

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE GRADO

**DEVELOPMENT OF AN APPLICATION FOR VISUAL IMPAIR PEOPLE
IN THE EDUCATIONAL FIELD**

Silvia Morando Fernández
Tutor: Carolina A. Islas Sedano
Ponente: Rosa M. Carro

Septiembre 2014

DESARROLLO DE UNA APLICACIÓN PARA PERSONAS CON
PROBLEMAS DE VISIÓN EN EL ÁMBITO DE LA ENSEÑANZA

Silvia Morando Fernández
Tutor: Carolina A. Islas Sedano
Ponente: Rosa M. Carro

Summary and Keywords

Abstract – This Project has been mostly motivated by the important and current technological improvements. My stay in Finland one year ago allowed me to be introduced in interesting aspects about technology in the educational field.

Impair people have a limitation when using these technologies in education due to they can only use their hearing sense. For this reason, it would be interesting to give these people the chance to obtain information by hearing, but also by touch throughout vibrations.

To start with this objective, it has been built an application for mobile devices called 'PicRead' that simulates the colours with sounds and vibrations, so the user, when touching the screen, may be able to obtain information that cannot be reached by sight. If at the end of the project development the visual impair people can read graphics and colours, then a learning opportunity is built to include this technology to the actual ecology of resources to support the transmission of information.

Keywords – Impair visual people, education, sonification, vibration, kinesthesia, haptic, haptic technologies, perception, forces, hearing, touch, sight, colours, graphics, forms.

Resumen y Palabras clave

Extracto – Este proyecto ha estado motivado principalmente por los grandes avances tecnológicos que tenemos actualmente. Mi estancia en Finlandia hace un año me permitió conocer interesantes aspectos de la tecnología aplicada a ámbitos educativos.

Las personas con problemas visuales tienen una limitación a la hora de usar estas tecnologías en educación, puesto que solamente pueden usar su sentido del oído. Por este motivo, es interesante el poder darle a estas personas la oportunidad de obtener información mediante el oído, pero también mediante el tacto a través de vibraciones.

Para comenzar con este objetivo, se ha creado una aplicación para dispositivos móviles llamada 'PicRead' que va a simular los colores con sonidos y vibraciones, de tal manera que cuando el usuario toque la pantalla, sea capaz de obtener esa información que visualmente no puede tener. Si los usuarios al final del desarrollo de esta aplicación son capaces de reconocer colores y figuras a través de sonidos y vibraciones, se habrá creado una gran oportunidad en el campo educativo para incorporar estos recursos como apoyo en la transmisión de información.

Palabras clave – Invidentes, educación, sonificación, vibración, kinesthesia, háptico/táctil, tecnologías hápticas/táctiles, percepción, fuerzas, oído, tacto, vista, colores, gráficos, formas.

Index

Summary and Keywords	I
Resumen y Palabras clave.....	II
Figures Index.....	V
Tables Index	VII
Glossary	IX
1. Introduction	1
1.1. Goals	2
1.2. Challenges	2
1.3. Stages of Development	3
1.4. Report Structure	5
2. State of Art	7
2.1. Previous Work Research	7
2.2. Plugin Description: Immersion UHL Plugin	9
3. Requirements Analysis	15
3.1. Functional Requirements	15
3.1.1. Colours Management.....	16
3.1.2. Sounds Management.....	17
3.1.3. Training Management.....	18
3.1.4. Gallery Management	19
3.1.5. Picture Management	20
3.1.6. Results Management	20
3.1.7. Description of the application	21
3.2. Non-Functional Requirements.....	23
4. Solution Design	25
4.1. Architecture of the application	25
4.2. Modules description	27
4.2.1. General logic of the application	29
4.2.2. Modules logic	29
4.3. Interface	32
4.4. Conclusions	40

5. Implementation	43
5.1. Code structure	43
5.2. XML files structure	46
5.3. Interface with HTML5-JQueryMobile	48
5.4. Conclusions	50
6. Implantation and Tests	51
6.1. Implantation	51
6.2. Tests battery and Results	52
6.2.1. Unit Testing	53
6.2.2. Integration Testing	54
6.2.3. System Testing	55
6.2.4. User Testing	56
7. Conclusions and Future work	59
Bibliography	61
Appendix A. Colours Ranges	63

Figures Index

Figure 1.1. Software Development Life Cycle	4
Figure 2.1. PHANTOM™ device.....	8
Figure 2.2. Rotary knob created by Immersion Corporation	9
Figure 2.3. Virtual reality force-feedback glove.....	9
Figure 2.4. UHL components	11
Figure 2.5. Haptic Effect Preview Application from the Android Market	11
Figure 2.6. Haptic Muse Application from the Android Market	12
Figure 2.7. Categories in the Haptic Effect Preview Applications	12
Figure 2.8. Haptic Effects of 'Single Clicks' category	13
Figure 2.9. Haptic Muse map	13
Figure 2.10. Speedway Stop on Haptic Muse Application	14
Figure 2.11. Effect played after touching the car	14
Figure 4.1. Diagram with the Design of the application.....	26
Figure 4.2. colour.js diagram	29
Figure 4.3. training.js diagram.....	30
Figure 4.4. gallery.js diagram	30
Figure 4.5. picture.js diagram	31
Figure 4.6. results.js	31
Figure 4.7. Splash Screen.....	33
Figure 4.8. Main Page	33
Figure 4.9. White Training.....	33
Figure 4.10. Yellow Training	33
Figure 4.11. Orange Training	34
Figure 4.12. Red Training	34
Figure 4.13. Light blue Training	34
Figure 4.14. Dark blue Training.....	34
Figure 4.15. Green Training	34
Figure 4.16. Purple Training.....	34
Figure 4.17. Black Training.....	35
Figure 4.18. Gallery Menu	35
Figure 4.19. Shapes Gallery Page I	36
Figure 4.20. Shapes Gallery Page II	36

Figure 4.21. Yellow Triangle	36
Figure 4.22. Green Square	36
Figure 4.23. Shapes Gallery Questionnaire	37
Figure 4.24. Message for a wrong answer	37
Figure 4.25. Message for a correct answer	37
Figure 4.26. Signals Gallery Page.....	38
Figure 4.27. Warning Signal.....	38
Figure 4.28. Signals Gallery Questionnaire	38
Figure 4.29. Food Gallery Page I.....	39
Figure 4.30. Food Gallery Page II.....	39
Figure 4.31. Strawberry Picture.....	39
Figure 4.32. Food Gallery Questionnaire	40
Figure 5.1. Ranges of a colour in ColourRangesXML.xml file.....	46
Figure 5.2. Audio file of a colour in SoundsXML.xml file.....	47
Figure 5.3. Colour file of the training in TrainingColourXML.xml file	47
Figure 5.4. Pictures files on a page of the ShapesGalleryXML.xml file.....	47
Figure 5.5. HTML5 basic structure	48
Figure 5.6. Modules references of the PicRead application	48
Figure 5.7. Definition of a page with HTML5/JQueryMobile	49
Figure 5.8. Definition of the header element on a page with HTML5/JQueryMobile	49
Figure 5.9. Definition of the content element on a page with HTML5/JQueryMobile.....	50
Figure 6.1. Coordinates and Colour Test.....	53
Figure 6.2. User's finger path Test.....	54
Figure 6.3. Example of non-responsive application	55
Figure A.1. RGB box values represented with x,y,z.....	55

Tables Index

Table A.1. RGB ranges for the colours recognized in PicRead application	64
---	----

Glossary

SLDC	Software Development Life Cycle
UHL	Universal Haptic Layer
Plugin	Complement of an application that offers a new and specific functionality
UHL Plugin	Haptics plugin
Haptic	Nonverbal communication involving touch
Haptic Technology	Tactile feedback technology that simulates the sense of touch by using vibrations and forces
Sonification	Use of non-speech audio to transmit information and perceive data
Haptic Technology	Tactile feedback technology that simulates the sense of touch by using vibrations and forces
HTML5	HyperText Markup Language, version 5. Basic language for WebPages.
JavaScript	Dynamic programming language. Used as part of web browsers
JQuery	JavaScript library that allows a simpler interaction with HTML documents
JQueryMobile	HTML5 based user interface system. It is designed to create responsive webs or apps.
Framework	Reusable software platform to develop software applications. It includes compilers, debuggers, testers, code libraries, etc.

1

Introduction

It is surprising how important has been the progress of technologies in the last decade in numerous fields. Currently, there are thousands of different products created to make our lives easier and more comfortable, that we cannot even imagine. From the digital technology the most spread product worldwide are the mobile phones. These products allow users to carry out plenty of actions, from information searches till e-shopping, commonly used lately. Constantly new applications and products appear on the market, compatible with these smartphones that try to meet the needs of the society, encouraging the development of new ideas that fit in the goal of making easier user's life.

However, not all progress has only advantages. The current society has many needs and these products cannot cover all of them, especially those that involve a change in the human and social area. Progress in this field is still slower than in other fields, which is creating an inequity in adapting to changes for different users. This is the case of people with visual disabilities, who have been limited in their ability to obtain information from a Smartphone because of touchscreens. It is true that they have the support of applications or functionalities in the devices that play sounds to guide them in its use [1], but the other senses, such as touch, have been limited.

Nevertheless, although progress in smartphones is more limited in the social field, there are products that cover these limitations and offer what has not been achieved with smartphones yet. Such is the case of the 'Haptic Technologies', which will be discussed along the paper and which have managed to reproduce forces and vibrations to simulate the touch of an object.

The goal of the project is to explore how the combination of haptic technologies with sonification can, potentially, support non-visual individuals to interpret colours and figures through their mobile devices, in particular their mobile phones. This combination can be also used as a tool for improving skills in imagination or spatial vision. Therefore, the application could be use in the educational field by all kind of users.

1.1. GOALS

The motivation of this project started after some months in Finland where I was able to attend few courses related to technologies for development and education. It was then when I started thinking that it would be interesting to develop an application to support individuals with disabilities. Hence, I decided to contact the Educational Technologies research group at University of Eastern Finland and ask for a supervisor for my project.

Dr. Carolina Islas Sedano agreed to be my tutor for my work and it was agreed to develop an application for visual impair individuals. At the moment of this work, we had the advantage that one visual impair person from Iran, who was studying in Finland, would be our tester of the application. Some studies [2] [3] say that the better way to learn is by using at least two of our senses: sight and hearing, or hearing and touch. With the use of Smartphones or tablets, blind people are too limited because they can only use one of their senses: hearing.

The main goal of this Bachelor Project is to create an application where an individual should be able to interpret graphics by using vibrations and sounds, so two senses are used: touch and hearing. This application will be run on tactile devices, such as smartphones or tablets, and it will be used to give access to visual impair individuals to information that otherwise they would not have access too. If visual impair people can read graphics and colours, then a learning opportunity is built to include this technology to the actual ecology of resources to support the transmission of information. This learning opportunity may also be an opportunity to improve inner skills such such as imagination, spatial vision and memory.

1.2. CHALLENGES

This project present two main challenges: technological and in communication.

Technology:

This application is based on an existing plugin called '*Immersion UHL Plugin*' composed by 124 different vibrations divided in fourteen categories that simulates different aspects: textures, pulses, alerts, ramp ups, ramp downs, explosions, etc. The use of this plugin means a challenge in the development of the application because it involves testing the plugin and understanding its functions and functionality.

Additionally, due to the technological resources available in Spain and Finland, it was decided to develop this application for Android devices. Before starting this project, the author needed to

understand the use of Android platform and study how to develop applications for it: programming language, libraries, and modules, among others. In addition, the programming languages to be used for developing this application will be HTML5, JQuery, JavaScript and JQueryMobile that will be explained more in detail in chapter 4. Before this project I have not worked with those languages and neither with Android devices, so it will be a challenge for me due to its novelty.

Communication:

Due that my tutor and I were based in different countries, Spain and Finland, there were challenges in the schedules of our meetings, at time of the technology (we mainly used Skype and emails), might have problems. This adds some difficulty in order to organize and develop the project, so it was needed to synchronize times and schedule meetings to monitor the project and resolve incidents as they arise.

Another challenge to be added is the difficulty of testing and debugging, which are not fully understood via Skype only.

1.3. STAGES OF DEVELOPMENT

In order to guarantee success and good execution of the project it is necessary to set the stages that will be followed in the development of the final application. During the degree we have studied the *Software Development Life Cycle* [4] as an important part in developing a software product from the initial stage till the last one. The purpose of this cycle is to define the intermediate stages that are needed to validate and test the development of the application. That is, to guarantee that the software accomplish with the requirements of the application and to verify the development procedures. The stages are the ones drawn of the Figure 1.1 and are explained below:

1. **Requirement Analysis:** In this stage the idea and the objective of the project are analyzed by defining its requirements and limitations. This is very important to mark out the area of the project so it can be decided what has to be developed exactly.
2. **Design:** Once it is known what has to be built, it can be defined the interface and architecture of the final product. This stage is important to establish if the project will be composed by modules, and if so, specify how these modules will be connected. Here, the programming language that will be used or how the data will be stored in the application it is going to be discussed.

3. **Implementation (plus Unitary Tests):** This is the longest stage of the development. Here the code of the application and the data storage will be implemented in the programming language that has been decided in the previous stage. If in the 'Design' stage it was decided to define modules in the application, in this stage there will be the '*Unitary tests*' to test each module one by one and after that the integration between them.
4. **Testing (plus Implantation):** In this stage the whole application will be tested and it will be installed in Android devices so the final user can play it and evaluate it. The fulfilment of the requirements, the efficacy of the application and the approval of the final user will be the criteria to determine whether the development is successfully completed or there are still things that can be improved.
5. **Evolution or Maintenance:** Like in every software project, it is essential that the developer provides the user with maintenance of the application. Here, and as the user needs it, new or better functionalities will be added or the detected errors will be corrected. This stage will be followed by the first one, '*Requirement Analysis*', so we can start the cycle again.

