



UNIVERSITY OF BEIRA INTERIOR  
Department of Informatics

**MARA - Mobile Augmented Reality for  
Advertising  
(Discover Covilhã)**

**Andreia Ferreira**

Dissertation submitted in candidature for the degree of Master of Science in  
**Computer Science and Engineering**  
(2nd Cycle Degree)

Supervisor: Prof. Dr. Frutuoso Silva

**Covilhã, June 2015**



Dedicated to my parents, my sisters and my grandparents for always supporting me.



## Acknowledgements

It was quite a journey. I could not have done it without the help and strength from my family, friends and professors. Therefore I want to thank you all, I would not be here without you.

First, I want to thank my parents for being supportive and pushing me to not only proceed but to finish my studies. I would not have done it if it was not with their support and direction.

I want to thank to my supervisor, Professor Dr. Frutuoso Silva, for accepting to be my supervisor once more and for helping me every time that I needed, giving me suggestions and material to study which helped me improve and create a better app.

Lastly, I want to thank to the rest of my family and friends that supported me, not only during the development of this project but during both my academic and personal life.

For all of you thank you is not enough,

Andreia Ferreira



## Resumo

As aplicações para dispositivos móveis aplicadas à publicidade têm visto a sua utilização a crescer massivamente. São mais interativas, não necessitam de um grande investimento monetário e chegam a um grande número de utilizadores. Esta dissertação tem como alvo criar uma aplicação para publicidade/divulgação em Realidade Aumentada para dispositivos móveis (MARA), tendo-se optado por desenvolver uma aplicação de divulgação turística para a Covilhã.

A aplicação desenvolvida pretende dar a conhecer um percurso turístico na Covilhã, no qual é apresentada informação através da Realidade Aumentada, nos diversos pontos do percurso. Esta aplicação faz ainda uso de sistemas de geolocalização (GPS) permitindo assim guiar os utilizadores ao longo dos vários pontos do percurso.

A Realidade Aumentada (RA) funciona como um suplemento para o mundo real usando objetos virtuais criados em computação gráfica. Esta é uma tecnologia em grande expansão, com aplicações cada vez mais variadas e diversificadas, podendo ser utilizada em diversas áreas. Nesta dissertação são apresentados exemplos de aplicações em áreas como publicidade e marketing, medicina, entretenimento, militar, educação e finalmente turismo.

A aplicação descrita nesta dissertação foi desenvolvida para Android utilizando o *Apache Cordova*, *Wikitude* e *Google Maps*.

O Android é um sistema operativo (SO) para dispositivos móveis baseado no Linux. É o SO para dispositivos móveis com mais utilizadores do mundo. Por ser open source facilita aos programadores o processo de desenvolvimento de aplicações, bem como a sua publicação na *Play Store* (loja de aplicações do Android).

Sendo esta aplicação uma aplicação de browser foi necessário o *Apache Cordova* para a sua realização. Esta framework facilita a criação de aplicações para dispositivos móveis, já que estas serão desenvolvidas utilizando *HTML*, *CSS* e *JavaScript* sem ser necessário código em *Java*.

Para criar as funcionalidades da aplicação foi necessário um plugin para realizar a RA, *Wikitude*, assim como os mapas e direções, *Google Maps plugin para phonegap*, presentes na aplicação.

## Palavras-chave

Realidade Aumentada, Android, Aplicação, Apache Cordova, Wikitude



## Abstract

The use of applications for mobile devices applied to advertising has been increasing during the last few years, being more interactive, not requiring a large monetary investment and reaching a large number of users. The aim of this dissertation is to create an application in mobile Augmented Reality for advertising/divulcation (MARA), and it was decided that an application for tourism divulgation in Covilhã would be developed.

The developed application aims to present a tourist route in Covilhã, which gives information through Augmented Reality, in the various waypoints. This application also makes use of geolocation systems (GPS) that guides users through the various points.

Augmented Reality (AR) is like a supplement to the real world using virtual objects created with computer graphics. This technology is expanding, its applications are varied and their diversity has been increasing, being used in several areas. In this dissertation there are examples of applications in areas such as advertising and marketing, medical, entertainment, military, education and finally tourism.

The application described in this paper was developed for Android using the *Apache Cordova*, *Wikitude* and *Google Maps*.

Android is an operating system (OS) for mobile devices based on Linux. It is the OS for mobile devices with more users in the world. Being open source it facilitates the work of developers in the application development process as well as its publication in the *Play Store* (Android application store).

Being this application a browser application, *Apache Cordova* was needed to its execution. This framework facilitates the creation of applications for mobile devices, since these will be developed using *HTML*, *CSS* and *JavaScript*, without having to code in *Java*.

To create the functionalities of the application it was necessary *Wikitude*, a plugin to perform the AR, as well as *Phonemap Google Maps plugin* for maps and directions, presented in the application.

## Keywords

Augmented Reality, Android, Application, Apache Cordova, Wikitude

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

# Contents

<b>Dedicatory</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Listings</b>	<b>xv</b>
<b>List of Acronyms</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Focus and Scope . . . . .	1
1.2 Problem Definition and Objectives . . . . .	1
1.3 Organization of the Dissertation . . . . .	2
<b>2 Android</b>	<b>3</b>
2.1 Introduction . . . . .	3
2.2 Android History . . . . .	3
2.3 The Android Platform . . . . .	4
2.3.1 The Android Architecture . . . . .	4
2.4 The Android Development Environment . . . . .	6
2.4.1 Getting Started . . . . .	6
<b>3 Apache Cordova</b>	<b>9</b>
3.1 Introduction . . . . .	9
3.2 The Command-Line Interface . . . . .	9
3.2.1 Cordova CLI Installation . . . . .	9
3.3 Setup and Configuration . . . . .	10
3.3.1 Create an empty new app . . . . .	10
3.3.2 Add Platforms . . . . .	10
3.3.3 Build and Run the App . . . . .	11
3.3.4 Add Plugins . . . . .	11
<b>4 Augmented Reality</b>	<b>15</b>
4.1 Introduction . . . . .	15
4.2 The history of Augmented Reality . . . . .	17
4.3 Related Work . . . . .	17
4.3.1 Advertising and Marketing . . . . .	17
4.3.2 Medical . . . . .	18
4.3.3 Entertainment . . . . .	19
4.3.4 Military . . . . .	20
4.3.5 Education . . . . .	20
4.3.6 Tourism . . . . .	21

<b>5</b>	<b>Development</b>	<b>23</b>
5.1	Why Wikitude? . . . . .	23
5.2	System Architecture . . . . .	23
5.3	System Implementation and Design . . . . .	24
5.3.1	Initial Menu . . . . .	25
5.3.2	Paths . . . . .	26
5.3.3	Image Recognition . . . . .	28
5.3.4	Near You . . . . .	33
5.3.5	Directions . . . . .	39
<b>6</b>	<b>Conclusions and Future Work</b>	<b>41</b>
	<b>Bibliography</b>	<b>43</b>

## List of Figures

2.1	The first Android phone, HTC Dream/T-Mobile G1 [Tay15]	3
2.2	Android Architecture [Joh15]	4
2.3	Android SDK Manager	7
3.1	Display the SHA-1 fingerprint	12
3.2	Google Maps API Key for Android	12
4.1	Simplified representation of a RV Continuum [MTUK94]	15
4.2	Sensor-Based Tracking [Bla12]	16
4.3	How markers work [Gra14]	16
4.4	Webcam Social Shopper - Virtual Dressing Room [Web13]	18
4.5	2D augmented reality image overlay device to guide needle insertion procedures [Que]	18
4.6	<i>MonstAR Truck</i> game using AR. [Out14]	19
4.7	Game <i>WonderBook: Book of Spells</i> . [Son15]	19
4.8	Example of the information that smart helmets can add to the soldier's field of vision [Pri14]	20
4.9	<i>Elements 4D</i> by Daqri [Ara14]	20
4.10	<i>Anatomy 4D</i> by Daqri [Kim14]	21
4.11	Example of POI Geolocation on a map [ARS]	21
4.12	Museum of London <i>StreetMuseum</i> app [Jaw14]	22
5.1	System Architecture	24
5.2	Discover Covilhã icon	24
5.3	Discover Covilhã initial menu	24
5.4	Points of the different paths and how to change them	26
5.5	Page with the pictures of the places with Image Recognition	28
5.6	Image Recognition with information in AR	29
5.7	Image Recognition with video content in AR	32
5.8	Image Recognition with animated 3D object in AR	33
5.9	The radius menu, the POIs and the information shown when they are clicked	34
5.10	The directions to the desired place with Maps	39



## List of Listings

3.1	Command to install Cordova on Windows . . . . .	9
3.2	Command to install Cordova on OS X and Linux . . . . .	9
3.3	Command to create a <i>Cordova</i> application . . . . .	10
3.4	Command to add a platform, label=lst:plat . . . . .	10
3.5	Command to add the platform Android . . . . .	10
3.6	Command to remove a platform . . . . .	11
3.7	Command to list the added platforms . . . . .	11
3.8	Command to build the app . . . . .	11
3.9	Command to build the Android app . . . . .	11
3.10	Command to run the Android app . . . . .	11
3.11	Command to add plugins . . . . .	11
3.12	Command to get the SHA-1 fingerprint . . . . .	12
3.13	Command to add the Google Maps plugin . . . . .	12
3.14	Command to add the Wikitude plugin . . . . .	13
5.1	HTML code for initial menu . . . . .	25
5.2	HTML code to create map canvas . . . . .	26
5.3	Definition of POIs . . . . .	26
5.4	Create Map . . . . .	27
5.5	Target Collection load . . . . .	29
5.6	Creating HTML information page and button . . . . .	29
5.7	Creating video content and respective play button . . . . .	30
5.8	Adding 3D object and animation . . . . .	32
5.9	Places definition . . . . .	34
5.10	Loading the POIs from the local data . . . . .	34
5.11	<i>loadPoisFrom.JsonDataFn</i> function . . . . .	35
5.12	Radar div definition . . . . .	36
5.13	<i>Near You</i> header . . . . .	36
5.14	<i>showRangeFn</i> function . . . . .	36
5.15	Panel with range information . . . . .	37
5.16	Initialize the range in 100% . . . . .	37
5.17	Function <i>onMarkerSelectedFn()</i> . . . . .	38
5.18	Panel with POI information . . . . .	38
5.19	Functions <i>goToMaps()</i> and <i>selectd()</i> . . . . .	40



## List of Acronyms

ADT	Android Development Tools
AOT	Ahead-of-time
API	Application Programming Interface
ATR	Android Runtime
App	Application
AR	Augmented Reality
CGI	Computer Generated Imagery
CLI	Command-line Interface
CSS	Cascading Style Sheets
GPS	Global Positioning System
HMS	Helmet-Mounted Sights
HTML	HyperText Markup Language
HUD	Head-Up Displays
MAR	Mobile Augmented Reality
MARA	Mobile Augmented Reality for Advertising
OS	Operating System
SDK	Software Development Kit



# Chapter 1

## Introduction

### 1.1 Focus and Scope

The focus of this project is the MARA that combines the AR with mobile devices, in this case Android ones, and advertisement, being this advertisement applied to a product/service or tourism/culture.

