Disagree □—□—□—□—□ Strongly Agree
12. **It was easy to learn the different gestures.** Strongly Disagree □—□—□—□—□ Strongly Agree
13. **The gestures were easy to perform.** Strongly Disagree □—□—□—□—□ Strongly Agree
14. **It was easy to use the device as a gesture controller.**
    Strongly Disagree □—□—□—□—□ Strongly Agree
15. **The game reacted quickly to my gestures.** Strongly Disagree □—□—□—□—□ Strongly Agree
16. **The game recognized the gesture I intended.** Strongly Disagree □—□—□—□—□ Strongly Agree
17. **Sometimes I wonder if I used the correct gesture.**
    Strongly Disagree □—□—□—□—□ Strongly Agree
18. **The feedback given by the game when I performed a gesture was useful.**
    Strongly Disagree □—□—□—□—□ Strongly Agree
19. **Sometimes I forget to release the button for performing a gesture.**
    Strongly Disagree □—□—□—□—□ Strongly Agree

Results

20. I understood the main ideas of the game.

Strongly Disagree □—□—□—□—□ Strongly Agree

21. The game reinforced environmental ideas I previously had.

Strongly Disagree □—□—□—□—□ Strongly Agree

22. My overall opinion about the game is good.

Strongly Disagree □—□—□—□—□ Strongly Agree

23. I won the game.

□ Yes

□ No

Suggestions and Comments