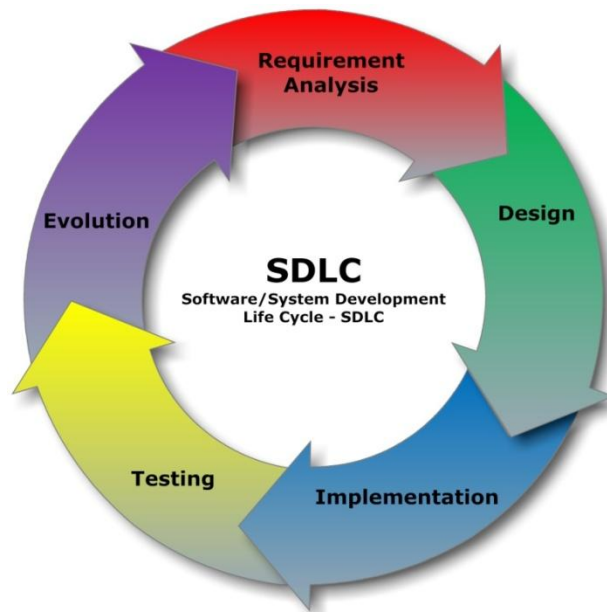


Figure 1.1. Software Development Life Cycle

These stages are created in order to avoid making mistakes because then, in the final stages (for example, in the '*Implementation*' stage) it is much harder to solve the errors than in the previous ones.

Although it has not been mentioned yet, there is another stage that should be considered when developing a product, and that is the '**Documentation**' part. It is necessary to spend time writing on a paper what has been done in each stage and the results of each one, including commenting the code of the application. This is very important because, if someone wants to join the project in the future, it would be easier if there is documentation available to understand the aim of the project, what has been achieved and what has to be improved.

1.4. REPORT STRUCTURE

This report includes the stages of the Software Development Life Cycle mentioned in the previous section. This paper also includes a chapter that is not in the SDLC but it is important to situate the reader of this paper before start talking about the application, its requirements or its design. This is the next chapter where the State of Art will be analyzed to know which can be found currently related to the topic of this Bachelor Project.

After the State of Art chapter the stages of development explained in the SDLC will be presented:

- In **chapter 3** can be found the **requirement analysis** of the application, correspondent with the first stage of SDLC. Its comprehension will be important to understand the purpose and the aim of the application.
- In **chapter 4** can be found the **final product design**. In this design will be explained the architecture of the application, its interface, user cases, modules description and the decision that affects next chapter, Implementation.
- In **chapter 5** can be found the aspects related to the **implementation** and developing process of the application, based on the programming language and the design set in the previous chapter.
- In **chapter 6** can be found the different types of **testing** made to check the correct functionality of the application. Here will be also explained how was the **implantation** of the application in Android devices.
- In **chapter 7** can be found the personal conclusions and evaluation of the developed application. Also, possible improvements will be introduced for future work.

2

State of Art

Throughout this project we are going to introduce the term of 'Haptics' followed by a presentation about what it is known as '*Haptic Technologies*', mentioned before in the introductory chapter.

Although we will see that there is not any application on the market related with the aim of this project, other gadgets and current researches related to haptics that are quite interesting will be analyzed. It will be also explained why this analysis was important to define the ideas for the project, and how later, based on these ideas, the objectives and the goals were set for this Bachelor Project.

Lastly, it will be analyzed the lack of coverage of some of these products and the possible upgrades that can be offered to build an original and novel product based on existing products.

2.1. PREVIOUS WORK RESEARCH

The term '*haptics*' comes from the Greek term *ἅπτικός* (*haptikos*) which refers to the sense of touch [5]. So when talking about 'Haptic technologies' this term is referring to those technologies that act on human touch. More in detail, haptic technologies are those that try to simulate exchanges of forces, by the use of vibrations, between the gadget and user's hands or bodies.

This communication through touch between a person and a machine is allowed by the haptic interface. These interfaces produce signals that simulate the sensation of movement in muscles,

tendons and joints, what it is known as *kinesthesia* [6]. It also includes the ability to perceive self body's position, weight or movement.

A haptic device has to be designed to 'read' and 'write' from the human body. Currently, the reading part has been extensively explored, creating devices and gadgets that are commonly known, such as joysticks or pointing devices, but the writing part needs to be explored more in detail because it can involve a technology revolution [7].

In order to achieve this, it is needed programmable devices that may recreate mechanical issues easy to recognize and with functional importance. Haptic perception relies on how the perceiver's nervous system processes information arising from interaction. A full understanding of haptic perception requires controlling the haptic stimuli occurring during interaction with an environment [8].

Today, advances in haptic technologies allow a better perception research through haptic devices which allow reading and writing from the human body. They track a user's physical manipulation (reading) and provide realistic touch sensations coordinated with on-screen events (writing) [9]. Some examples are shown below. Other examples can be found in [10]:

PHANTOM™ Device: The position and orientation of the pen are tracked through encoders in the robotic arm. There are three actuated degrees of freedom and three sensed orientations to obtain the direction.



Figure 2.1. PHANTOM™ device

Rotary controllers: A rotary knob with haptic feedback has been created by Immersion Corporation, the creators of the plugin that will be used for the project application, but this will be explained later. This device achieves a crucial function substituting touch for vision when integrated in a vehicle.



Figure 3.2. Rotary knob created by Immersion Corporation

Virtual reality force-feedback glove: Burdea and co-workers have created force-producing elements which act on discrete areas inside a user's hand. The use of pneumatic pistons makes it possible to create a low weight and portable device to simulate the feeling of a virtual object.



Figure 4.3. Virtual reality force-feedback glove

2.2. PLUGIN DESCRIPTION: IMMERSION UHL PLUGIN

If we think about it, for the people without any visual disabilities, the sense of touch helps us completing our mental picture of an object and not just by sight. This idea is even more important for people with visual disabilities because the sense of touch means a world for them and helps them to get more complete information that they cannot get with other senses.

Based on the idea of 'Haptics' referring to the act of stimulating a user's sense of touch, the creators of the plugin, Immersion's TouchSense® solutions and Haptic Development Platform, are focused on enabling devices and software developers to generate touch sensations that a device user can feel.

In the past 20 years, mobile devices have focused on vibration feedback but unfortunately they have been extremely mundane and simple. An example of this is the very well known '*buzz buzz*' that alerts us of some important event, like an incoming phone call, message or alarm clock. Some years ago, Immersion's TouchSense® solutions wanted to change how vibrations are experienced on mobile devices and the result was the '*TouchSense Player 3000*', which is now deployed across many mobile phone handset models, and the '*Haptic SDK*', that allows software developers to create range of vibrations that can be used in mobile devices with a vibration capability.

These tools allowed to create the 'Immersion UHL Plugin', in which is based this Bachelor Project. The Universal Haptic Layer (UHL) is a JAR file and .SO file containing all the classes, interfaces and algorithms that developers include in their applications to create haptics [11]. This is what UHL has on its inside:

Effect Library:

A Haptic effect library with the most commonly used effects in the applications. These are indexed and hard-coded in the UHL. In the plugin are 124 effects designed, which are divided in fourteen different categories [14].

Launcher:

The Launcher interface helps creating good haptics by indentifying the characteristics of the devices and by rendering the effects based on those characteristics. If the device where it is launched is a TouchSense handset, the TouchSense Player will render the effects. If not, the UHL will use the TS3000 Emulator.

Haptic API:

It is the API that controls de TouchSense Player or the TS3000 Emulator depending on whether it is a TouchSense handset or not.

TS3000 Emulator:

With the Emulator, the haptic effects will be displayed even if the application it is not played in an Android device with the TouchSense Player. The quality is better playing with the TouchSense player, but the emulator has good results too.

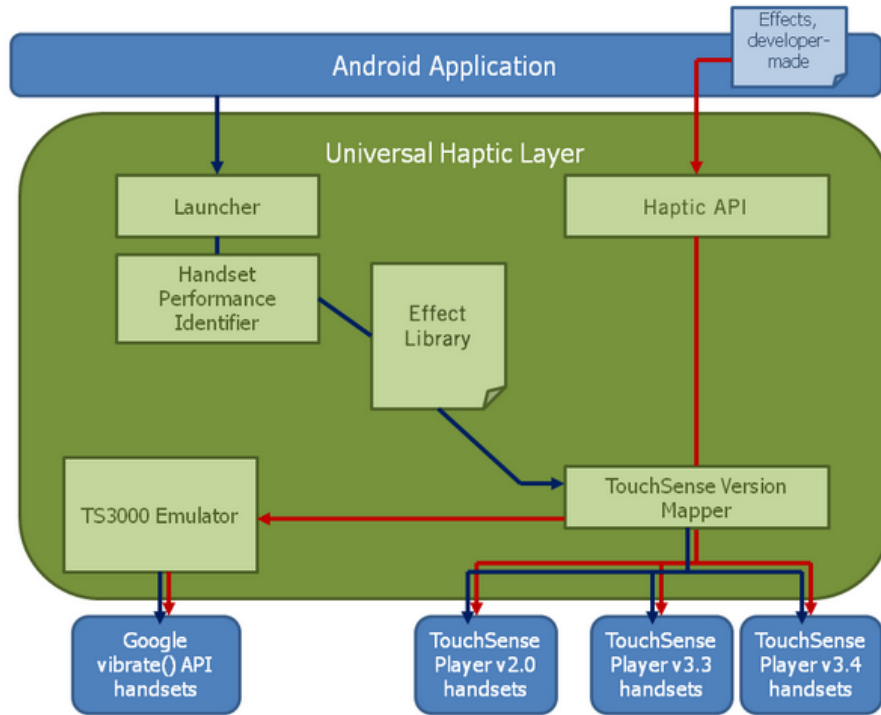


Figure 5.4. UHL components

To help the developers to get familiar with the different haptic effects, there are two available applications that can be downloaded from the Android market: 'Haptic Effect Preview', whose icon can be found at Figure 2.5, and 'Haptic Muse', whose icon can be found at Figure 2.6.

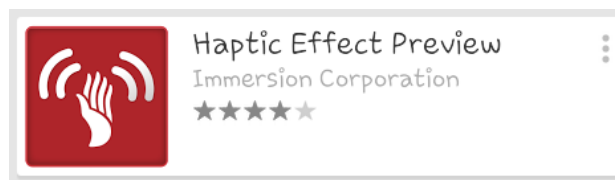


Figure 6.5. Haptic Effect Preview Application from the Android Market

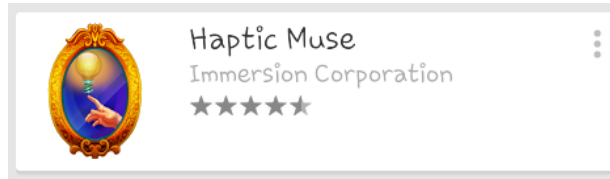


Figure 7.6. Haptic Muse Application from the Android Market

The first one, 'Haptic Effect Preview' is a simple application where the developer can find a list composed with fourteen categories: Single clicks, Double clicks, Triple Clicks, Buzz, Ramp ups, Ramp downs, Pulses, Buzz-bumps, Alerts, Explosions, Weapons, Collisions, Engines and Textures.

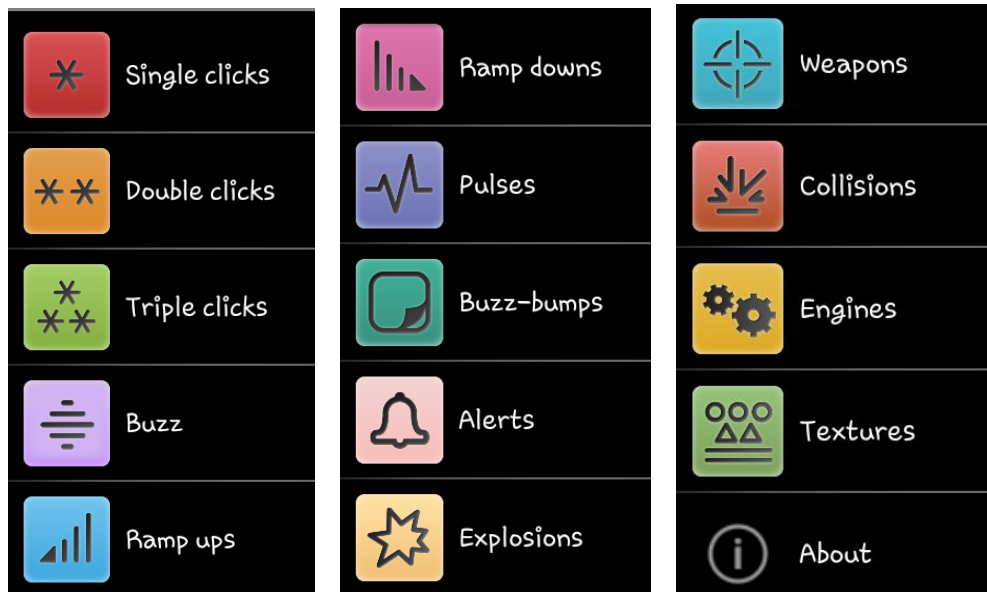


Figure 8.7. Categories in the Haptic Effect Preview Applications

By selecting one of them, the user can access to the haptic effects belonging to that category so they are able to test, feel and learn each of them. For example, by clicking on 'Single clicks' the user will access to the list shown in Figure 2.7.

The list of each category will have its belonging effects ordered by their names. By selecting one of these effects, there is a section on the top with its code and how to call that haptic effect inside the programming code. This is very useful for the developer because it allows him to select which effect exactly he wants to use depending on what he wants to program.

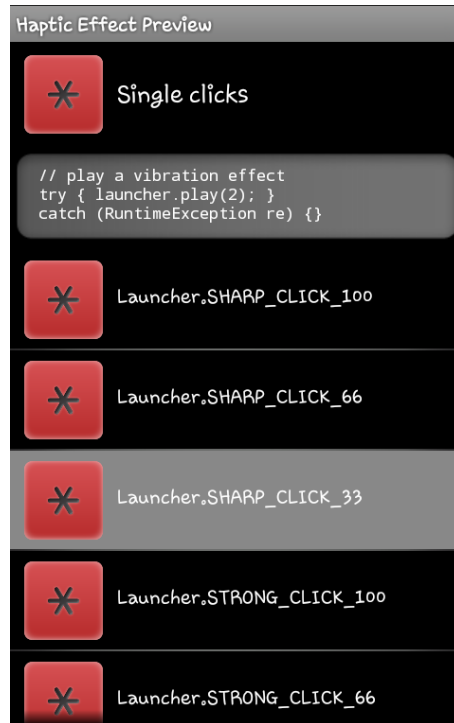


Figure 9.8. Haptic Effects of 'Single Clicks' category

The other application, 'Haptic Muse' is a friendlier and animated application. It is more attractive than the 'Haptic Effect Preview' application because it is design in a gaming context trying to inspire developers to enhance their games with haptic effects. When starting the application, the user access to a map with different stops simulating a treasure map:



Figure 10.9. Haptic Muse Map

In each of the places of the map the user will be able to play some different vibrations trying to simulate something related to that place. For example, if going to 'Speedway' place, the user will access to the place in Figure 2.10 and by pressing on one of the places of the screen indicated by the fingers, for example on the car, an effect or a mix of effects will be played, as we can see in Figure 2.11.