AR is commonly confused with Virtual Reality even though they are completely different, being even on opposite sides on the Milgram's Reality-Virtuality Continuum. While briefly augmented reality is the addition of CGI elements into the real world through a device, virtual reality is the addition of the real world into the virtual world through, for instance, HUD.

Being Android open source and the most used OS to mobile devices, it was the right choice to use it to create this application.

Since it was decided to create a touristic application to Covilhã it was intended to use Augmented Reality and the geolocation systems, not only to represent the different paths that already exist around the city but also to interactively show to the tourists the main points of interest.

Augmented Reality provides contents that can not usually be found in the POIs information given at the location, such as videos or even 3D objects and animations, which the tourist would only be able to visualize in the internet and in multimedia presentations of the place.

### 1.2 Problem Definition and Objectives

The main objectives for this dissertation were:

- To investigate and learn how to create an Android MAR application
- To development of a prototype of an application using AR

In order to accomplish these objectives some tasks were needed to be executed, such as the elaboration of a state of art about Augmented Reality, especially applied to mobile devices, selection of the right environment to the development of the application and the definition and implementation of it, lastly it was needed to test the app and verify if everything was right and the image recognition was working properly.

The state of art, or the related work (Section 4.3), was the first thing to do. There was a large amount of AR applications so it was choose to reference the one that were more relevant on different areas.

*Vuforia*, *Metaio* and *Wikitude* were the environments considered to develop the Augmented Reality part of the application. In the end was the *Wikitude* the chosen plugin since it was the most suitable one.

About the app definition it was decided to create a MARA app about Covilhã points of interest (POIs) intended for the tourism of the city, since it did not exist and could be a good improvement to the city tourism.

The tests of the application were made on different devices but the opportunity to test it with people from different ages was not possible.

### 1.3 Organization of the Dissertation

This dissertation is divided into four main chapters preceded and succeeded by the Introduction and Conclusions and Future Work, respectively. After chapter 6 the bibliographical references used on this dissertation can be found. The contents of each chapter of this dissertation can be summarized as follows:

*Chapter 1* is an introduction to the project, explaining the context of the dissertation. It also includes the identification of the main objectives and the problem to be solved.

*Chapter 2* gives an overview about Android and its history. It also explains how to get started with the Android Development Environment.

*Chapter 3* provides information about Apache Cordova and its Command-Line Interface. This chapter teaches how to setup and configure Apache Cordova properly and how to create, build and run a new app.

*Chapter 4* Augmented Reality and its history are presented in this chapter. Afterwards it is shown some of the related work on different areas, such as Advertising and Marketing, Medical, Entertainment, Military, Education and finally Tourism.

*Chapter 5* this chapter is the main core of the dissertation as it presents the app and how it was developed. The reasons why some decisions were taken are also explained in this chapter.

*Chapter 6* the final chapter wraps up the conclusions of this dissertation and the directions for future work.

# Chapter 2

## Android

### 2.1 Introduction

Android powers hundreds of millions of mobile devices around the world, being the largest installed base of any mobile platform. It is increasing exponentially with about a million new users per day who start looking for applications as soon as they turn their devices on. [ANDb]

### 2.2 Android History

The Android is an open source OS initially developed by Android, inc. that was bought by Google in 2005. Founded in 2007 by a group of IT companies, like Google, Intel, Samsung, Acer, Motorola, Asus, DoCoMo, HTC, Huawei, Sprint, Kyocera, T-Mobile, LG, Vodafone, SonyEricsson, Qualcomm and NVidia, Open Handset Alliance (OHA) is nowadays its owner, being the company is responsible for the development and the growing of Android [OB14].

The first device that came into the market with the Android OS was the HTC Dream/T-Mobile G1 (Figure 2.1), in 2009 and was available specifically for the Australian market[Tay15]. After that several devices were released using Android, including not only phones but also a netbook and cars with smart panels and automotive navigation system [OB14]. Recently it is also possible to find the Android OS in watches, televisions and tablets.



Figure 2.1: The first Android phone, HTC Dream/T-Mobile G1 [Tay15]

Android allows developers to access all its native apps such as contacts, calendar, Google Maps, and many others. Developers can also access to the hardware of the devices [OB14]. Furthermore, as it is an open source system, the process of creating apps and publishing them is a lot easier than other platforms.

## 2.3 The Android Platform

Android is developed by Google Inc. and the Open Handset Alliance. It was based on the operating system Linux, therefore incorporating characteristics from this system, such as OS Kernel, archive manager, etc [Rab07].

The Android Platform consists in:

- A software stack for mobile devices
- Android SDK for creating apps.

### 2.3.1 The Android Architecture

The android is composed of five different layers of software (Figure 2.2)each one providing different services to the layer immediately above it [Joh15]. These layers are:

- Applications
- Application Framework
- Libraries and Android Runtime
- Linux Kernel

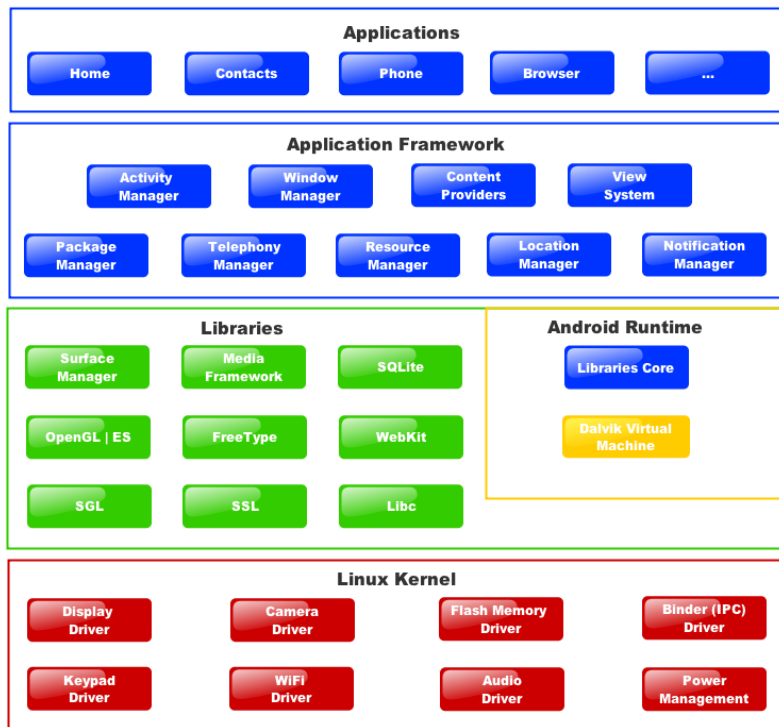


Figure 2.2: Android Architecture [Joh15]

### 2.3.1.1 Applications

It consists on standard apps which include, for instance [Rab07]:

- Home - Main Screen
- Browser - View web pages
- Contact - Contacts database
- Phone - Dial phone numbers
- Email Reader - Compose and read email messages

And other applications that will be developed, being all of them coded in Java.

### 2.3.1.2 Application Framework

At this level of the architecture one can find some components that will allow new structures to be used in future applications[Rab07].

These components are [Rab07]:

- Package Manager - Keeps track of app packages on device
- Window Manager - Manages the windows comprising an app
- View System - Provides common user interface elements, such as icons, text entry boxes, buttons, etc
- Resource Manager - Manages non-compiled resources, such as strings, graphics and layout files
- Activity Manager - Manages the application lifecycle and navigation stack
- Content Provider - Inter-application data sharing
- Location Manager - Provides location and movement information
- Notification Manager - Place notification icons in the status bar when important events occur

### 2.3.1.3 Libraries and Android Runtime

This layer is divided in libraries and android runtime. The libraries are coded in C/C++ and they are the following [Rab07]:

- System C Library - implementation of C standard library (libc) to systems that support the Linux platform
- Surface Manager - display management
- Media Framework - libraries to audio, video and image visualization

- WebKit - Browser engine
- OpenGL - Graphic engine
- SQLite - Relational database engine

Before Android 5.0.x (Lollipop or L) the apps were executed by the Dalvik Virtual Machine. In the most recent versions of Android the Dalvik is no longer available and it was substituted by ART [ART, Dal].

ART is the managed runtime used by applications and some system services on Android [Dal]. In ART, at the installation process, the dex code is compiled to native OAT files [ART].