Figure 11.10. Speedway Stop on Haptic Muse Application



Figure 12.11. Effect played after touching the car

Thanks to both applications, we will be able to get use with the plugin to use it in our plugin. For the moment, we will see that the vibrations played in our application will be simpler, like in the Haptic Effect Preview application. For future development or improvements of the application, the Haptic Muse will be a good reference, but we will talk later on in Chapter 7.

3

Requirement Analysis

During this chapter, the requirements of the mobile application that is been developed will be introduced and explained in detail. These requirements will be the base for the design of the application to determine the elements that will be available in the interface of the application, the structure of the application to define its modules or the communication between the interface and the different modules that may compose this Bachelor Project.

With the combination of the requirements and the design, it will be possible to implement the project taking into consideration the necessities of the user and what has to be developed in the application. The correct understanding of the requirements, and after this the design of the application, will guarantee a good comprehension of the aim and purpose of the project.

Below it can be found the description of each functional and non-functional requirement that has been considered.

3.1. FUNCTIONAL REQUIREMENTS

Along this section the functional requirements will be described, based on the behaviour and functionality that is wanted for the final product. For a better understanding of the requirements, they will be divided in six subsections corresponding with the main functionalities of the product:

colours management, sounds management, training management, gallery management, pictures management and results management.

3.1.1. Colours Management

In this section the functional requirements related with the analysis of the colours will be presented. Although we will have nine different colours (yellow, orange, red, light blue, dark blue, green, black and white), the following requirements are common for all of them.

Before presenting them it is important to know that every colour is formed by the combination of the basics colours: red, green and blue. These three colours can get values from 0 till 255, and depending on the value of each of them, we will have different colours.

The functional requirements of Colour Management are:

RF1.1: Demarcate each colour by the range of the basics colours.

1. When starting the application, the range of each colour will be set by the combination of the basic colours: red (r), green (g) and blue (b). Each of these variables will have values between two numbers. We can see these ranges in the Appendix A of this report.
2. Due to the application will be very limited with the number of colours that will be recognized, it is important that each colour covers from the lighter till the darker tones of its colour. This is in order to recognize the most range of colours we can by grouping them in the nine available colours that we have in the application.

RF1.2: Link each colour with its corresponding vibration.

1. The colours will be associated with one of the available vibrations in the plugin.
2. For the pair of colours 'light blue' and 'dark blue' it has been selected the same vibrations, but with a different intensity. The dark blue colour has a vibration that is more intense than the one of the light blue colour.
3. For the pair of colours 'white' and 'black' it has been selected two opposite and completely different vibrations, based on the idea that they are opposite colours.

RF1.3: Detect when the user touches the screen and play the vibration of the colour where the finger is.

1. The user will touch the screen. Meanwhile, the application will read the coordinates where the finger is and it will detect the background colour of those coordinates.
2. The application will start playing the vibration of the background colour detected.

RF1.4: Detect when the user is not touching the screen and stop playing the vibration.

1. The user will stop touching the screen. The application will automatically stop playing the vibration that was being played at that moment, ready to start playing it when the user will touch the screen again.

3.1.2. Sounds Management

In this section the functional requirements related with the association of colours and sounds will be presented. As in the 'Colour Management' section, these requirements are common for each colour of the application:

RF2.1: Link each colour with its corresponding sound.

1. The colours will be associated with its corresponding sound.
2. The audios will be of different intensities and velocities, easy to recognize and differentiate from the other audios of the other colours.
3. The sounds will try to evoke feelings or thoughts to the user in order to help memorizing to which colour belongs that sound.

RF2.2: Detect when the user touches the screen and play the sound of the colour where the finger is.

1. The user touches the screen. The application will read the coordinates where the finger is and it will detect the background colour of those coordinates.
2. The application will start playing the audio of the background colour detected. The audio will be played in loop while the user is touching the screen to not stop the audio.

RF2.3: Detect when the user is not touching the screen and stop playing the audio.

1. The user will stop touching the screen. The application will automatically stop playing the audio that was being played at that moment.

RF2.4: Turn off the sound in the application.

1. Automatically, when the application is started the audio is on.
2. To turn the audio off when reading a picture the user will press the button 'OFF' in the right corner at the top of the screen. This button will only appear when the user has the chance to read a picture or a colour in the training. It will not appear when the user is on a menu.
3. The button will change to 'ON' allowing the user to turn the audio on again.

RF2.5: Turn on the sound in the application.

1. The application will let the user turn the audio on again after turning it off.
2. The user will have to press the button 'ON' in the right corner at the top of the screen. This button will only appear when the user has the chance to read a picture or a colour in the training. It will not appear when the user is on a menu.
3. The button will change to 'OFF' allowing the user to turn the audio off again.

3.1.3. Training Management

Due to the application that is been developed is a novelty for many people, it is not supposed that the user knows how the application works at the beginning. But not only how it works, but also how to read the colours or which vibration and sound belongs to every colour. For this, is very important to define functional requirements for Training Management that we are explaining below:

RF3.1: Start the colour training.

1. The user will select the option 'Start Training' in the menu that appears when the application is started.
2. The training will start with the white colour. By touching in the screen, it will start reproducing its vibration and sound.
3. To go to the next colour the user will click on the 'Next' button at the bottom of the screen. To go to the previous colour the user will click on the 'Previous' button at the bottom of the screen.
4. The order of the colours in the training is: White – Yellow – Orange – Red – Light blue – Dark blue – Green – Purple – Black

RF3.2: Leave the colour training.

1. The user can leave the training by pressing the 'Home' button at the top of the screen.
2. The last read colour will be memorized in the local memory of the application in case the user wants to return to the training and continue where he left it.

3.1.4. Gallery Management

In this section the functional requirements of the galleries will be described. The application will have three available galleries of different levels of difficulty. The user is free to access any of them, and their requirements are the followings:

RF4.1: Consult a gallery

1. The user will click on the 'Read Picture' button in the main page. The user will enter to another page where he will be able to select a gallery.
2. This page will have three buttons, each of them corresponding to one category: 'Level 1 – Shapes', 'Level 2 – Vegetables &Fruits' and 'Level 3 – Signals'. These levels are ordered by difficulty, being the Level 1 the easiest.
3. Each gallery has two pages of pictures. To go to the second page, the user will have to press the 'Next >' button at the bottom of the screen. To go to the first page, the user will have to press the '< Previous' button at the bottom of the screen.

RF4.2: Change to a different gallery

1. The user will have to press on the 'Menu' button at the top of the screen to leave the current gallery and access to the gallery menu.
2. The user will have to choose another gallery of the three available.

RF4.3: Return to the gallery after reading a picture

1. The user will be able to return to the gallery by clicking in the 'Back' button at the bottom of the screen.
2. It will then appear a questionnaire that the user will have to answer. This questionnaire contains different questions, depending on which gallery is the user in.

3.1.5. Picture Management

As mentioned before, the application will have three different galleries available. Each of them has pictures that the user can read and listen, and here we describe the functional requirements about this management:

RF5.1: Read and listen a picture.

1. The user will enter into a category. The user will be then able to select one of the available pictures.
2. The user will touch the screen to start reading the picture. The application will be checking the background colour constantly. As soon as a different colour is detected, the application will change the vibration and the sound to the ones belonging to the new colour.

3.1.6. Results Management

In this section the functional requirements related with the results after reading the pictures will be analysed. These results will be important to analyze how successfully was the reading process and which gallery had better results. Here we describe the functional requirements about this management:

RF6.1: Store the results after answering a questionnaire of the picture that has been read.

1. The user will have to complete the questionnaire. The answers will be stored in the local memory.
2. The answers will be then compared with the correct answers. If it matches, it will be stored as a successful reading. If not, it will be stored as a fail reading.

RF6.2: Present the results of the read pictures to the user.

1. The user can check his results of that current session. If the user has visual disabilities, he will need the help of someone else to obtain the results.
2. The results will be divided in three parts, each of them corresponding to each category. So for each category, the user will have the successful percentage and the failure percentage of the read pictures corresponding to that category.
3. After shutting down the application, those results will be removed.

3.1.7. Description of the application

The application that is going to be implemented is supposed to be used in two different scenarios:

- To help people with visual disabilities to learn colours and read and hear pictures.
- To help people to improved different skills like: imagination, spatial vision or shape recognition and memory.

The purpose of this application is to recognize the colours that compose a picture, and with the help of vibrations, recognize what the user is touching, delimit the form of the object or guess if there are corners, curves, straight lines... To evaluate if the user can do this, there are two tests that will be described later on:

- A gallery questionnaire included in the application
- A personal questionnaire apart from the application

However, before the evaluations, it would be recommended for the user to start with the colour training in order to be able to learn the colours and then apply this learning when reading the pictures. Once the user is ready and can recognize easily every vibration and audio of the colours, the reading process can be started. For this, there are three different galleries available in the application:

- **Level 1 – Shapes Gallery:** This is the easiest gallery to start with. After the learning process the user should enter to this gallery to start applying what was learnt in the colour training. This gallery is composed by pictures of shapes, so the aim of this first level is to be able to:
 - o Detect the colour of the shape.
 - o Detect the colour of the background.
 - o Delimit with the finger the form of the shape.
 - o Guess which shape is being touched.
- **Level 2 – Signals Gallery:** This is a gallery a bit more complicated than the previous one. It is recommended for the user to not enter into this gallery until he has not become an expert on the Shapes gallery. This gallery is composed by pictures of signals which are composed by different shapes (for example, the prohibition signal), so the aim of this second level is to apply what was learnt in the first gallery. This is:
 - o Delimit the form of the signal.
 - o Detect the colours that compose the signal.
 - o Detect the main colour of the signal.
 - o Guess which signal is being touch.
 - o Describe the meaning of the signal.

- **Level 3 – Food Gallery:** This last gallery is the most difficult gallery, although it is composed by pictures of vegetables and fruits that are really common for the people. However, at the same time this makes it even more difficult because they do not have a concrete shape, but lots of curves, and there are many vegetables or fruits with similar forms and colours. For this reason, it is recommended for the user to not enter into this gallery until he has become an expert reader with the other two galleries. The aim of this third level is to:
 - Delimit the form of the vegetable or fruit.
 - Detect the colours that compose the vegetable or fruit.
 - Detect the main colour of the vegetable or fruit.
 - Guess if it is a vegetable or a fruit.
 - If it is a vegetable, guess which one it is. If it is a fruit, guess which one it is.

Each time a picture is read, the user has to answer the first test: **Gallery Questionnaire**. There are three different questionnaires, one per gallery because there are different things to evaluate in each of them. In each gallery there are three questions related to the aims previously described in each gallery. So these are the questions for each gallery questionnaire:

- **Level 1 – Shapes Gallery:**
 - *Which shape did you read?* And the options between the user can choose are: triangle, heart, square, circle, pentagon, star, arrow or rhombus.
 - *Which was the background colour?* And the options between the user can choose are: white, black, yellow, orange, red, light blue, dark blue, green or purple.
 - *Which was the shape colour?* And the options between the user can choose are: white, black, yellow, orange, red, light blue, dark blue, green or purple.
- **Level 2 – Signal Gallery:**
 - *Which is the shape of the signal?* And the options between the user can choose are: triangle, heart, square, circle, pentagon, star, arrow or rhombus.
 - *Which is the main colour of the signal?* And the options between the user can choose are: white, black, yellow, orange, red, light blue, dark blue, green or purple.
 - *Which is the meaning of the signal?* And the options between the user can choose are: warning, prohibition, emergency, obligation or information.
- **Level 3 – Food Gallery:**
 - *Which type of food is it?* And the options between the user can choose are: vegetable or fruit.
 - *Which is the name of the food?* And the options between the user can choose are: eggplant, pumpkin, onion, corn, potato, pepper, tomato, carrot, cherries, strawberry, lemon, apple, grapes, orange, pear or banana.

- *Which is the main colour of the food?* And the options between the user can choose are: white, black, yellow, orange, red, light blue, dark blue, green or purple.

Those answers will be compared to the ones stored in the application. A correct answer will mean a +1 in the correct pictures in that gallery, and a wrong answer will mean a -1 in the wrong pictures of that gallery. At the end, the application will show the percentage of correct and wrong read pictures for each gallery.

After the user finishes playing the application, it would be appreciated that the user answers to the personal questionnaire to give feedback of the application. In this questionnaire he is supposed to answer the following questions:

- Was it easy to learn the colours or did it take a lot to memorize them?
- How difficult was to delimit the form of what you were reading?
- In which aspect of your daily life do you think this application can be used for?
- Which functionalities would you add to improve this application?
- Which things of the application do you think should be improved? Why?

3.2. NON-FUNCTIONAL REQUIREMENTS

In this section the requirements that are not related with the behaviour or functionality of the application will be presented. The non-functional requirements are the followings:

RFN1: The application will be run only on Android devices that has the TouchSense Player integrated.

RFN2: The application will have to be adapted to the different screen sizes and dimensions that present the Android devices.

RFN3: The application will have a simple interface. Due to the majority part of the final users will be visual impair people, the application will have oral instructions integrated to guide the user and describe the components of the different pages of the application. Although the application will have this functionality, it will be good that the user could also have someone to guide and support him.

RFN4: The application will be design in modules for a better and more organized implementation. This will help future developers of the application.

RFN5: The code will be ordered and correctly commented for the future maintenance.

RFN6: The size of the pictures and sounds stored in the application should be low to help with the rapidity execution of the application. If not, it will cause pauses, and in worst cases, errors in the application.

RFN7: For a better reading of the pictures, the vibrations and sounds associated to the colours have to be easy to distinguish from each other in order to do not be mistaken about guessing the colour and make the learning process of the colours easier and more attractive to the user and easier to keep them in mind.

RFN8: The application will be adapt to allow the user to change the volume of the sound by using the device controls for the volume.

RFN9: The application will have two kinds of users. Although the main users will be people with visual disabilities to let them hear and read colours to imagine what they are touching, this application may be also used by people without disabilities, for example, kids. This application can be a way of learning how to use human senses, specially the touch and hearing. By improving these senses, the users can improve other abilities: imagination, spatial vision or shape recognition and it will be a complete and important brain training.

4

Solution Design

Based on the requirements catalogue described in the past chapter, in this chapter the design process will be presented. There are going to be presented also the reasons of the decisions made. What is pretended to be on this chapter is how the things has been developed following the things we wanted in the requirements. It is important to have a thorough, complete and solid the design to guarantee the success of the application.

First of all, it will be explained the general design of the application and its architecture and interface for a better understanding of the idea. After that, it will be detailed all the different modules that compose the application, the interaction between them and their future integration.