The ART has various features such as AOT compilation, improved garbage collection and development and debugging improvements (support for sampling profiler and for more debugging features and improved diagnostic detail in exceptions and crash reports) [Dal].

#### 2.3.1.4 Linux Kernel

The android has a linux kernel, version 2.6 and 3.0 (API level 11 and posteriors). This layer is responsible for the power management, android shared memory, low memory killer, interprocess communication and many other tasks. It is the Android's responsibility to perform the communication between the developed software and the device hardware[OB14].

## 2.4 The Android Development Environment

The android development environment is the workbench for writing Android applications.

### 2.4.1 Getting Started

First of all, one needs to download and install the ADT Bundle, which can be downloaded from <http://developer.android.com/sdk>. The ADT Bundle contains the latest android platform, the Eclipse and the ADT plugin, the latest system image for emulator and additional development tools.

Since on this project it was necessary to compile and run the android project from the Cordova CLI (Section 3.2), there was the need to add the Android path and Java path to the environment variables.

#### 2.4.1.1 Android Emulator

Even though the Android emulator, which runs virtual devices, does not require an actual phone, and possesses a reconfigurable hardware and non-destructive changes, it can be extremely slow

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

and some features are unavailable, like the bluetooth and USB connections. Moreover, the performance/user experience can be misleading. This is the reason why on this project the application was run on an actual android device.

### 2.4.1.2 Adding SDK Packages

To add SDK packages one needs to execute the *Android SDK Manager* (Figure 2.3), which can be found at the root of the Android SDK directory on Windows and on Mac/Linux by opening a terminal and navigate to the *tools* directory in the location where the Android SDK was installed, then execute `android sdk` [Anda].

To get the latest SDK tools select the *Android SDK Tools*, *Android SDK Platform-tools* and *Android SDK Build-tools* (latest version) from the directory *Tools* and the *SDK platform* and a system image for the emulator, such as *ARM EABI v7a System Image*, from the first *Android X.X* folder (Figure 2.3) [Anda].

In the *Extras* directory one can select the *Android Support Repository* and *Android Support Library* to get the support library for additional APIs and/or select *Google Repository* and *Google Play Services* to get Google Play services for even more APIs [Anda].

After selecting everything that one desires it is just necessary to click *Install X packages*, accept the license agreement for each one of them and finally click *Install* [Anda].

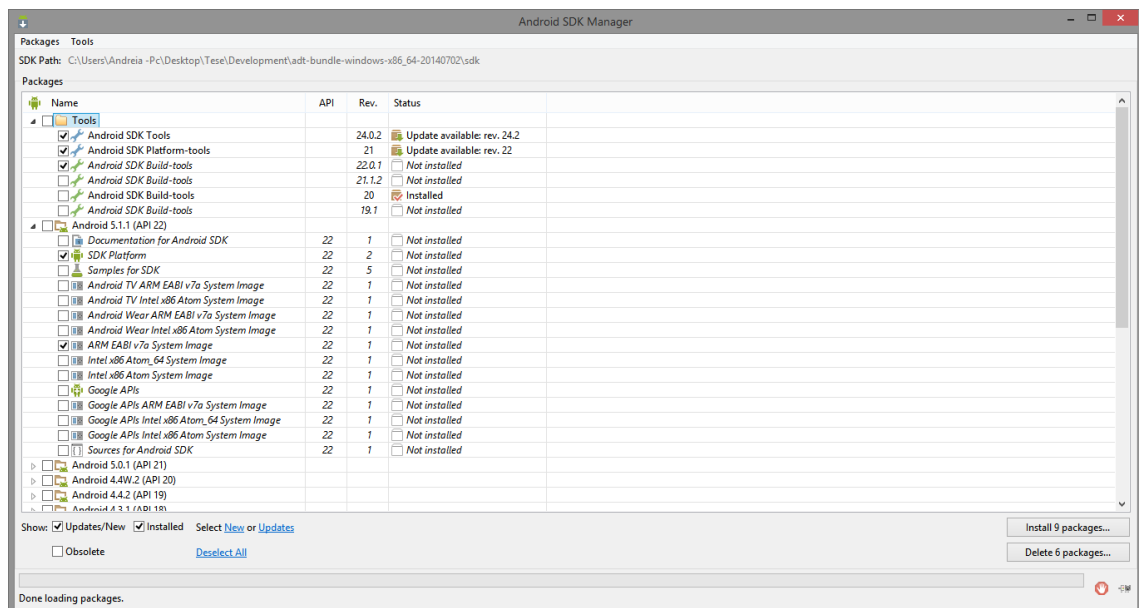


Figure 2.3: Android SDK Manager



# Chapter 3

## Apache Cordova

### 3.1 Introduction

*Apache Cordova* is a set of device APIs that allow the access of native device functions such as the camera or accelerometer from JavaScript. The UI frameworks that *Cordova* combines, like *jQuery Mobile*, allows a smartphone app to be developed with just *HTML*, *CSS*, and *JavaScript* without any native code (*Java*, *Objective-C*, etc) [The13a].

The *JavaScript* APIs are consistent across multiple device platforms and built on web standards which makes the app portable to other device platforms with minimal to no changes [The13a].

*Cordova* is available for iOS, Android, Blackberry, Windows Phone, Palm WebOS, Bada, and Symbian. In this project *Cordova* was used with the Android platform [The13a].

### 3.2 The Command-Line Interface

#### 3.2.1 Cordova CLI Installation

- Installation of *Node.js*, if not installed already, in order to be possible to invoke *node* and *npm* on the command line. May be necessary to add the *npm* directory to the *PATH*, in the environment variables. Usually, on Windows, *npm* can be found at `C:\Users\username\AppData\Roaming\npm`, on OS X and Linux it can usually be found at `/usr/local/share/npm` [The13b].
- Installation of a *git client*, if not already installed, so it is possible to invoke *git* on the command line
- Finally it is possible to install Cordova using *npm* utility of *Node.js*. The `-g` flag on the Cordova installation tells *npm* to install Cordova globally [The13b].

On Windows:

```
npm install -g cordova
```

Listing 3.1: Command to install Cordova on Windows

On OS X and Linux:

```
sudo npm install -g \textit{Cordova}
```

Listing 3.2: Command to install Cordova on OS X and Linux

### 3.3 Setup and Configuration

#### 3.3.1 Create an empty new app

To create an application the user just needs to go into a directory, where the application is supposed to be created and run the listing 3.3 [The13b].

```
cordova create hello com.example.hello HelloWorld
```

Listing 3.3: Command to create a *Cordova* application

If the user wants to get information about the process of the installation, he just needs to insert the flag `-d` [The13b].

The first *hello* determines the name of the directory to be generated for the project. This directory should not exist [The13b].

The second argument *com.example.hello* provides a reverse domain-style identifier. This argument is optional, only if the last argument is also omitted [The13b].

The last argument *HelloWorld* provides the display name of the application. This argument is optional [The13b].

#### 3.3.2 Add Platforms

In order to add platforms the user needs to be within the project directory, or any subdirectories within its scope. Then one just needs to run the listing ?? [The13b].

```
cordova platform add PlatformName
```

Listing 3.4: Command to add a platform, label=lst:plat

The *PlatformName* must be replaced with the platform the user wants to add to the project. The ability to use some platforms is related to whether the machine supports each SDK and whether the SDK is already installed [The13b].

In a Mac the names of the platforms supported are: *ios*, *amazon-fireos*, *android*, *blackberry10* and *firefoxos*. In a Windows machine the names of the platforms supported are *wp8* (relative to the different versions of the windows phone), *windows*, *amazon-fireos*, *android*, *blackberry10* and *firefoxos* [The13b].

For instance, in this project the platform used was the android platform, in which the SDK was previously installed. To add this platform the command is the following.

```
cordova platform add android
```

Listing 3.5: Command to add the platform Android

If the user wants to remove a platform (e.g. android) the command [The13b] to run is the listing 3.6.

```
cordova platform remove android
cordova platform rm android
```

Listing 3.6: Command to remove a platform

To list all the current set of platforms added the command is:

```
cordova platforms ls
```

Listing 3.7: Command to list the added platforms

### 3.3.3 Build and Run the App

To build the application the command [The13b] is:

```
cordova build
```

Listing 3.8: Command to build the app

Or it can be limited to a certain platform by adding the name of the platform to the end of the command [The13b], like the example below.

```
cordova build android
```

Listing 3.9: Command to build the Android app

To run the application the command is the one on listing 3.10. [The13b]

```
cordova run android
```

Listing 3.10: Command to run the Android app

This command will run and install the application on the device connected to the machine or in an emulator. To specifically run the app on an emulator it is just necessary to substitute the word run for emulate on the command above.

### 3.3.4 Add Plugins

The command [The13b] to add plugins to the project is the following.

```
cordova plugin add PluginName
```

Listing 3.11: Command to add plugins

In this project the plugins used were the *Wikitude Plugin to Cordova* and *Phonegap googlemaps plugin* that supports both *Phonegap* and *Apache Cordova*.

### 3.3.4.1 Phonegap Googlemaps Plugin

First of all it is necessary to display the debug certificate. The *keytool* can be found inside the bin folder of the *Java* installation directory, which can usually be found in `C:\ProgramFiles\Java\jdk1.8.0_31\bin`. The path will change according to the *jdk* version.

Inside this folder the following command [Kat15] should be run.

```
keytool -list -v -keystore "C:/Documents and
Settings/username/.android
/debug.keystore" -alias androiddebugkey -storepass android -keypass
android
```

Listing 3.12: Command to get the SHA-1 fingerprint

The result will be the one shown in figure 3.1, and the *SHA-1 fingerprint* must be saved.