4.1. ARCHITECTURE OF THE APPLICATION

In this section we are showing the architecture of the application. That is, the way that an application is designed based on its requirements.

As we have seen in the past chapter, the requirements have been divided in subsections. The diagram that we are presenting below shows the design of the application with all of the requirements subsections and the relation between them. In the following sections these modules will be described more in detail.

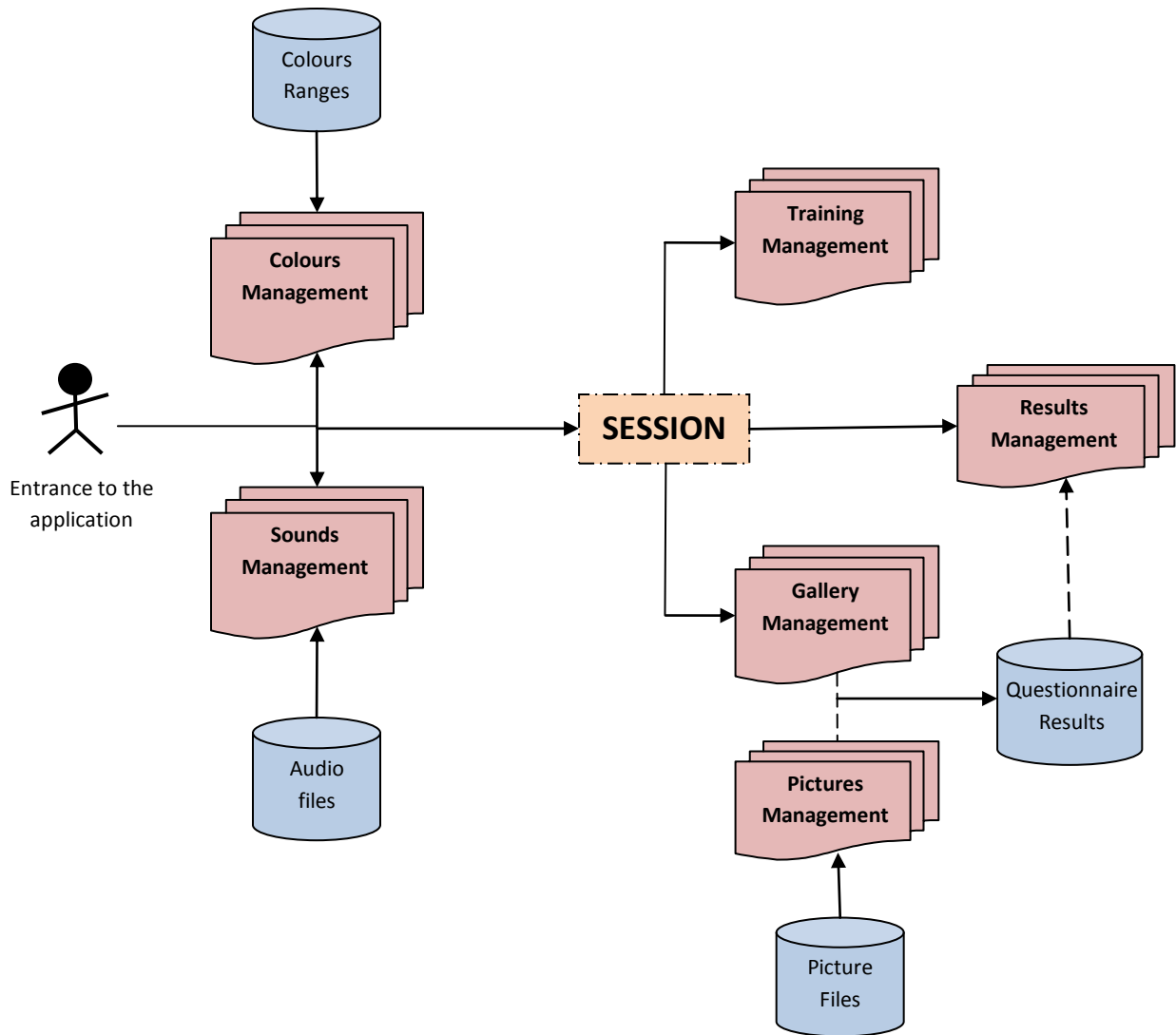
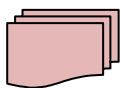


Figure 4.1. Diagram with the Design of the application

As we can observe in the diagram, the design is composed by two main elements:



Modules: They correspond with the functionality of the application.



Data: They correspond with the local data and files that will be used in the application.

For a better understanding of this diagram, we are describing it with use cases corresponding with real cases that the user can execute when using the application:

1. The user starts the application. While the application is loading, the system will define the colours using the 'Colours Management' module and the data stored with the colour ranges. It will define the sounds too, using the 'Sounds Management' module and the data stored with the name of the audio files.
2. The user will start a 'Session' and will access to the main page of the application where two actions are possible: start the training or read a picture. This creates two different use cases from this point.

Case 1: The user decides start the training

3. When the user selects this option, the application will use the 'Training Module' to define the training.
4. The user will be able to learn each of the nine colours by playing its vibration and sound.

Case 2: The user decides read a picture

3. When the user selects this option, the application will let the user select one of the three galleries available by the 'Gallery Management' module.
4. The user will select a gallery and he will access to the pictures available on that gallery. He will be able to read each of them by the 'Picture Management' module.
5. After reading a picture, the user will have to answer the questionnaire of that gallery. The results will be stored in the local memory by the 'Results Management' module.
6. When the user finishes reading pictures, he will be able to consult the results of the read pictures by returning to the main menu and clicking on the option of the results. This will be also managed by the 'Results Management' module.

4.2. MODULES DESCRIPTION

In this section the aim is to explain the logic of the application, explaining how the modules will be implemented, which programming language will be used in the modules or how the application will work.

To begin with, it is important to set the programming language that is going to be used for the development of the application. Due to it will be played on Android devices, we had two possible languages to choose that we are presenting below: Java and HTML5. Although it was decided to use HTML5, we will see that both of them have their advantages and disadvantages, but we are going to explain the reasons why we decided HTML5 was the best option for this development.

Java is the typical language that is use for Android application's development. It allows a better use of the resources, native graphics and other system components available in the device, higher stability and a better functionality. It has an interface development based on Eclipse with emulators on the desktop that makes the development easier and more comfortable. However, creation of new versions of the Operating System (OS) or new and different devices can present the difficulty that the application has to be updated regularly in order to not be obsolete.

On the other hand, **HTML5** is a language of description of user interface, so it needs another language to make the association with data. HTML5 can use many frameworks that support this association. Developing an application with HTML5 may be very difficult because it has to be tested on multiple browsers and adapt it to the characteristics of each of them, but it also provides multiple advantages. It guarantees the fact of working on all kind of operating systems and types of computers or mobiles, so it is sure that it will work on Android devices. It also allows a drawing surface or animation, or even sounds, being added to the application thanks to the new tag 'Canvas'.

As mentioned previously on the report, the aim of the project is to allow the user to read pictures by vibrations and sounds. For this reason, and based on the advantages presented before, we decided that the best language for programming this application would be **HTML5** by using its new feature: *canvas*, which allows introducing pictures, graphics and sounds. The problem of choosing HTML5 is that it is needed another language for the association with data, and here is where **JavaScript** takes part. It will be used to develop the main functionalities of the application, and with the help of **JQuery** it will be implemented the communication between the pages of the application. In the advantages of HTML5 it was mentioned that there are many frameworks that can be used when coding these applications. After a long search, there was a framework that prevailed over the rest: **JQueryMobile**, designed to make responsive web sites and apps, and accessible on all kind of smartphones and tablets. So, to sum up, the languages that will be used for the development of the application will be: the HTML5-based user interface system JQueryMobile, JavaScript and JQuery. In addition to this, and to be able to use JQuery and JQueryMobile, we are going to need three libraries, two JavaScript libraries and one CSS library:

- **JavaScript:** *jquery.mobile-min.js* and *jquery.min.js*
- **CSS:** *jquery.mobile-min.css*

In order to program and build the application, we are going to use an Intel tool called '*Intel XDK*', which allows programming an application in HTML5 and it also has the JQueryMobile builder integrated on it. For this reason, this tool makes it easier for the developer in order to code, test, debug and build the application.

4.2.1. General logic of the application

In the next section 4.3.2 the logic of the different functional modules that compose the application will be explained with their diagrams.

Before that, it is important to know that those modules are all connected between them, and at the same time with the interface which is the one that creates each module and calls the main methods of each of them with the functionalities of the application. These methods will be explained below.

4.2.2. Modules logic

In this section the details of each module are presented by using UML diagrams that will include the variables and methods available on that module. Each diagram will be followed by a short description.

The first module that will be analyzed is **colours.js**:

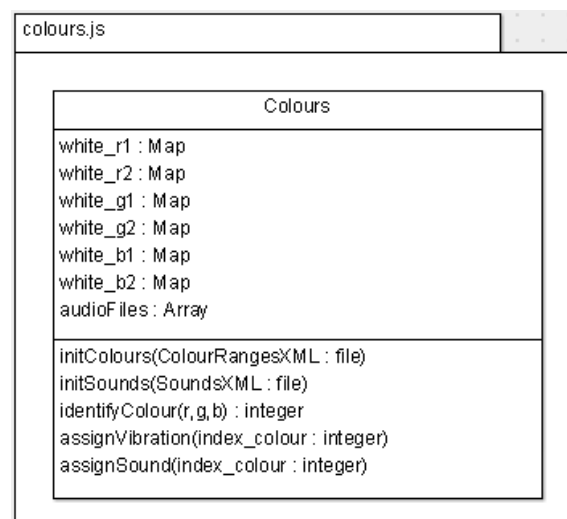


Figure 4.2. colour.js diagram

In the diagram can be seen different variables for the white colour. These variables correspond to the ranges of RGB (red, green, blue) values for the white colour that are read from the ColourRangesXML file and stored in those variables.

Although in the diagram it is not shown, each colour recognized on the application has these six variables _r1, _r2, _g1, _g2, _b1 and _b2 too like the white colour.

In this module the audio files correspondent to each colour are also read from its XML file (SoundsXML) and stored in array audioFiles.

The second module that will be analyzed is **training.js**:

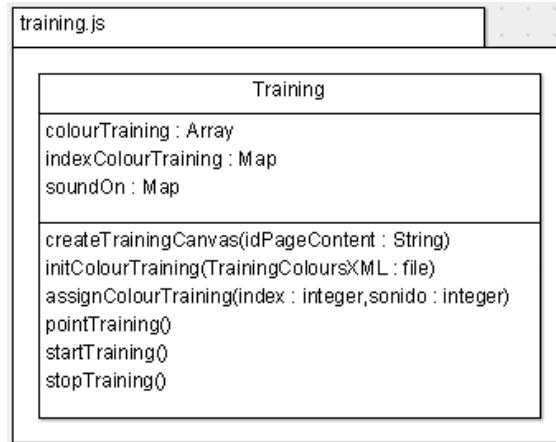


Figure 4.3. training.js diagram

In this module the file TrainingColoursXML is read to store the colours files of the training in an array. The application will access to this array to set the correspondent colour in the training each time the user presses the Next/Previous button. In this module the canvas that will include the training is created too. The functions pointTraining(), startTraining() and stopTraining() correspond to the definition of the event handlers to detect when the user is or is not touching on the screen to start/stop the training.

The third module that will be analyzed is **gallery.js**:

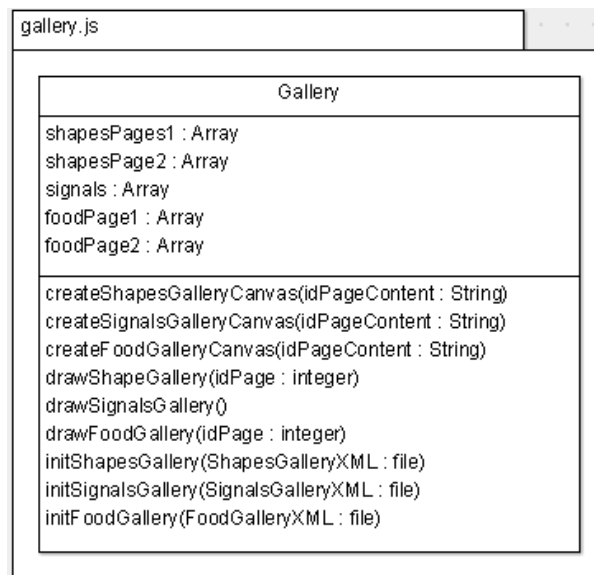


Figure 4.4. gallery.js diagram

In this module the XML files with the pictures files of the galleries are read and stored in their correspondent arrays. When a gallery is selected, the application will access to its array to paint its pictures in the canvas of that gallery, which is created in this module too.

The fourth module that will be analyzed is **picture.js**:

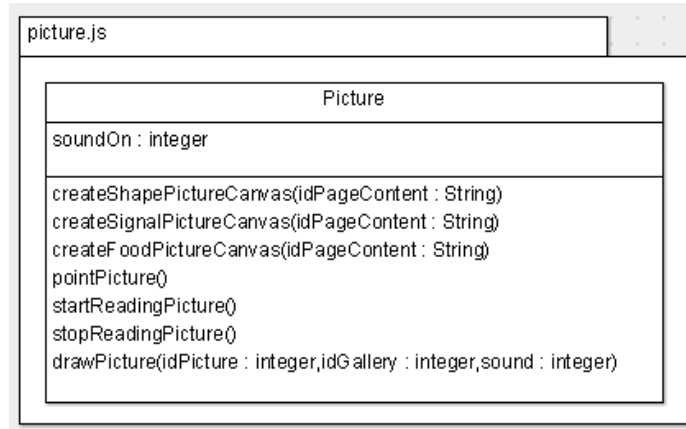


Figure 4.5. picture.js diagram

In this module the canvas to paint the pictures are created. There are three functions for the canvas element, one per gallery. The functions `pointPicture()`, `startReadingPicture()` and `stopReadingPicture()` correspond to the definition of the event handlers to detect when the user is or is not touching on the screen to start/stop reading the picture.

The last module that will be analyzed is **result.js**:

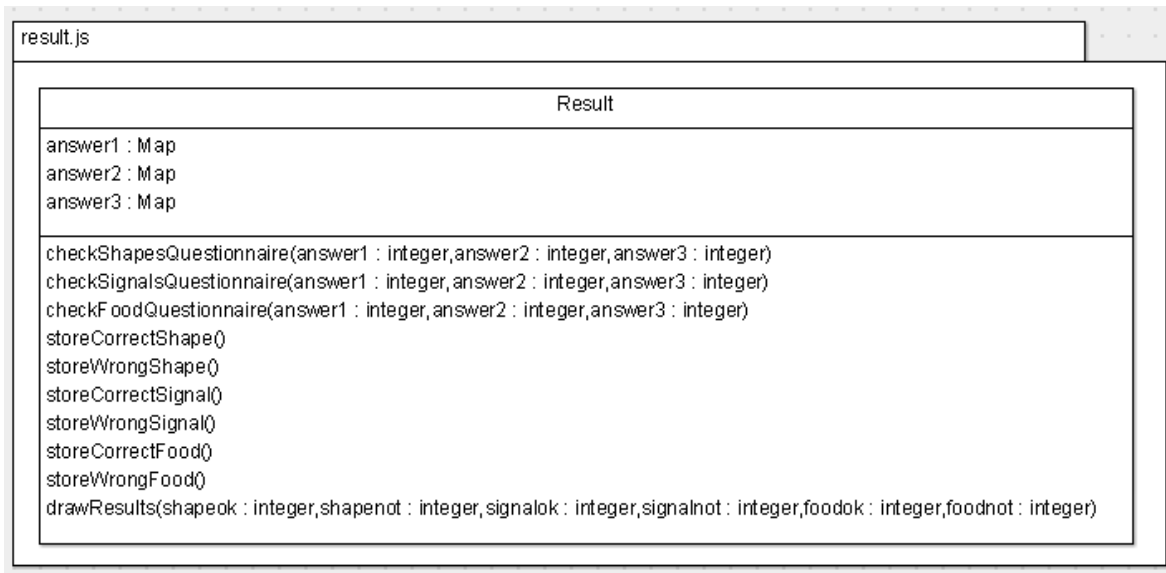


Figure 4.6. result.js diagram

This module will be the responsible to check the answers of the gallery questionnaires depending on the picture read. These results will be store in each gallery results and they could be checked at the end and draw the results on the screen of the application.

4.3. INTERFACE

In this section the interface of the application is going to be explained, accompanying the explanation with the ideas of its design and some graphics that will define what it is going to be developed lately.

Due to this application is just a beginning of something bigger that could be implemented in the future, for the moment it does not include a specific interface for non-visual users. They will need the help of someone else to guide them while navigating on the application. Although it is not specific for them, the interface will be simple and easy to use, avoiding the excess of elements in a page, and placing the buttons of the navigations in the corners following these patterns:

- The **'Home'** button will be always placed in the left corner at the top of the screen. If we are inside a Gallery, this button will be replaced for a **'Menu'** button to go to the **'Gallery Menu'** page, and then the user will have the **'Home'** button again.
- The **'ON'** and **'OFF'** buttons will be always placed in the right corner at the top of the screen but only in two cases: when the user is inside the training or when the user is reading a picture.
- The **'Next'** and **'Previous'** buttons will be always placed in the left and right corners at the bottom of the screen, but they will only appear in the training or in the galleries.

Now the pages that compose the application are presented, following the graphics with a little explanation of the elements that are inside of them.

Firstly, when the application will be started, the user will access to the main page:



Figure 4.7. Splash Screen

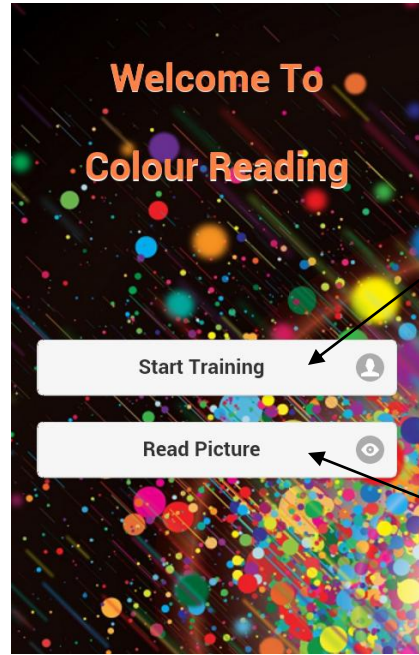


Figure 4.8. Main Page

Button to access to the training page

Button to access to the galleries page

If the user access to the training, the canvas element of the training will be created in the page, where the application will insert the colours, so later the audio and the vibration can be played. There will be the following pages in this order:

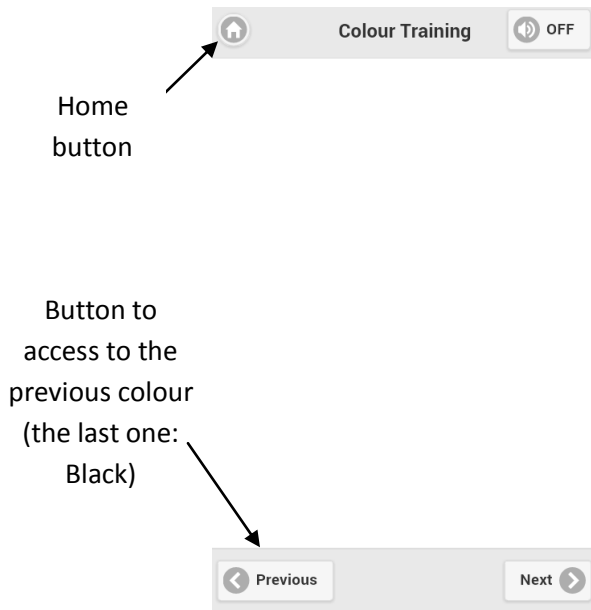


Figure 4.9. White Training

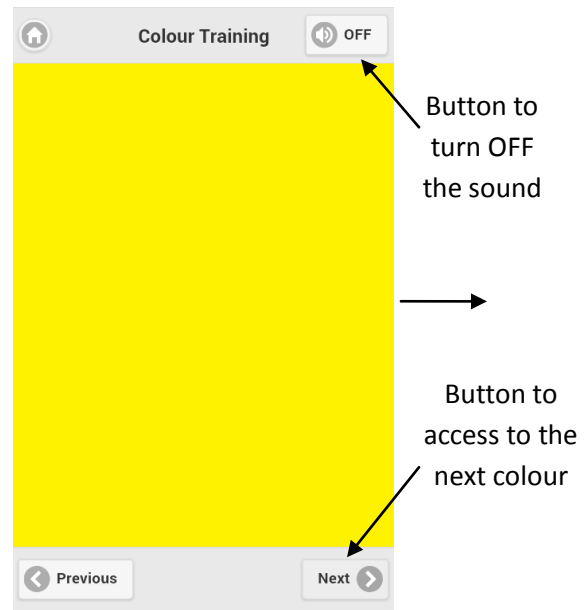


Figure 4.10. Yellow Training

Home button

Button to access to the previous colour (the last one: Black)

Button to turn OFF the sound

Button to access to the next colour

If the user presses the 'OFF' sound button, this button will be hidden and the 'ON' sound button will be shown. In addition, each time the user presses the 'Next' or 'Previous' buttons, the canvas element will be refreshed to show the next/previous colour of the training. This will happen in each colour page of this training.

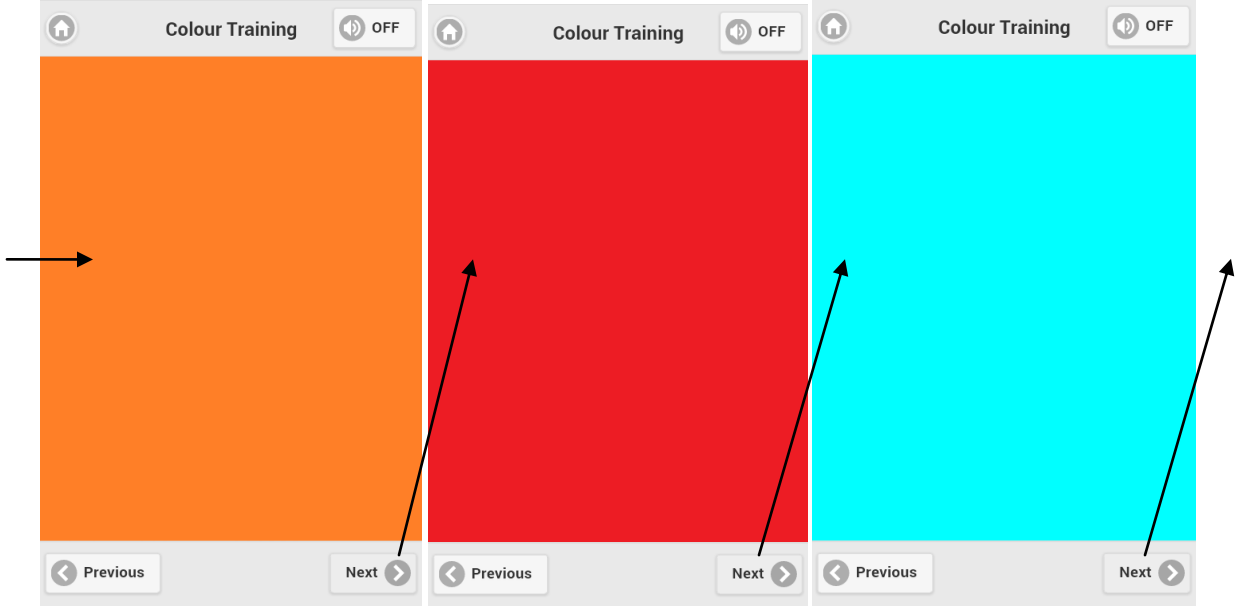


Figure 4.11. Orange Training

Figure 4.12. Red Training

Figure 4.13. Light blue Training

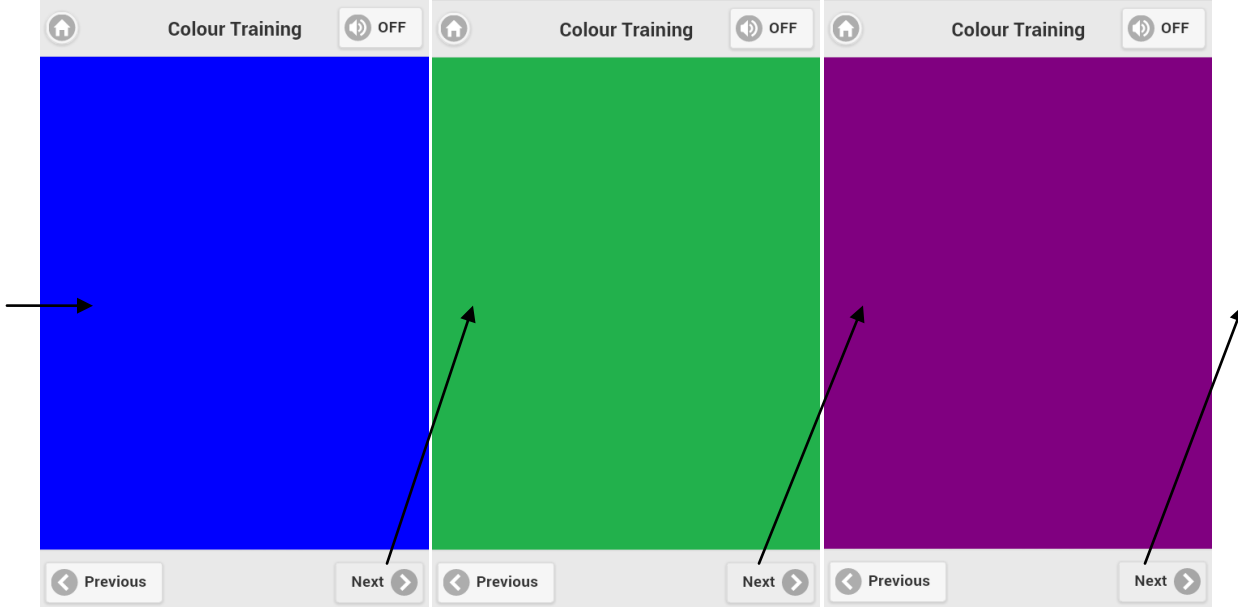
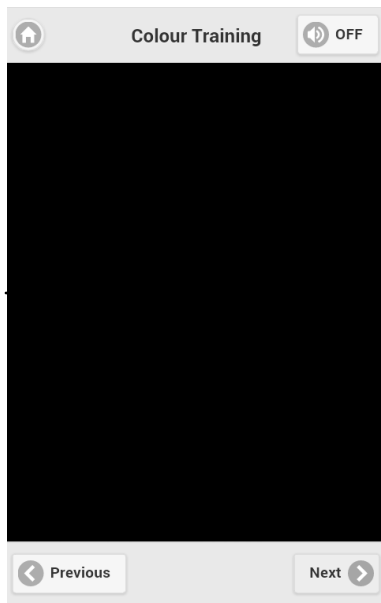


Figure 4.14. Dark blue Training

Figure 4.15. Green Training

Figure 4.16. Purple Training



If the user presses the 'Next' button after Black colour, he will start again in White colour.