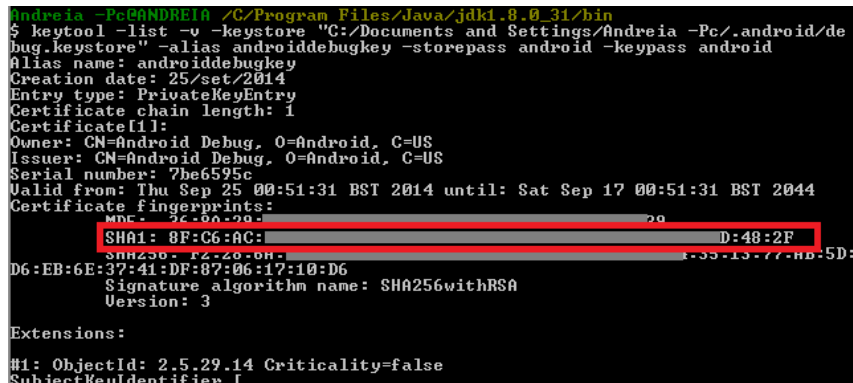


Figure 3.1: Display the SHA-1 fingerprint

The next step is to obtain the *Google Maps API Key for Android*, which can be done by going to the *Google APIs Console* and then turn on *Google Maps Android API v2*. After turning on the API the user should go to API Access page and click on *Create New Key* button and then on *Android Key*, in the resulting dialog, the user should enter the *SHA-1 fingerprint*, previously saved, a semicolon and then the application's package name (Figure 3.2) [Kat15].



Figure 3.2: Google Maps API Key for Android

Finally to add the plugin, the user must run the command below [Kat15] and substitute the *KEY* with the *API Key* previously generated.

```
cordova plugin add plugin.google.maps --variable
API_KEY_FOR_ANDROID="KEY"
```

Listing 3.13: Command to add the Google Maps plugin

### 3.3.4.2 Wikitude Plugin

The following command add the *Wikitude plugin* to the *Cordova* project [Wik15b].

```
cordova plugin add  
https://github.com/Wikitude/wikitude-cordova-plugin.git
```

Listing 3.14: Command to add the Wikitude plugin

This process takes several minutes and when it is done the command quits with a success message, *Plugin "com.wikitude.phonegap.WikitudePlugin" fetched* [Wik15b].

After the plugin is added the user must build the app [Wik15b].



# Chapter 4

## Augmented Reality

### 4.1 Introduction

Augmented Reality is a component of mixed reality, according to the Milgram's Reality-Virtuality Continuum (Figure 4.1). The RV Continuum puts Augmented Reality and Virtual Reality in opposite ends of the continuum. On one side, one has the real environment, and on the other side there is the virtual environment. Closer to the real environment side lays the augmented reality [MTUK94].

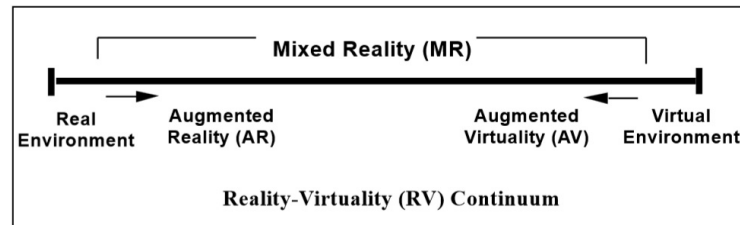


Figure 4.1: Simplified representation of a RV Continuum [MTUK94]

AR is a kind of supplement to the real world using virtual objects created by CGI. It raises the interaction between humans and technology, making the real world more complex, and the virtual world more realistic [KK14].

The Augmented Reality: [vKP10]

- Combines real and virtual images
- Is interactive and in real-time
- Aligns real and virtual objects in 3D

The combination of the real and virtual images is done with several different display technologies such as, head attached (head mounted display/projector), body attached (handheld display/projector, e.g. mobile phones) and spatial (spatially aligned projector/monitor) [BCL15].

Registration (positioning virtual objects in regarding to the real world) and Tracking (continually locating the users view point) are two components of objects aligned in 3D[BCL15].

Tracking techniques can be divided into: [ZDB08]

- Sensor-Based Tracking Techniques
- Vision-Based Tracking Techniques

- Hybrid Tracking Techniques

Sensor-Based Tracking Techniques (Figure 4.2) can be either active, such as mechanical, magnetic and ultrasonic sensors (GPS, Wifi and cell location) or passive, such as inertial sensors (compass, accelerometer and gyroscope) [BCL15].

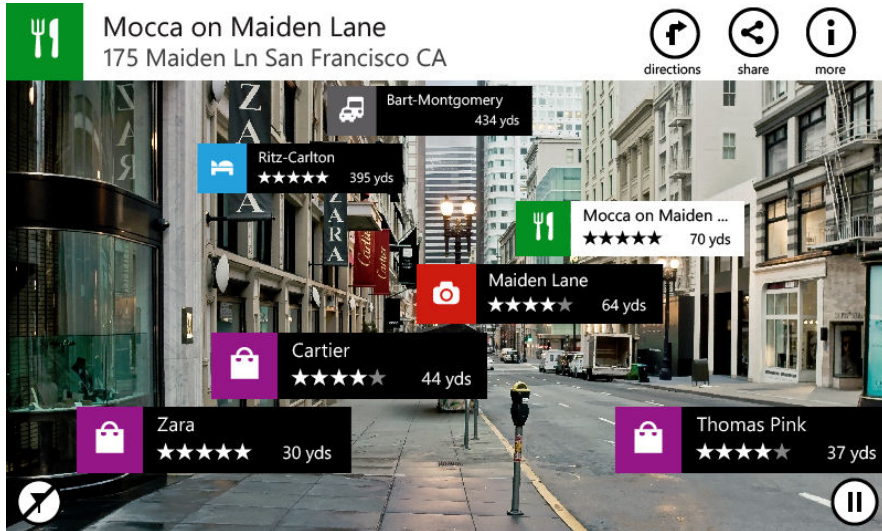


Figure 4.2: Sensor-Based Tracking [Bla12]

Vision-Based Tracking Techniques can be divided in specialized tracking, marker-based tracking and markerless tracking.

Marker-based tracking (Figure 4.3) uses a technique to track the markers already agreed. This system is used in a space with a specific purpose. The target markers are recognized by the device via the camera and are defined by the developer [KK14].

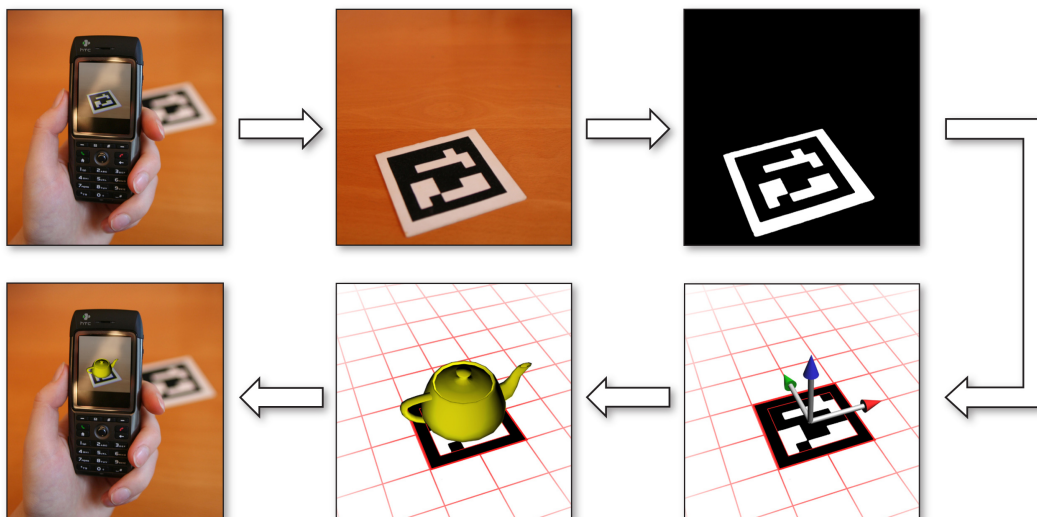


Figure 4.3: How markers work [Gra14]

The markerless augmented reality system uses the real world to compensate for the weak points of marker augmented reality system.[KK14] It is the most difficult technique in this field and it can be divided mainly in feature-based and model-based [PM06].

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

Feature-based methods focuses on finding a correspondence between 2D image features, such as geometrical primitives (points, segments, circles, etc), object contours, regions of interest and other, and their 3D world frame coordinates [PM06, ZDB08].

Model-based trackers usually construct their models based on lines or edges in the model[ZDB08].

The hybrid tracking uses the combination of sensors (e.g. computer vision and inertial sensors)[BCL15].

## 4.2 The history of Augmented Reality

The first AR interface, Head Mounted Display, was developed in the 1960's by Ivan Sutherland [Rob13].However, AR did not become popular until October 1998, when the first International Workshop on Augmented Reality '98 (IWAR 98) conference was held,in San Francisco, which lead to an increase of AR developers and researchers [ZDB08, Rob13].

Due to this increase of popularity some other augmented reality conferences were born, such as the International Symposium on Mixed and Augmented Reality (ISMAR) and The International Symposium on Augmented Reality (ISAR) [Rob13].

AR is becoming increasingly popular over the years, and so are the applications of this technology, especially for smart phones.

## 4.3 Related Work

Augmented Reality does the merge of the virtual and real world through the interaction with sensorial channels, such as vision, sound and touch. This means that AR is being used increasingly more through the years and applicable diversely. Between many others, there are AR solutions for:

### 4.3.1 Advertising and Marketing

There are different applications of AR in this field like the creation of AR solutions that can be manipulated by the user in order to have a more realistic vision, in 3D, 3D animations or just in 2D but framed in the real world, of a determined product or object.

One example of this application is full-size AR virtual cars in shopping centers, so that users can see the whole car and interact with it virtually.