Figure 4.17. Black Training

If the user access to the 'Read picture' option, he will access to the Read Picture menu where there is the list of the three galleries available:

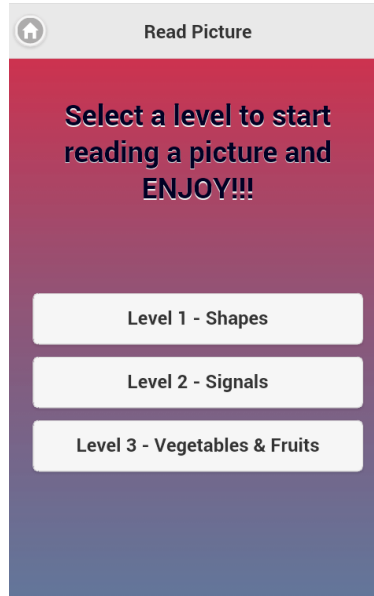


Figure 4.18. Gallery Menu

If the user selects the first gallery (Level 1 – Shapes Gallery), the canvas element of the shapes gallery will be created in the page, where the application will insert the pictures that compose that gallery. Later, the user will be able to select one of them to start reading:

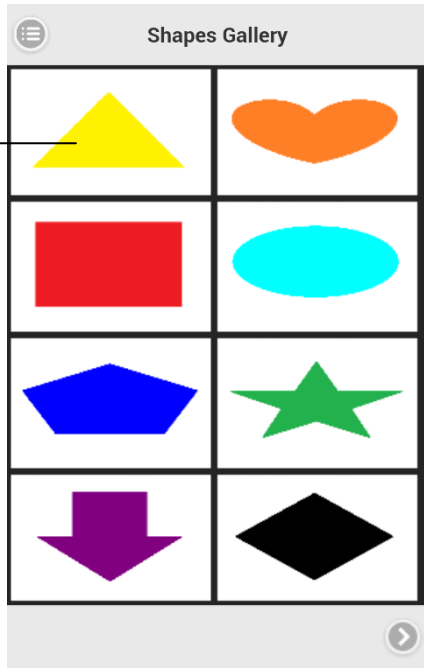


Figure 4.19. Shapes Gallery Page I

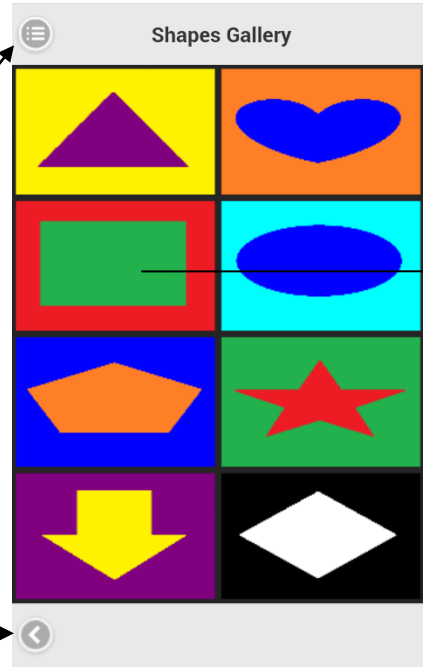


Figure 4.20. Shapes Gallery Page II

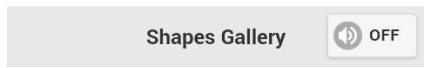


Figure 4.21. Yellow Triangle

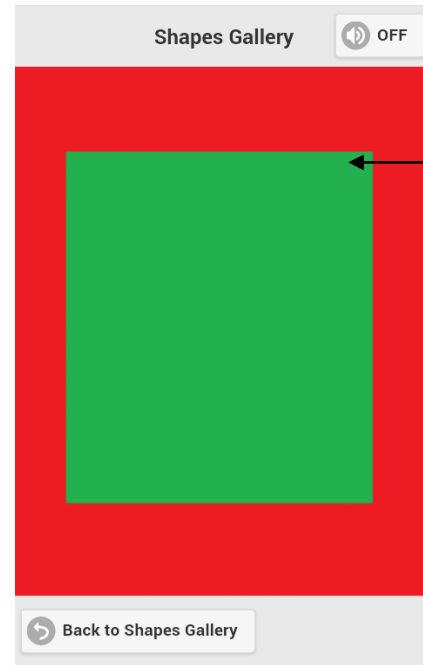


Figure 4.22. Green Square

When the user presses on the 'Back to Shapes Gallery' button he will access to another page where it will appear a questionnaire that must be answered. The user will have to answer the

questions by selecting an option for each question and then he will have to press the answer button. Depending on how the answer is correct or wrong, it will appear two different messages:

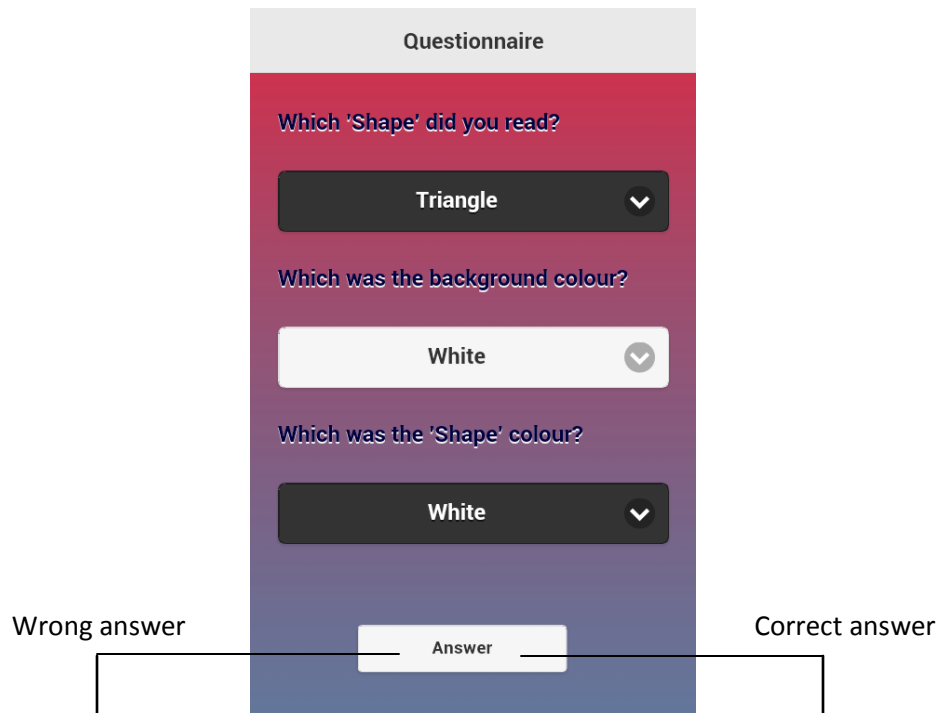


Figure 4.23. Shapes Gallery Questionnaire

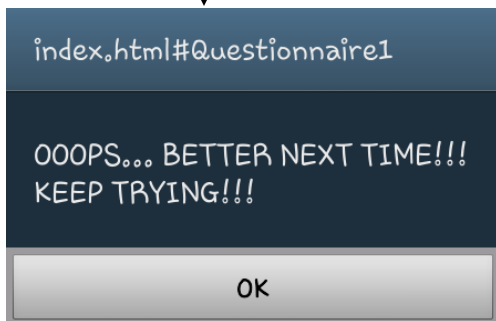


Figure 4.24. Message for a wrong answer

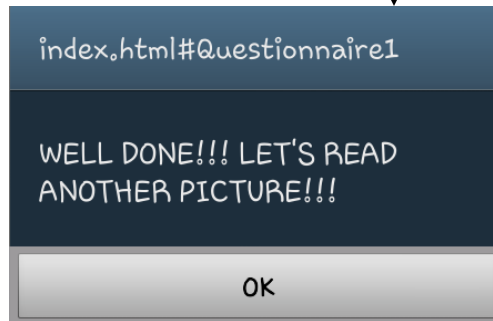


Figure 4.25. Message for a correct answer

If the user selects the second gallery (Level 2 – Signals Gallery), the canvas element of the signals gallery will be created in the page, where the application will insert the pictures that compose that gallery. Later, the user will be able to select one of them to start reading:

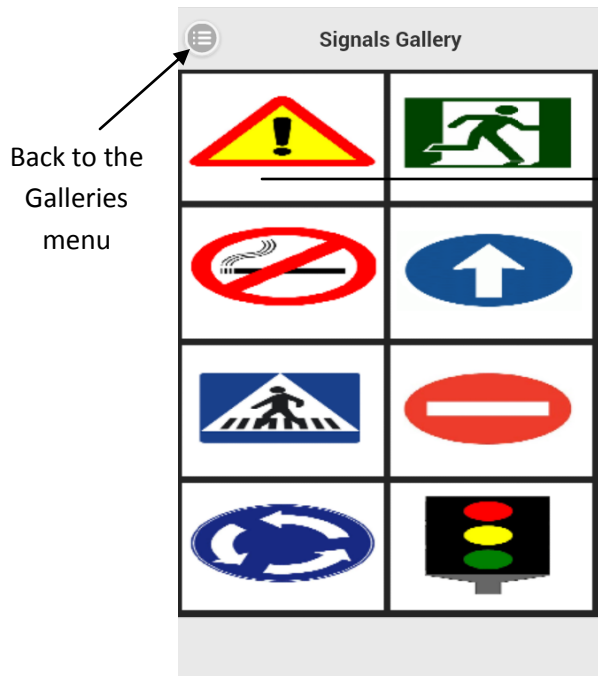
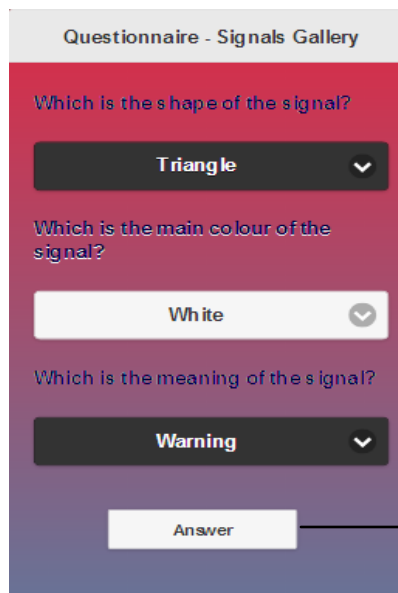


Figure 4.26. Signals Gallery Page 1



Figure 4.27. Warning Signal

When the user presses on the 'Back to Signals Gallery' button he will access to another page where it will appear a questionnaire that must be answered. The user will have to answer the questions by selecting an option for each question and then he will have to press the answer button. Depending on how the answer is correct or wrong, it will appear two different messages:



The messages for correct or wrong answers are the ones shown in Figures 4.19 and 4.20

Figure 4.28. Signals Gallery Questionnaire

If the user selects the third gallery (Level 3 – Food Gallery), the canvas element of the food gallery will be created in the page, where the application will insert the pictures that compose that gallery. Later, the user will be able to select one of them to start reading:

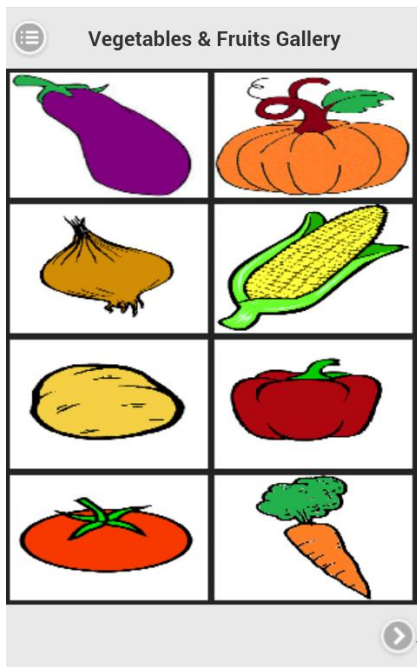


Figure 4.29. Food Gallery Page I

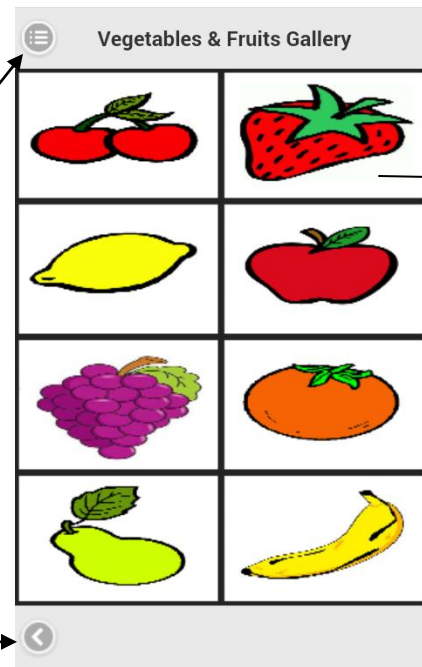


Figure 4.30. Food Gallery Page II

Back to the Galleries menu

Navigation between pages

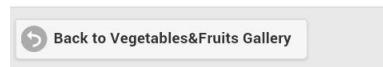
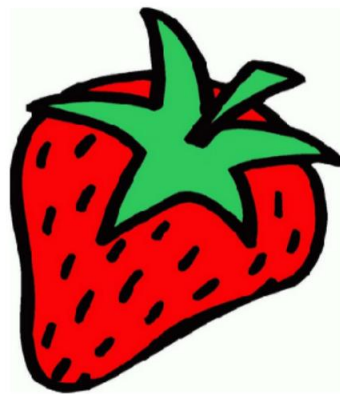
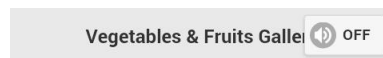


Figure 4.31. Strawberry Picture

When the user presses on the 'Back to Food Gallery' button he will access to another page where it will appear a questionnaire that must be answered. The user will have to answer the questions by selecting an option for each question and then he will have to press the answer button. Depending on how the answer is correct or wrong, it will appear two different messages:

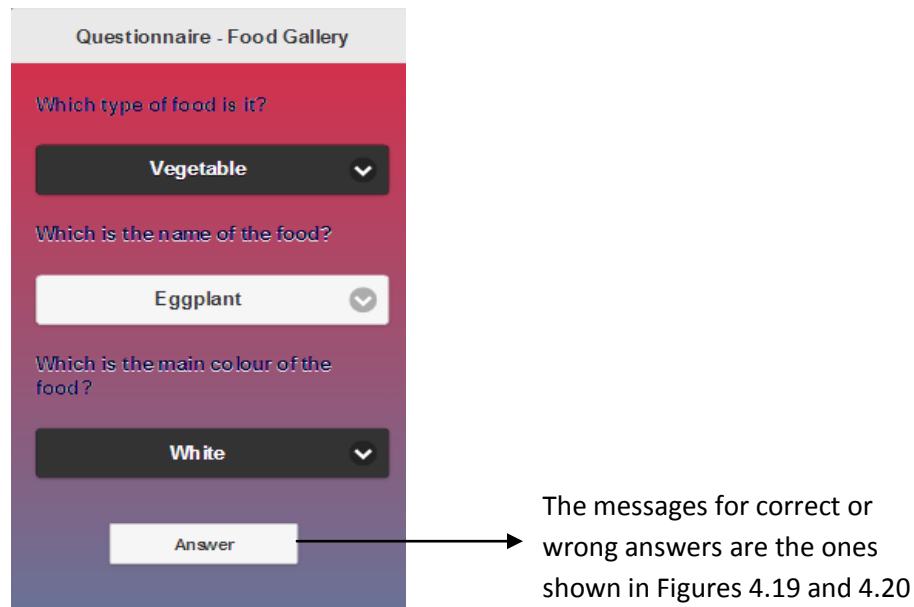


Figure 4.32. Food Gallery Questionnaire

When the user has read at least one picture, in the main page will appear a new button to check the results. If the user presses the button he will access to a page where the results of each gallery will be presented as can be seen in the following figure.

If the current user has visual disabilities, he will need the help of someone else to check the results. Anyway, this functionality of the application is for developers when doing the user testing. These results will be used to check the usability and effectiveness of the application.

4.4. CONCLUSIONS

In this section the ideas previously explained in detail are being resumed as a synthesis of the decisions made that will affect the 'Implementation' chapter that is coming after this. The following statements are highlighted:

- A designed for every module of the application have been created, and the interaction and communication between them as well.
- The data to be used in the application will be stored in XML files. Each file will have different data stored. For example, there will be one for the range colours, other for the sound files, etc.
- The data will be charged when the application is started to avoid overcharging the local memory and a slow execution of the application.
- It will be used the following programming languages: HTML5, JQueryMobile, JQuery and JavaScript because of the advantages that they can offer based on the objectives and requirements set in chapter 3.

5

Implementation

During this chapter all the details of the design of the project are going to be presented, based in the requirements catalogue exposed in chapter 3. So along this section the following items are going to be explained:

- How the code has been structured and developed, but more specifically which is the content of each module and how have they been organized.
- How is the structure of the XML files. These files will be stored each colour and their ranges, the name of the audio files or the name of the pictures that compose the galleries of the application.
- How is the structure of the pages that compose the application and how it has been created with the use of HTML5 and JQueryMobile.
- How the functions of the plugin work on the application and how they allow the developer to set a vibration.
- Which problems have appeared along the development process and if they were sort out, how we did it.

5.1. CODE STRUCTURE

During this chapter we are presented the structure and organization of the code. This part is really important in order to a future maintenance, mostly if the future developer is a different person.

Along this paper it has been mentioned that it would be nice if the functionality of this project could be improved and expanded, because this technology area has to be deeply studied and in detailed to achieve huge and revolutionary improvements.

The organization followed in the creation of the JavaScript modules is a division of the functionalities depending on the requirement catalogue. In total there are five modules, four of them corresponds to one of the subsections in the catalogue, so its module has the name of the subsection: “*subsection.js*” (for example: *training.js* or *results.js*). The fifth module is a combination of the catalogues of ‘Colour Management’ and ‘Sound Management’ in a file called ‘*colours.js*’. We are explaining those modules below:

colours.js: This JavaScript module is the one that contains the functionalities of the Colour and Sound Management requirements. More specifically in this module can be found:

- The range colours. Those will be stored in variables to simplify the code, so the application does not have to access to the XML file each time it wants to read a colour.
- The definition of each colour delimited by their ranges. As we have mentioned before, nine colours will be defined and with their ranges we are trying to obtain from the lighter till the darker tones of each colour, so the scope of detected colours will increase.
- The functionality to detect the colour of the coordinate of the screen where the user is touching in a punctual time. It has to be considered that the user will be moving the finger constantly to try to read a picture, so the application has to be prepared to read the coordinates at each minimal movement of the user.
- The vibrations and sounds associated with each colour, so as soon as a colour is detected, the application will play the vibration plus the sound. These vibrations and sounds will be also stored in variables to simplify the code, as mentioned before.

training.js: This JavaScript module is the one that contains the functionalities of the Training Management requirements. More specifically in this module can be found:

- The creation of the canvas element of the page where the colours of the training will be painted. This canvas has to be the same as the size of the screen available without counting the header and the footer of the page. That is, this functionality will have to consider the size of the screen to delimit the size of the canvas. The developer has to be careful with treating the size of the screen because it may change from one device to another.
- The functionality to detect when the user wants to change the colour that is reading to the next or the previous colour. The order of the colour training is explained in chapter 4, interface design.

gallery.js: This JavaScript module is the one that contains the functionalities of the Gallery Management requirements. More specifically in this module can be found:

- The creation of the canvas element of the page where the galleries of the training will be painted. Like in the training module, the canvas has to be the same as the size of the screen available.
- The creation of the gallery based on the option selected in the menu of galleries where the user can select three levels of difficulty.
- The functionality to detect when the user selects a picture of the gallery to communicate with the picture module and let the user read a picture.

picture.js: This JavaScript module is the one that contains the functionalities of the Pictures Management requirements. More specifically in this module can be found:

- The creation of the canvas element of the page where the pictures from the galleries will be painted. Like in the training module, the canvas has to be the same as the size of the screen available.
- The functionality to read the picture selected. This functionality can only be implemented with the help of an event listener to detect when the user is touching the screen or when has stopped touching. This event will reference to the colour module to play the vibration and sound of the colour detected.
- The functionality to detect when the user has finished reading the picture to communicate the application that the user must answer the questionnaire and then called the results module to check the answers.

results.js: This JavaScript module is the one that contains the functionalities of the Result Management requirements. More specifically in this module can be found:

- The functionality to detect the answers selected by the user in the questionnaire of the read picture and to check them with the correct ones.
- The functionality to store the correct or wrong result so at the end of the application execution the user will be able to check its results and the percentage of success or failure.
- The creation of the canvas element of the page where the results will be shown to the user. Like in the training module, the canvas has to be the same as the size of the screen available.

Apart from these modules, the application is using the **immersion.js** file that contains the constructor, the function to register its library and the proper functions of the Immersion UHL Plugin. In this file we can find the two functions that are going to be constantly used for the application to play and stop the different vibrations when the user touches the screen and the application detects a colour that has to be read. These functions are:

Immersion.worker.playEffect(index): This function receives the index of the vibration that has to be played. This index is set by the colour module when each colour is defined. When the application is reading a picture and detects a colour it calls these function to start reproducing its vibration.

Immersion.worker.stopEffect(): This function does not need to receive an index because it automatically stops any vibration that is being played. When the application detects that the user is not touching the screen to continue reading a picture, it calls this function to stops the vibration.

5.2. XML FILES STRUCTURE

Along this chapter it will be explained the structure of the XML files that compose the application. As has been previously mentioned, there are XML files for:

- The range colours
- The audio files of each colour
- The colour files of the training
- The picture files of the Shapes Gallery
- The picture files of the Signals Gallery
- The picture files of the Food Gallery

The XML file for the range colours, **ColourRangesXML.xml**, is composed by the low and high values between the RGB values can be each colour of the application. Below can be found the code corresponding to one colour:

```
<ranges>
  <colour>
    <name>White</name>
    <index>1</index>
    <r1>210</r1>
    <r2>255</r2>
    <g1>210</g1>
    <g2>255</g2>
    <b1>150</b1>
    <b2>255</b2>
  </colour>
</ranges>
```

Figure 5.1. Ranges of a colour in ColourRangesXML.xml file

In this file, apart from the ranges of the RGB values, it is also stored the name of the colour and its index, for a better comprehension of the code for future developers.

The XML file for the audio files, **SoundsXML.xml**, is composed by the name of the audio files of each colour. Below it can be found the code corresponding to these audio files:

```

<audio>
  <colour>
    <name>White</name>
    <file>sounds/white.mp3</file>
  </colour>
</audio>

```

Figure 5.2. Audio file of a colour in SoundsXML.xml file

In this file, apart from the URL of the sound file, it is also stored the name of the colour for a better comprehension of the code for future developers.

The XML file for the colour training files, **TrainingColoursXML.xml**, is similar to the previous XML file, but this time with the URL of the colour pictures:

```

<training>
  <colour>
    <name>White</name>
    <file>./media/white.png</file>
  </colour>
</training>

```

Figure 5.3. Colour file of the training in TrainingColoursXML.xml file

The XML file of the three different galleries, **ShapesGalleryXML.xml**, **SignalsGalleryXML.xml** and **FoodGalleryXML.xml**, is composed by the name of the picture files that compose each page of the gallery. Below it can be found the code corresponding to two picture files in one gallery page:

```

<shapes>
  <page>
    <picture>
      <id>0</id>
      <url>./media/shapes/triangle.png</url>
    </picture>
    <picture>
      <id>1</id>
      <url>./media/shapes/heart.png</url>
    </picture>
  </page>
</shapes>

```

Figure 5.4. Pictures files on a page of the ShapesGalleryXML.xml file

In this file, apart from the URL of the picture file, it is also stored the id of the picture that will be used to place the picture in its corresponding position in the gallery.

5.3. INTERFACE WITH HTML5-JQUERYMOBILE

In this section it is going to be explained how has been structured the interface of the application and the pages that compose this application.

The structure of the application is created in the file called **index.html**. This file has to follow the structure that determines the HTML5 language, which is the language we are using to build the PicRead application. It follows this structure:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Application Title</title>
</head>
<body>
  <header></header>
  <footer></footer>
</body>
</html>
```

Figure 5.5. HTML5 basic structure

In the **<head>** part of the file, the different modules that compose the application must be defined. Based on the application developed in this project, the modules that have to be added are the ones that were explained in the first section of this chapter. To these modules, it has also to be added the libraries mentioned in chapter 4 to use JQuery and JQueryMobile:

```
<script src="js/colours.js"></script>
<script src="js/training.js"></script>
<script src="js/gallery.js"></script>
<script src="js/picture.js"></script>
<script src="js/result.js"></script>
<script src="js/immersion.js"></script>

<script type="application/javascript" src="js/jquery.min.js"></script>
<link rel="stylesheet" type="text/css" href="jqm/jquery.mobile-min.css">
<script type="application/javascript" src="jqm/jquery.mobile-min.js" data-ver="0"></script>
```

Figure 5.6. Modules references of the PicRead application

In the **<body>** part of the file, the pages of the applications and the components that are on each of them are defined. In this part is where JQueryMobile is going to be used. PicRead is going to be composed by ten different pages:

- The Main page
- The Colour Training page
- The Gallery Menu page
- The Shapes Gallery page

- The Shapes Questionnaire page
- The Signals Gallery page
- The Signals Questionnaire page
- The Food Gallery page
- The Food Questionnaire page
- The Questionnaire Results page

Each of these pages has to be defined like in Figure 5.4, where the attribute that is on the tag 'class' refers to the background to be set in that page, the 'id' will be the label identification that has to be unique in all the application and the 'data-role' of that component, that is a page:

```
<div class="upage MainPageBK" id="mainpage" data-role="page">
```

Figure 5.7. Definition of a page with HTML5/JQueryMobile

Inside this page many elements can be added. The structure of the page is quite similar to the <body> part of the HTML5 file. It can have different sections but the commonly used ones, and the sections that are being included in the pages of this application are: header, content and footer.

In the <header> part of the pages there are: the title of the page, a back or a menu button and the off/on sound button which were shown in the interface design in chapter 4. An example of a header definition can be found below in Figure 5.5:

```
<div data-role="header" class="container-group inner-element uib_w_4"
  data-uib="jquery_mobile/header" data-ver="0">
  <h1>Colour Training</h1>
  <div class="widget-container wrapping-col single-centered"></div>
  <div class="widget-container content-area horiz-area wrapping-col left">
    <a class="widget uib_w_6 d-margins" data-uib="jquery_mobile/button" data-ver="0"
      data-role="button" data-icon="home" data-iconpos="notext" id="BTHome"
      href="#mainpage" data-transition="fade">Home</a>
  </div>
  <div class="widget-container content-area horiz-area wrapping-col right">
    <a class="widget uib_w_43 BSoundMargin" data-uib="jquery_mobile/button" data-ver="0"
      data-role="button" data-icon="audio" id="BTON1">ON</a>
    <a class="widget uib_w_42 BSoundMargin" data-uib="jquery_mobile/button" data-ver="0"
      data-role="button" data-icon="audio" id="BTOFF1">OFF</a>
  </div>
</div>
```

Figure 5.8. Definition of the header element on a page with HTML5/JQueryMobile

As it can be appreciated on Figure 5.5, to add an element on a page, for example a button, it has to be added on a widget-container. In this case, there is one at the left side for the 'Home' button that refers to the main page (*href="#mainpage"*) and two at the right side: the 'ON' sound button and the 'OFF' sound button. Later, with the use of JQuery, it will be able to manage when these buttons have to be shown or hidden, because they cannot be shown at the same time. These buttons has the data-role of button.

The `<content>` role is empty in the majority of the pages because it is where it will be placed the canvas element to host the galleries or pictures of the application. In these cases, the content section it is only declared to associate it with an id that will be used in the creation of the canvas.