It is also possible to see a furniture object in one's own environment so they can have a better impression of how that item is going to look there or which color looks better.

Some implementations are also making possible to virtually try clothes using virtual mirrors or even their webcams in virtual dressing rooms (Figure 4.4), making this process a lot easier, more fun and quicker.

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)



Figure 4.4: Webcam Social Shopper - Virtual Dressing Room [Web13]

Probably the biggest and most practical usage of AR in Advertising and Marketing is through smartphones. Users can access to more information about a product or propaganda over applications on their smartphones (e.g. Blippar).

### 4.3.2 Medical

One of the medical uses of augmented reality is in the field of minimally-invasive surgery (Figure 4.5), which reduces the trauma of an operation by using small incisions or no incisions at all [Azu97].

Augmented Reality can provide the patient's anatomy, making this type of surgery possible without larger incisions [Azu97].

The combination of computer graphics and the real world view can be achieved by using 3D datasets from the patient. This datasets can be collected in real time using sensors like Magnetic Resonance Imaging (MRI), Computed Tomography scans (CT), or ultrasound imaging [Azu97].

AR is also used to help improve clinical and surgical practices, which results in a lowering of the costs and an increasement in safety of procedures [Rob13].

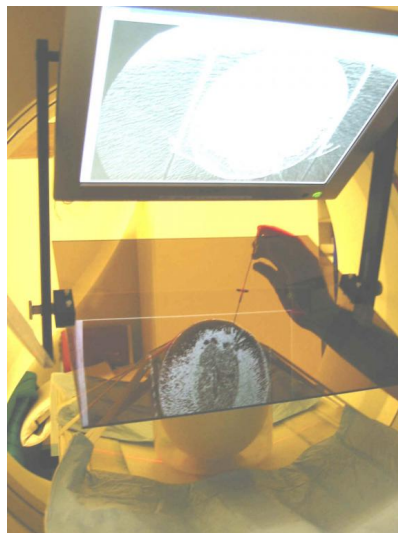


Figure 4.5: 2D augmented reality image overlay device to guide needle insertion procedures [Que]

This technology can also be used to perform needle biopsy (e.g. of breast tumors), being less invasive [Azu97].

### 4.3.3 Entertainment

In this field, AR is noticeable mainly in games. Users can have a more realistic gaming experience with this technology, having the possibility to bring the game into the real world.

There are several examples of AR in games, like the one on the image below, *MonstAR Truck* by Outplay Entertainment, it is a game for mobile devices and turns the real world (e.g. user's home) into a racetrack, like it is visible on figure 4.6 [Out14].



Figure 4.6: *MonstAR Truck* game using AR. [Out14]

Other games using AR can also be played in the most recent video games consoles (e.g. Playstation 3, Xbox 360, etc). On figure 4.7 there is an example of a game for Playstation 3, *WonderBook: Book of Spells* by SCE London Studio [Son15].



Figure 4.7: Game *WonderBook: Book of Spells*. [Son15]

#### 4.3.4 Military

One recent development in the military field was the smart helmet (Figure 4.8), an invention quite similar to Google Glass which has the capacity to soldiers on the battlefield. It attaches to a military helmet, gives live updates, overlays maps into the soldier's field of vision and it can also be integrated with weapons control system [Pri14].



Figure 4.8: Example of the information that smart helmets can add to the soldier's field of vision [Pri14]

Augmented reality is also used to superimpose vector graphics upon the military aircraft and helicopters pilot's view of the real world using HUDs and HMS. This provides to the pilot basic navigation, information about the flight and targets in the environment [Azu97].

#### 4.3.5 Education

There is a vast amount of educational applications for mobile devices that can provide a more interactive way of learning. These act not only as a motivational factor to students, who become more interested in learning, but also as a support during the learning experience [Nes13].

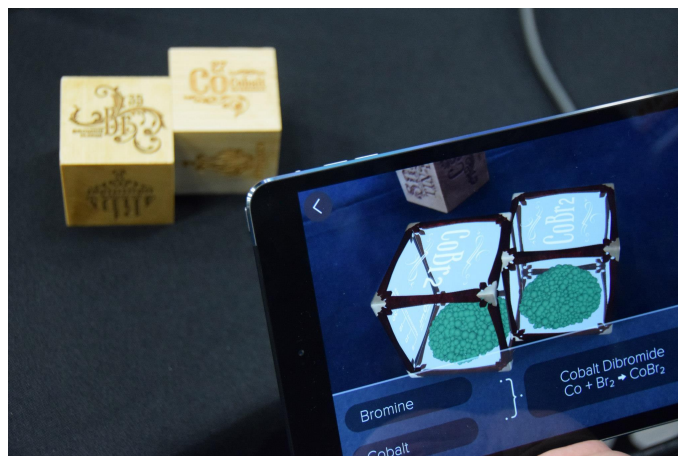


Figure 4.9: *Elements 4D* by Daqri [Ara14]

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

Applications such as *Elements 4D* Daqri 4.9 allow students to learn about chemical elements and how these interact. Using blocks, it shows students what happens when the elements are combined, and how the chemical reaction occurs. *Anatomy 4D* also by Daqri 4.10 allows users to explore the human body [Nes13].



Figure 4.10: *Anatomy 4D* by Daqri [Kim14]

Other types of applications allow students to scan their homework, leading to a video of the teacher helping the students on solving the problem [Nes13].

### 4.3.6 Tourism

In this field are several applications for smart phones, tablets and other types of devices, using the sensors available on the system it is possible to create complete guides of cities and show some POIs from that place (Figure 4.11).

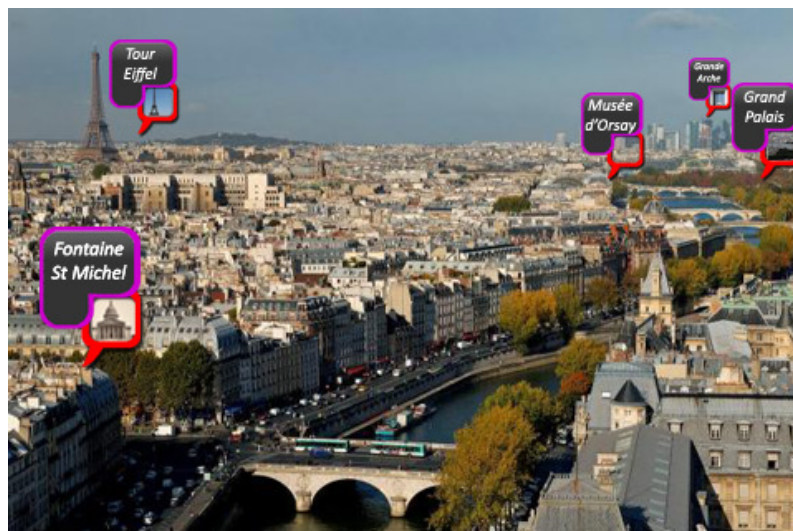


Figure 4.11: Example of POI Geolocation on a map [ARS]

There are also examples of AR apps that show to the user images of the history of some monuments or places, like is shown in Figure 4.12 [Jaw14].



Figure 4.12: Museum of London *StreetMuseum* app [Jaw14]

# Chapter 5

## Development

This project is a browser app that was developed using Apache Cordova (Chapter 3), Wikitude Plugin (Section 3.3.4.2) and Phonegap Google Maps Plugin (Section 3.3.4.1).

### 5.1 Why Wikitude?

It was decided to use *wikitude* since it was the plugin that:

- Could be used completely with a free license, it just shows a *Trial* watermark
- Had more information and documentation
- Was able to use the GPS, accelerometer and gyroscope
- Had the best results in the image recognition of buildings

Another option was *Vuforia* however it was excluded not only because to use the GPS another plugin would be necessary, but also because the image recognition is centered on pattern recognition which did not correspond to what was needed.

*Metaio* was also an option that was tested but the lack of information and documentation about the plugin made the experience quite difficult to concretize.

### 5.2 System Architecture

The system has an initial menu (Section 5.3.1) that is divided in the following options: *Paths*, *Image Recognition*, *Near You* and *Directions* (Figure 5.1). These options were divided because they used different plugins and functionalities of those plugins.

In the *Paths* option (Section 5.3.2) the user can choose a path and see the location of the points of the path in a map, as well as its own location, this being possible by using the *Phonegap Google Maps Plugin*. The *Image Recognition* option (Section 5.3.3) allows the user to get information, in AR, about some POIs using the image of that place, *Wikitude* is the plugin responsible for this possibility. *Near You* (Section 5.3.4) shows to the user, using AR, the POIs near him, also allowing the user to get further information about that place, such as the distance to it, by clicking in the POI. Finally the *Directions* option (Section 5.3.5) asks the user which place he would like to visit and using *Maps* application, that it is native on Android devices, shows the path to that place from the user current location.