```
<div id="StartTrainingsub" class="upage-content ">  
</div>
```

Figure 5.9. Definition of the content element on a page with HTML5/JQueryMobile

However, the content can also be formed by buttons or text area. In those cases, the declaration of the elements follows the same pattern as in the header section: they have to be added in a widget-container. The `<footer>` section is also similar to the `<header>` element. The only difference is that one element is located at the top of the screen (header) and the other one is at the bottom of the screen.

The navigation between pages it is defined in the button declarations because they are the responsible to access to the different pages of the application. To add other functionalities to the buttons it has to be developed on a `<script>` section with JQuery language in the `<body>` tag of the HTML5 file (where the pages are declared). JQuery is very simple and easy to use, and it has an important role in the application when it is needed to hide or show some buttons on a page.

5.4. CONCLUSIONS

It has been developed the modules defined in section 5.2, the connections between them has been set and the interface has been created as defined in section 5.4, but it has been found many difficulties in its developing process.

The use of the different languages was the most complicated part at first because it was necessary to learn how to use them, how to call functions or how to structure the code. To achieve this, Internet has been an important tool to consult and obtain information about the problems that were appearing, mostly with information and advices found in programming forums.

6

Implantation and Tests

During this chapter two items are being presented: the implantation on different Android devices and the tests battery used during and after the developing process.

Both of these parts are important for the project. The first one because is one of the non-functional requirements described in chapter 3, RFN1. The other, the tests battery it is needed to check that the functionalities developed in PicRead corresponds to the functional requirements explained in chapter 3 too. These items will check that the objectives of the project have been achieved.

6.1. IMPLANTATION

The implantation of the application has been tested mainly on a Samsung Galaxy S II, with a 4.1.2 Android version. The application has been built with the Intel XDK tool for Windows mentioned in chapter 4 and from it the application has been sent to the Android device.

Before starting with the development of the application, an application built with just the Immersion UHL Plugin was installed on the Android device to check if the device was compatible with the plugin. As mentioned in chapter 2, it is needed that the device has integrated the TouchSense Player in order to be able to play the vibrations of the plugin. The application test was successfully installed and played on the Samsung device so it was defined as the main device tester.

The opportunity of having an Android device available during all the development process has been satisfactory to make continuous tests while functionalities were being included or the interface was being modified in the application.

Due that every device has its own dimensions [12] the application had to be adapted to fit in the screen of the device as mentioned in the non-functional requirement **RFN2**. To check if the application accomplished this requirement, PicRead was tested in other Android devices:

- Huawei G510, with a 4.1.1 Android version
- HTC Desire X, with a 4.1.1 Android version
- Samsung Galaxy S II, with a 4.0.4 Android version
- Samsung Galaxy S Plus, with a 2.3.4 Android version
- Samsung Galaxy Core, with a 4.2 Android version

6.2. TESTS BATTERY AND RESULTS

Apart from the implantation, in these sections the tests made on the product will be explained to determine if the objectives and requirements, functional or non-functional, have been accomplished.

Different types of tests have been taken place along the development of the project:

1. **Unit Testing:** Also known as component testing. It corresponds to the tests that verify the functionality of the modules that compose the application. These tests have been taken place during all the developing process.
2. **Integration Testing:** It is a software testing whose aim is to verify the communication between modules and the data transfer between them. This testing also works to detect the defects in the interface.
3. **System Testing:** Also known as end-to-end testing. It tests a completely integrated system once the product is implanted on the Android device to check that the requirements exposed in chapter 3 have been achieved and that it can be ran correctly on the device.
4. **User Testing:** This testing has been done with real impair visual users. The utility and interaction have been the two main items that have been tested and evaluated.

Below, these types of testing are explained more in details, presenting how these tests were made, its correspondent results and which were the results expected.

6.2.1. Unit Testing

The aim of this type of testing is to verify that the developing process of each module is being correct. The way to check this is by testing parts of the code in each module. For example, by testing function by function on a module.

The main objective is to detect implementation mistakes to avoid errors in future tests, for example in integration or system testing. If the code is checked function by function, a module will trust on the functions developed in other module, because it will know that those functions have already passed the unitary tests.

The development of the application and the subsequent tests have been sequentially created based on the necessities that were appearing while developing the application. In order to make these tests, it was created an html module in which the new functionalities developed were being tested. When the functionality was not the expected, it was used the debug option in the Intel XDK to check where the error was happening. This debug option allows the user to debug it also once the user has the application installed on the device.

Before creating the interface of the application, it was only composed by one page with a canvas element. It has been mentioned along the paper that the main functionality of the application happens on the canvas elements. For this reason, the first thing that was implemented and checked was the correct creation of the canvas element depending on the screen size of the device. Once it was verified that it worked correctly, the next step was to insert an image on the canvas, function that can be found in the module **picture.js** and then the function to detect coordinates and colours which can be found in the **colour.js** module. In Figure 6.1 can be seen a test to obtain the RGB values (red, green and blue) and the correspondent colour:



Figure 6.1. Coordinates and Colour Test

Another important unitary test was to verify that the application was detecting the correct place where the user was touching on the screen. The solution to check this was to paint on the screen the path of the user's finger as can be seen in Figure 6.2.



Figure 6.2. User's finger path Test

Due to all the unitary tests were passed, the functional requirements explained in chapter 3 have been successfully achieved.

6.2.2. Integration Testing

The aim of this type of testing is to verify the communication between the interface, the modules of the application and the data transferred between them.

Before this testing can be executed, the different modules that compose the application were created and developed. The interface was also implemented for this testing because from it, the different functions of the modules are called to create the elements that will be included in the canvas components of the pages. These functionalities were presented in chapter 5, Implementation.

In general, these tests were successfully concluded thanks to the previous execution of the unitary tests in each function of each module that compose PicRead. Thanks to this, the errors found were easier to sort out.

The test with more problems was the one of checking the results of the questionnaire and storing its results. For this functionality there are three modules affected: the results module that calls the check function, the picture module to give to the result module the correct answers, and the interface module that transfers the data from one module to another. There was not a solution adopted for this, but it was needed to be careful with the function calls and data transmission.

Another problem that has been found in the integration testing was in the navigation between galleries. The first time in entering to a gallery, it was created perfectly. However, if another

gallery was accessed after that one, and then return to the first gallery, the pictures were not drawn in the gallery. This was happening because every gallery was using the same canvas element, so the solution was to create three different canvas elements, one for each gallery.

Once these problems were solved, the integration testing was repeated to verify they were not happening again.

6.2.3. System Testing

This testing was done each time a prototype of the application was built in order to check the correct integration between the modules and the interface, and to check that the application was adapted to the Android device where it was going to be installed.

As it has been defined in the non-functional requirement **RFN2**, the size of the pages of the application had to fit perfectly on the screen of the device. It is not accepted the example shown in Figure 6.3. If happens what is shown in that figure, it means that the application has not a responsive design and the system testing will not pass.



Figure 6.3. Example of non-responsive application

The tests regarding to this aspect were made in the devices mentioned in the Implantation section of this chapter.

In these test are also included the ones for the components of the application. It was checked for example that the buttons of the application were not overlapping with other components of the application, for example, with the headers' title.

Apart from the components of the interface, the system testing has to undertake the tests about operability and efficiency of the application, evaluating for example that the application is:

- Not corrupting shared memory
- Not consuming excessive resources
- Not delay page transitions with data exchange
- Not leaving any parallel processes

With these tests a problem was found about the page transitions. JQueryMobile has plenty of advantages for developing app interfaces with HTML5, but it also has some disadvantages that affects in the page transitions introducing white blinks occasionally.

6.2.4. User Testing

This testing was made with real visual impair users with no previous familiarization with the PicRead application. The following aspects will be evaluated on this section:

- The fulfilment of each functional and non-functional requirement described in chapter 3.
- The adaptation to the application.
- The correct learning of the sounds and vibrations used to identify the colours in the application.

These tests have been made with two real users: Sirilo and Azra. In order to be able to contrast the results obtained, the tests were different for each one. Sirilo had the chance to test the application by using sounds and vibrations, but Azra only tested it with sounds.

Below are explained the results obtained:

- Both agree that the sounds accessibility integrated in the application helps them to navigate through it. So the non-functional requirement **RFN3** was successfully achieved. However, they both agree that it would be nice to include the screen readers functionality that offers mobile devices.
- According to the sounds integrated in the application for each colour, on one hand Azra was comfortable with them and she learnt the association easily. However, for Sirilo it was a bit more complicated but not impossible. In his opinion, he would have preferred sounds that could evoke feelings or that could be easily associated with real things. For example, the green colour with nature, the blue colour with the sea, etc. The non-functional requirement **RFN7** has been achieved too, but it would be needed to create a sound system that please everyone.

- The interest for the application has been noteworthy in both of them. They were excited about the idea of touching colours and guessing graphics. Sirilo pointed out one advantage about this vibration functionality and is that it can be quite useful in places where it is not possible to hear, for example a crowded place.
- According to guessing the graphics of the application there are two different results for each user. Sirilo, after fifteen minutes practicing and learning the colours, was able to detect each colour and the easiest graphics of the Shapes Gallery: circle, square and triangle. With the heart picture for example he was able to detect vertexes but not the whole shape. On the other hand, Azra was able to detect the colours too, but she found a bit difficult to guess the graphic without vibrations. With these two results, a conclusion can be extracted: vibrations are important to let the user feel with the finger where are the colours changing and with it, the vibrations to delimit the form of the graphics.

The user testing has been really satisfactory, both in personal and professional sense. In a professional sense, this testing has helped to extract future improvements for the application so it can be developed to be useful in users' daily life in a nearby future. In a personal sense, this testing has been important because after seen the interest on the application, it motivated to continue with the project.

7

Conclusions and Future Work

This section of the paper is about the personal conclusions obtained after the development of the project. It is important as a constructive criticism of the work made, the results obtained and the features or functionalities that could be improved for future development of the application.

Regarding to the objectives exposed at the beginning of the paper, they have been accomplished successfully. The goal was to explore how the combination of haptic technologies with sonification can, potentially, support non-visual individuals to interpret colours and figures through their mobile devices. As mentioned in the '*User testing*' section in chapter 6 the user is now able to detect the colours and figures by the combination of vibrations and sounds, although the figures recognition depend on the level of difficulty they present. There is a lot to do yet in this technology field, but the first step has been successfully achieved.

I am quite proud of the work done. I have learnt lots of things during the development of the project, but also when I did the tests with impair people. It is fascinating how interested they are in new technologies and how they use the mobile devices thanks to the accessibility of screen readers integrated on the devices. Here is where I would like to introduce improvements in the future. The current application has sound support as mentioned along the paper, but it would be interesting to include the screen readers accessibility to the application in order to make it more comfortable for the user.

After the meeting with Sirilo, the impair user that test the application, there is one thing that could be interesting to think about, and it is about the sound effects. In the non-functional requirements it was mentioned that the sounds should be easy to distinguish and memorize. Currently, the sounds available fulfil the requirements, because they are completely different from each other. However, Sirilo mentioned that those sounds could evoke feelings or thought that make easier to memorize the relation colour-sound.

To the functionalities currently available there are many things that could be added. Below I am mentioning the ones that will be included in the nearby future:

- **A camera option to take pictures.** Let's think about a situation in which the user has three different cans of refreshments: coke, lemon and orange. The cans are all of the same size, so they cannot be differentiated from each other by touch. If the application includes the camera option, he can take a picture and read it to detect the colour. If it is red, he will know that it is a coke can. Yellow will refer to the lemon can, and orange colour to the orange can. In this case, this functionality will allow the user to use the application in its daily life.
- **A sound customization.** The sound integrated in the application could not please each user. For example, in Sirilo's opinion the sounds could be from music instruments, but for other user the bell sounds that currently integrate the application could be better. For this reason it would be interesting to include a sound setting so then the user could customize and decide which sounds he wants to include to recognize the colours.
- **A device gallery button.** With this button the user will be able to access to his device gallery so he will be able to read the pictures that are already stored in his device.

Regarding to technical aspects a good idea could be adapt the application to other operating systems, such as iOS, so it can be executed in other platforms apart from Android devices. Probably it will be needed to change the configuration of the Immersion UHL plugin too.

If I think about the developing process I have followed, I have to say that the beginning was complicated. At first I found the topic of the project quite interesting, but at the same time complicated and wide. It was difficult to imagine how I could reach the objectives and the product we have created, but the result has been satisfactory. I have learnt a lot about this technology area and I am amazed with the things that can be achieved with effort and dedication. I have created a project that can be further improved to help people that already need it and I had support from professional people in this area that guided me and encouraged me to continue with this beautiful project.

Bibliography

- [1] Azra Tayebi, Carolina Islas Sedano. (2014). *What can your ears do? A systematic literature review regarding the role of non-speech audio*. University of Eastern Finland, Joensuu, Finland.

- [2] Mehryar Nooriafshar. (2004). *A Multi-sense Approach to Information Reception and Knowledge Creation in Learning*. University of Southern Queensland Toowoomba, Australia.

- [3] *Brain based learning with the senses, hearing, vision, AND smell, touch, and taste in the classroom*. <http://kantrybe.hubpages.com/hub/Brain-based-learning-in-the-classroom>

- [4] *Software Development Life Cycle*.
http://es.wikipedia.org/wiki/Systems_Development_Life_Cycle
<http://www.techopedia.com/definition/22193/software-development-life-cycle-sdlc>

- [5] *Haptic Technology*. http://en.wikipedia.org/wiki/Haptic_technology

- [6] *Kinesthesia/Proprioception*. <http://en.wikipedia.org/wiki/Proprioception>

- [7] Sara Comai and Davide Mazza. (2012). *Integrating Haptics in Web Interfaces. State of the Art and Open Issues*. Politecnico di Milano, Milan, Italy.

- [8] Gabriel Robles-De-La-Torre. (2008). *Principles of Haptic Perception in Virtual Environments*. In Grunwald M (Ed.), *Human Haptic Perception*, pp. 363-379.

- [9] Jeffrey J. Berkley. (2003). *Haptic Devices*. 4033 Aurora Ave N. Suite 201 Seattle, WA 98103.

- [10] Vincent Hayward, Oliver R. Astley, Manuel Cruz-Hernandez, Danny Grant and Gabriel Robles-De-La-Torre. (2004). *Haptic Interfaces and Devices*.

- [11] *Haptic Development Platform for Android. Immersion UHL Plugin.*
http://immersion.com/developers/index.php?option=com_content&view=article&id=457&Itemid=731
- [12] *Supporting multiple screens. Android.*
http://developer.android.com/guide/practices/screens_support.html
- [13] *Colour Tools.* <http://html-color-codes.info/codigos-de-cores-hexadecimales/>
- [14] *Effects available in the UHL Plugin.*
http://immersion.com/developers/index.php?option=com_content&view=article&id=496&Itemid=814

Appendix A: Colour Ranges

The range colours have been mentioned in numerous occasions along this paper, so in this section these values are going to be explained.

The ranges are determined by the RGB color model which is an additive model where red, green and blue values are put together to reproduce an extended array of colors. The name of the model, RGB, comes from the first letter of the colours **R**ed, **G**reen and **B**lue.

In Figure A.1 can be seen a graphic with the different tones of colour that can be obtained by using the RGB values. Each of the RGB component can have a value from 0 till 255, so currently, **256³** different colours exist.

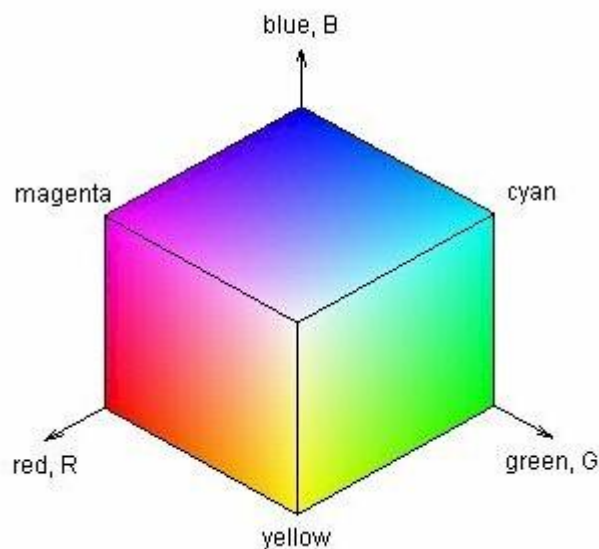


Figure A.1. RGB box values represented with x,y,z.

Due to there are plenty of colours, in the design of the application it was decided to recognize just nine colours, but including from the lighter tones till the darker tones.

The ranges for the RGB values decided are shown in Table A.1. To create these ranges, it was used the RGB box of Figure A.1, some colour tools found in the Internet [13] and lots of tests through pictures that were added to the application to verify if the application was detecting the correct colours or not.

	Low R	High R	Low G	High G	Low B	High B
White	210	255	210	255	150	255
Black	0	50	0	50	0	50
Yellow	220	255	190	255	0	110
Orange	205	255	95	185	0	115
Red	135	255	0	95	0	100
Light blue	0	165	130	200	130	255
Dark blue	0	115	0	115	60	255
Green	0	210	60	255	0	125
Purple	80	255	0	140	80	255

Table A.1. RGB ranges for the colours recognized in PicRead application