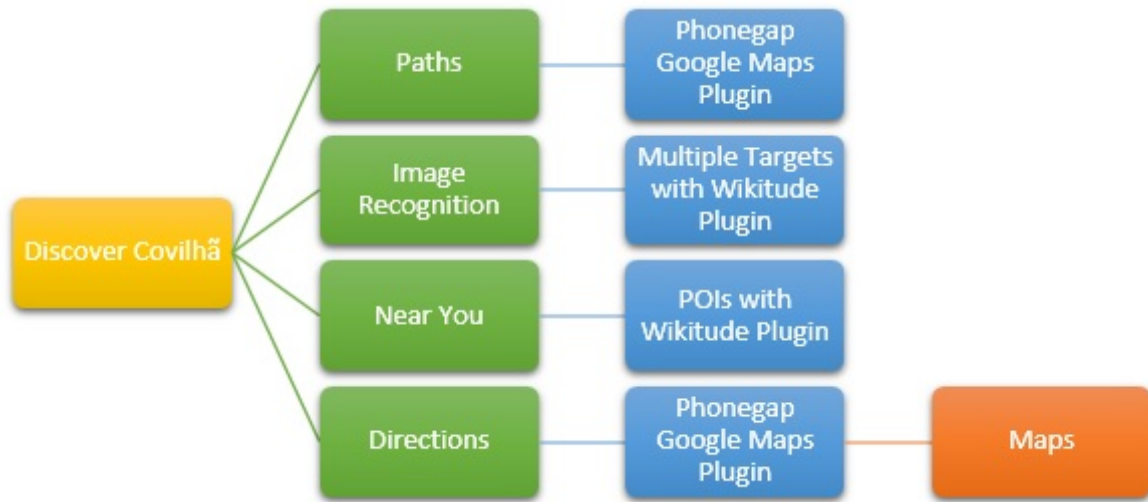


Figure 5.1: System Architecture

### 5.3 System Implementation and Design

Being a browser app it is not necessary to code in *Java* but with *HTML*, *CSS* and *JavaScript*. The visual design of the app is quite simple and minimalist, making the app more intuitive. Black, white and grey are the most used colors, as they allow everything to be easily seen and prevents the app from being tiresome for the eyes. The icon created for this app is the Covilhã logo with a magnifying glass, figure 5.2.



Figure 5.2: Discover Covilhã icon

This version of the application is in portuguese.



Figure 5.3: Discover Covilhã initial menu

## 5.3.1 Initial Menu

The initial menu of the app, figure 5.3, was created using *jquery-mobile* and it is in the same file as other pages, such as *Paths* and *Directions*.

To create the menu the code used was:

```

<!-- New Page -->
<div data-role="page" id="page1" style="background: none;" >

<!-- Header with app name -->
  <div data-role="header" data-theme="b">
    <h3 class="ui-title" role="heading">
      Discover <br> Covilhã</h3>
    </div>

<!-- Navigation Bar (Menu Options) -->
  <div data-role="navbar" data-theme="b" data-grid="a" >
    <ul>
      <li><a href="#page2" data-ajax="false">
        <br>
        Percursos <br> <br>
      </a>
      </li>
      <li><a href="javascript:app.loadARchitectWorld
        (getSamplePath(0, 0));" >
        
        <br>Reconhecimento<br> de imagens
      </a>
      </li>
      <li><a href="javascript:app.loadARchitectWorld
        (getSamplePath(0, 1));" >
        <br>
        Perto de si <br> <br>
      </a>
      </li>
      <li><a href="#page3" data-ajax="false">
        <br>
        Como chegar <br> <br>
      </a>
      </li>
    </ul>
  </div>
</div>

```

Listing 5.1: HTML code for initial menu

First of all it is created a new page with a header and navigation bar. In the header there is the

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

name of the app, *Discover Covilhã*, and in the navigation bar there are the icons and names of the menu options, such as *Paths*, *Image Recognition*, *Near you* and *Directions*.

### 5.3.2 Paths

This option is coded on the same html file as the initial menu. The page has a navigation bar with the name of the option and a button to open the side bar where the user can choose the intended path.

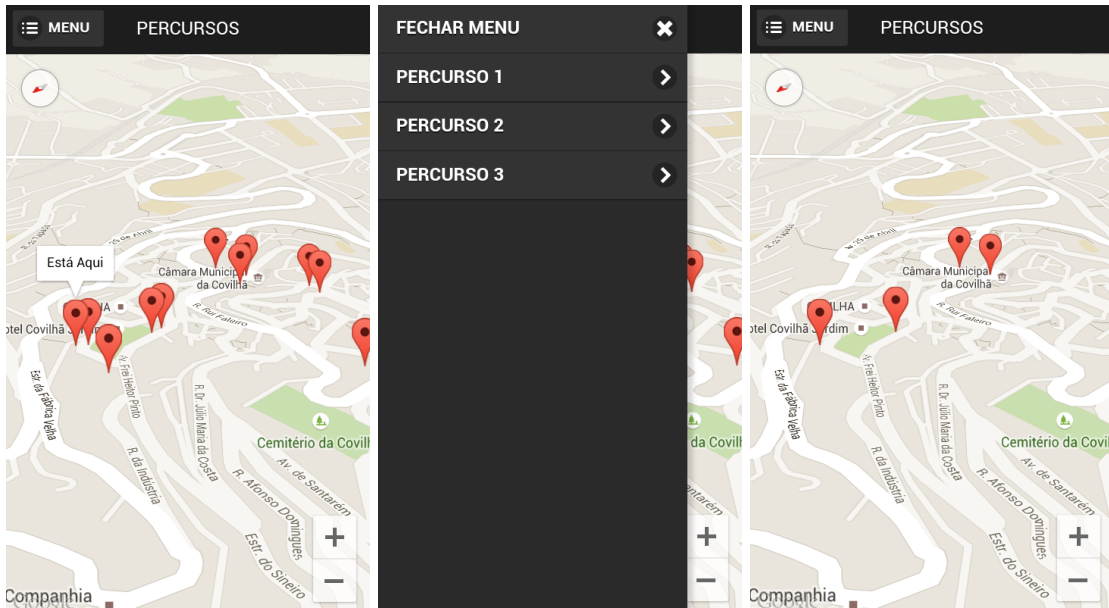


Figure 5.4: Points of the different paths and how to change them

The html code is similar to the one used in the initial menu, however it is necessary to create a canvas to the map.

```
<div style="width:100%;height:500px" id="map_canvas">
</div>
```

Listing 5.2: HTML code to create map canvas

After the canvas is created in the html code it is necessary to create the functions to originate the map, points and respective paths. On device ready it is defined the map and the points with their coordinates and other information (Listing 5.3).

```
var div = document.getElementById("map_canvas");
map = plugin.google.maps.Map.getMap(div);

pontos = [
  {"latlng": [40.2805882, -7.5043414],
   name: "Pelourinho",
   Percurso1 : 1,
   Percurso2 : 1,
   Percurso3 : 1},},
```

```

    {"latlng": [40.280711, -7.503644],
     name: "Igreja da Misericórdia",
     Percurso1 : 1,
     Percurso2 : 1,
     Percurso3 : 1},

    //etc
  ];

```

Listing 5.3: Definition of POIs

When the paths are equal to 1 (e.g. PercursoX: 1) this means that the point is within that path. If it is equal to 0 then the point is not contained on the path. The category will change the path according to the user's selection, making the markers for that path visible and the other ones invisible, and it is defined that *Percurso1* is defined as the default path. Furthermore, the user's current location is needed to be shown on the map. If getting the user location is successful, a marker is added to the map. If it is not successful, a message is shown saying that it was impossible to get user's current location. The parameter *option* on *getMyLocation* enables high accuracy of the GPS.

```

var bounds = pontos.map(

  function(info) {
    var latLng = new plugin.google.maps.LatLng
      (info.latlng[0], info.latlng[1]);
    map.addMarker({
      "position": latLng,
      "title": info.name
    },
    function(marker) {
      map.on("category_change",
        function(category) {
          marker.setVisible(
            info[category] ? true : false);
        });
    });
    return latLng;
  });

map.trigger("category_change", "Percurso1");

map.getMyLocation(option, onSuccess, onError);

```

Listing 5.4: Create Map

### 5.3.3 Image Recognition

To perform the image recognition, the first step needed was to generate a target collection on *Wikitude Target Manager*.

To have good results the images should have high contrast, there should be evenly distributed textured areas and many corner like structures, repetitive patterns should be avoided, such as large single-colored areas or large amounts of text. The images should not be of smaller dimensions than 500 pixels as they would not contain enough graphical information, or larger than 1000 pixels as they would not improve the tracking quality [Wik15a].

In this project a target collection of ten different images from different places of Covilhã was created, such as Saint Maria Church, San Francisco Church, Theatre, City Council, Mercy Church, Wool Museum and some more.

The *Wikitude* will run on Android devices that fulfil the following requirements [Wik15c]:

- Android 4.0+ (API Level 14+)
- High resolution devices (hdpi)
- Rear-facing camera
- Devices with a capable CPU (armv7a and NEON support)

As far as *HTML* is concerned in this case it is just necessary to load the architect world and call the *JavaScript* code to perform the image recognition and to display augmented information (AR). It was also added a button to a page with the images of the places where the user can perform the image recognition, without this page the user could not have any idea of the places where it was possible to use the app (Figure 5.5).

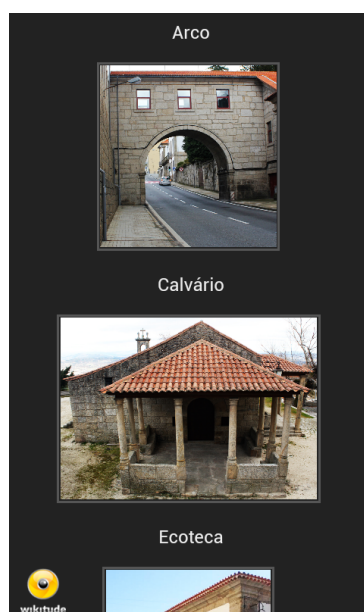


Figure 5.5: Page with the pictures of the places with Image Recognition



Figure 5.6: Image Recognition with information in AR

In what refers to the *JavaScript* code, first of all, in order to load the target collection, it is necessary to download it from the *Wikitude Target Manager*.

```

this.tracker = new
  AR.ClientTracker("assets/targetcollections.wtc", {
    onLoaded: this.worldLoaded
  });

```

Listing 5.5: Target Collection load

Then for each target it is going to be created the augmented reality content.

To the majority of targets it was developed an HTML page with some information about the place (Figure 5.6), for instance in the listing 5.6, it is called a button that will open the main page of the theatre, and the page with information about the theatre.

```

// Create HTML and button AR

var T = new AR.Trackable2DObject(this.tracker, "teatro", {
  drawables: {
    cam: [teatroButton, teatro]
  }
});

// Create button to page with information

var teatroButton = this.createWwwButton
("http://www.cm-covilha.pt/simples/?f=4815", 0.15, {
  offsetX: -0.3,
  offsetY: -0.25,

```

```

        zOrder: 1
    });

    // Import HTML page

    var teatro = new AR.HtmlDrawable({
        uri: "assets/pages/teatro.html"
    }, 0.5, {
        viewportWidth: 250,
        viewportHeight: 450,
        backgroundColor: "#000000",
        offsetX: +0.4,
        offsetY: 0.5,
        horizontalAnchor:
            AR.CONST.HORIZONTAL_ANCHOR.RIGHT,
        verticalAnchor:
            AR.CONST.VERTICAL_ANCHOR.TOP,
        clickThroughEnabled: true,
        allowDocumentLocationChanges: false,
        onDocumentLocationChanged: function
            onDocumentLocationChangedFn(uri) {
                AR.context.openInBrowser(uri);
            }
    });

```

Listing 5.6: Creating HTML information page and button

Another option is to insert a video (Figure 5.7) instead of the html page and the button. The listing 5.7 calls the video and the *playButton*. It also controls the pause and resume of the video according to the states of the video.

```

// Create Video Augmented Reality

var ML = new AR.Trackable2DObject(this.tracker, "Museu", {
    drawables: {
        cam: [video, playButton]
    },
    onEnterFieldOfVision:
        function onEnterFieldOfVisionFn () {
            if (video.playing) {
                video.resume();
            }
        },
    onExitFieldOfVision:
        function onExitFieldOfVisionFn () {
            if (video.playing) {
                video.pause();
            }
        }
});

```

```

        }
    }
});

// Import Video

var video = new AR.VideoDrawable("assets/contents/museu.mp4", 0.5,
{
    offsetY: playButton.offsetY,
    onLoad: function videoLoaded() {
        playButton.enabled = true;
    },
    onPlaybackStarted: function videoPlaying() {
        playButton.enabled = false;
        video.enabled = true;
    },
    onFinishedPlaying: function videoFinished() {
        playButton.enabled = true;
        video.playing = false;
        video.enabled = false;
    },
    onClick: function videoClicked() {
        if (playButton.clicked) {
            playButton.clicked = false;
        } else if (video.playing) {
            video.pause();
            video.playing = false;
        } else {
            video.resume();
            video.playing = true;
        }
    }
});

// Create play button

var playButtonImg = new AR.ImageResource("assets/playButton.png");
var playButton = new AR.ImageDrawable(playButtonImg, 0.3, {
    enabled: false,
    clicked: false,
    onClick: function playButtonClicked() {
        playButton.clicked = true;
    },
    offsetY: -0.3
});

```

Listing 5.7: Creating video content and respective play button

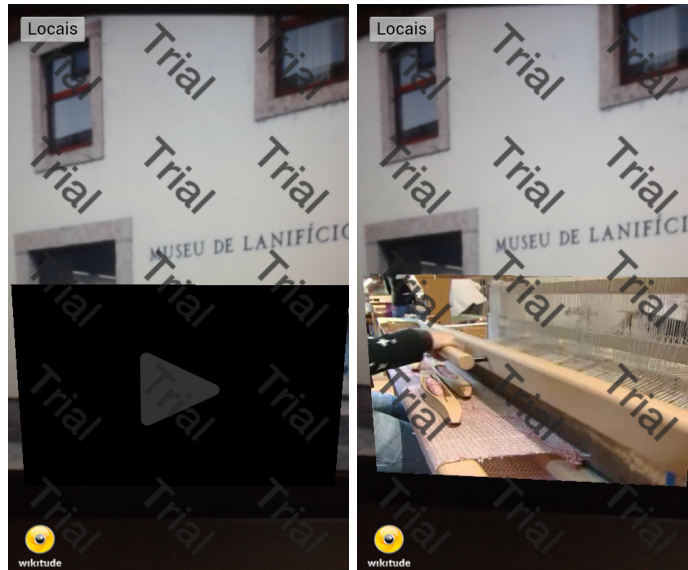


Figure 5.7: Image Recognition with video content in AR

Finally it is possible to add 3D objects (Figure 5.8) and animated 3D objects (Listing 5.8), which need to be converted to *wt3* using the *Wikitude 3D decoder tool*.

```
// Import 3D object

this.modelArco = new AR.Model("assets/contents/cuboverde.wt3", {
    onLoad: this.loadingStep,

    scale: {
        x: 0.05,
        y: 0.05,
        z: 0.05
    },
    translate: {
        x: 0.0,
        y: 0.0,
        z: -0.5
    },
    rotate: {
        tilt: -90,
    },

    // Start animation on click

    onClick : function (drawable){
        anim.start();
    }
}
```

```
});  
  
// Import animation  
  
var anim = new AR.ModelAnimation( this.modelArco, "Cube_animation"  
    );
```

Listing 5.8: Adding 3D object and animation



Figure 5.8: Image Recognition with animated 3D object in AR

### 5.3.4 Near You

The *Near You* shows using Augmented Reality the POIs around the user at a distance he defines. Even though it initially shows all the POIs, the user can change the radius.

It is essential that the user's Android device fulfils the following requirements [Wik15c]:

- Android 4.0+ (API Level 14+)
- Compass
- GPS and / or network positioning
- Accelerometer
- High resolution devices (hdpi)
- Rear-facing camera



Figure 5.9: The radius menu, the POIs and the information shown when they are clicked

- OpenGL 2.0

The first thing to do is to define the places and give information about them (Listing 5.9), so the *Wikitude* can show them on the right coordinates.

```
var myJsonData = [{
    "id": "1",
    "longitude": "-7.5043414",
    "latitude": "40.2805882",
    "description": "Pelourinho da Covilhã",
    "altitude": "670.0",
    "name": "Pelourinho"
}, {
    "id": "4",
    "longitude": "-7.5044621",
    "latitude": "40.2827572",
    "description": "A Igreja de Nossa Senhora da
    Conceição fazia parte do antigo Convento de
    São Francisco e data dos primeiros tempos da
    monarquia",
    "altitude": "661.0",
    "name": "Igreja São Francisco"
},
// (...)
}];
```

Listing 5.9: Places definition

After the definition of POIs one should create a file called *Marker.js* where the marker will be defined. The listing 5.10 loads the POIs from the local data and will be called within *request-DataFromLocalFn*.

```

requestDataFromLocal: function requestDataFromLocalFn(lat, lon,
    alt) {
    World.loadPoisFromJsonData(myJsonData);
},

```

Listing 5.10: Loading the POIs from the local data

The function *loadPoisFromJsonData* (Listing 5.11) will create an array (*MarkerList*) with marker objects.

```

// called to inject new POI data
loadPoisFromJsonData: function loadPoisFromJsonDataFn(poiData) {

    // empty list of visible markers
    World.markerList = [];

    // start loading marker assets
    World.markerDrawable_idle =
        new AR.ImageResource("assets/marker_idle.png");

    // loop through POI-information and create an
    // AR.GeoObject (=Marker) per POI

    for (var currentPlaceNr = 0; currentPlaceNr < poiData.length;
        currentPlaceNr++) {
        var singlePoi = {
            "id": poiData[currentPlaceNr].id,
            "latitude": parseFloat(
                poiData[currentPlaceNr].latitude),
            "longitude": parseFloat(
                poiData[currentPlaceNr].longitude),
            "altitude": parseFloat(
                poiData[currentPlaceNr].altitude),
            "title": poiData[currentPlaceNr].name,
            "description":
                poiData[currentPlaceNr].description
        };

        World.markerList.push(new Marker(singlePoi));
    }

    World.updateStatusMessage(
        currentPlaceNr + ' places loaded');
}

```

Listing 5.11: *loadPoisFromJsonDataFn* function

#### 5.3.4.1 Adding Radar

To add the radar one should define a div in the *HTML* file, like is shown in listing 5.12.

```
<div class="radarContainer_left" id="radarContainer"></div>
```

Listing 5.12: Radar div definition

The radar should be defined after that on a file with the name *Radar.js*. In this file it will be defined the radar design like the background and north indicator images, and custom actions as when the radar is clicked or to set the maximum distance, for instance when the user changes the radius on the range panel.

The radar also has a *CSS* file to define the radar position on the screen. To show the radar it is crucial to call the *PoiRadar.show()* function *onloadPoisFromJsonData*.

#### 5.3.4.2 Adding Range Panel

To add the range panel it is necessary to create a header and a panel on the *HTML* file (Listing 5.13). In the header when the user clicks on the *Raio* button it calls the function *World.showRange()* (Listing 5.14) that will open the panel and allow the user to change the current range.

```
<div id="header-status" data-role="header" data-position="fixed"
data-theme="c">
  <a href="javascript: World.showRange();" data-icon="gear"
  data-inline="true" data-mini="true">Raio</a>
</div>
```

Listing 5.13: Near You header

```
showRange: function showRangeFn() {
  if (World.markerList.length > 0) {

    // update labels on every range movement
    $('#panel-distance-range').change(function() {
      World.updateRangeValues();
    });

    World.updateRangeValues();
    World.handlePanelMovements();

    // open panel
    $('#panel-distance').trigger("updatelayout");
    $('#panel-distance').panel("open", 1234);
  } else {

    // no places are visible,
    // because they are not loaded yet
    World.updateStatusMessage('Os pontos ainda não foram
```

```

        carregados', true);
    }
},

```

Listing 5.14: *showRangeFn* function

The functions *World.updateRangeValues()* and *World.handlePanelMovements()* also need to be created.

The first one updates the range values according to what the user defines, getting the value on the slider to update it. It also updates the radar maximum distance so that the points shown are just the ones at the range.

The second one handles the opening and closing of the panel when the user clicks on the button to open it or to close it. The range panel *HTML* code is shown in listing 5.15.

```

<div data-role="panel" id="panel-distance" data-position="left"
data-display="overlay" style="background-color:#F0F0F0;"
    data-theme="c">

    <!-- header with close button -->
    <div data-role="header" data-theme="c">
        <h1>Raio</h1>
        <a href="#header" data-rel="close">Fechar</a>
    </div>

    <!-- distance information, calculated/updated in code -->
    <div data-role="content">

        <!-- Range in m/km-->
        <h4> Raio: <a id="panel-distance-value"></a></h4>

        <!-- Amount of visible places -->
        <h4> Visiveis: <a id="panel-distance-places"></a></h4>
        <!-- default slider -->
        <input id="panel-distance-range" type="range"
            data-highlight="true" name="rangeSlider" min="0"
            max="100" value="100" data-show-value="false"
            step="5" data-popup-enabled="false">
    </div>
</div>

```

Listing 5.15: Panel with range information

To initialize the range in 100% in *loadPoisFromJsonData* it is necessary to add the listing 5.16.

```

$("#panel-distance-range").val(100);
$("#panel-distance-range").slider("refresh");

```

Listing 5.16: Initialize the range in 100%

### 5.3.4.3 Adding Panel containing POI information

To add this panel, the POI object first needs to be selectable. In order to do this, a function called *onMarkerSelectedFn* (Listing 5.17) was implemented to show the selected POI information on the panel.

```
onMarkerSelected: function onMarkerSelectedFn(marker) {
    World.currentMarker = marker;

    // update panel values
    $("#poi-detail-title").html(marker.poiData.title);
    $("#poi-detail-description").html(marker.poiData.description);

    var distanceToUserValue = (marker.distanceToUser > 999) ?
        ((marker.distanceToUser / 1000).toFixed(2) + " km") :
        (Math.round(marker.distanceToUser) + " m");

    $("#poi-detail-distance").html(distanceToUserValue);

    // show panel
    $("#panel-poidetail").panel("open", 123);

    $(".ui-panel-dismiss").unbind("mousedown");

    $("#panel-poidetail").on("panelbeforeclose", function(event,
        ui)
    {
        World.currentMarker.setDeselected(World.currentMarker);
    });
},
```

Listing 5.17: Function *onMarkerSelectedFn()*

This panel shows the distance to the user's current location, the name of the POI and a brief description of the place (Listing 5.18).

```
<div data-role="panel" id="panel-poidetail" data-position="right"
data-display="overlay" style="background-color:#F0F0F0;"
data-theme="c">

    <!-- header with "close" button -->
    <div data-role="header" data-theme="c">
        <h1>Detalhes</h1>
        <a href="#header" data-rel="close">Fechar</a>
    </div>

    <!-- content of POI detail page -->
    <div data-role="content">
```

```

        <!-- title -->
        <h3 id="poi-detail-title"></h3>

        <!-- description -->
        <h4 id="poi-detail-description"></h4>

        <!-- distance -->
        <h4>Distância: <a id="poi-detail-distance"></a></h4>
    </div>
</div>

```

Listing 5.18: Panel with POI information

### 5.3.5 Directions

The *Directions* option gives the coordinates of the current user's location and the place to *Maps* that creates the direction to that place from the user location.

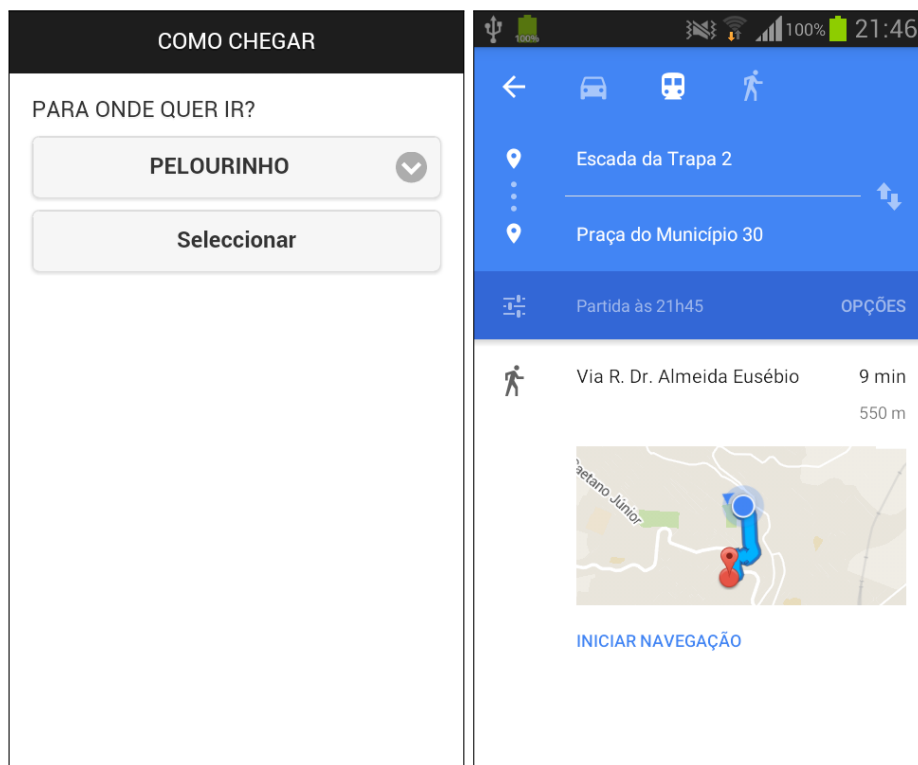


Figure 5.10: The directions to the desired place with Maps

This option is also on the same *HTML* file as the initial menu and the *Paths*. In *HTML* it is created a header with the page name and a combo box with the possible places. When the user clicks the button *Seleccionar* the app gets the places coordinates from the variable *pontos* (Section 5.10) as well as the current user location and sends it to *Maps* using the function *goToMaps()* (Listing 5.19). The function *selectd* (Listing 5.19) is the one responsible for getting the coordinates of the places.

```
function selectd(toLocation) {
    var selectDom = document.getElementById("categorySelector");
    var category =
        selectDom.options[selectDom.selectedIndex].text;

    var bounds = pontos.map(function(info) {
        if(info.name=== category){
            toLocation = new plugin.google.maps.LatLng(
                info.latlng[0], info.latlng[1]);
        }
    });

    return toLocation;
};

function goToMaps(){
    var toLocation;

    map.getMyLocation(onSuccess, onError);

    toLocation=selectd(toLocation);

    plugin.google.maps.external.launchNavigation({
        "from": yourLocation,
        "to": toLocation
    });
};
```

Listing 5.19: Functions *goToMaps()* and *selectd()*

## Chapter 6

### Conclusions and Future Work

The main conclusions are that augmented reality is increasing exponentially and in the future it will certainly have a much wider range of applications. Furthermore, the use of AR in the field of tourism allows a more interactive way of visiting cities and getting information about specific places. Through the recognition of buildings it gives the user various informations, such as text, videos or 3D animations, which can be reached just by using a personal smartphone, tablet or other device. The possibilities are endless, especially when associated with geolocation systems. Discover Covilhã is a prototype in tourism area that shows the potentialities of combining AR and geolocation.

However, the growing of AR goes far beyond the tourism area. It can be currently used in many other fields such as education, medicine, military, entertainment and many others, as it is possible to read in Section 4.3.

To improve this work the interface design should be improved. Being the application a prototype, all the routes from Covilhã should be added as different paths, which would make the app more complete and a complete guide of the city. Another important improvement would be the addition of better, more informative and interactive contents about the different places, such as some videos or 3D models of the buildings and image galleries.

The option to change the language would also be an important addition, since Covilhã receives numerous foreign students and tourists from different places in the world. It would be wise to be possible to change the language at least to English.

After all these improvements it is envisioned to publish the app on the Play Store so everyone can install it and use it. Since the app was not published it could not be tested by people from different ages, even though it was tested with different devices. In conclusion to this tests, the app was recognizing the buildings but the direct sunlight to the camera could lead to some trouble.

Finally, a goal to this project is to make it available for iOS and Windows Phone so it would be accessible to a larger amount of people without OS restrictions.



## Bibliography

- [Anda] Adding sdk packages [online]. Available from: <http://developer.android.com/sdk/installing/adding-packages.html> [cited 15 May 2015]. 7
- [ANDb] Android, the world's most popular mobile platform [online]. Available from: <https://developer.android.com/about/index.html> [cited 5 May 2015]. 3
- [Ara14] Arasam [online]. 3 June 2014. Available from: <http://www.realtaumentata.it/blog/life/sociale>. xiii, 20
- [ARS] ARSOFT. Augmented reality in tourism [online]. Available from: <http://www.arsoft-company.com/en/tourism/> [cited 12 April 2015]. xiii, 21
- [ART] Configuring art [online]. Available from: <http://source.android.com/devices/tech/dalvik/configure.html> [cited 15 May 2015]. 6
- [Azu97] Ronald Azuma. A survey of augmented reality. *Teleoperators and Virtual Environments*, 6(4):355–385, 1997. 18, 20
- [BCL15] M. Billinghurst, A. Clark, and G. Lee. A survey of augmented reality. *Foundations and Trends in Human-Computer Interaction*, 8(2-3):73–272, 2015. 15, 16, 17
- [Bla12] Blandford, Rafe. Nokia city lens 1.5 beta for windows phone 7.5 [online]. 28 November 2012. Available from: [http://allaboutwindowsphone.com/flow/item/16287\\_Nokia\\_City\\_Lens\\_1\\_5\\_beta\\_for\\_Wi.php](http://allaboutwindowsphone.com/flow/item/16287_Nokia_City_Lens_1_5_beta_for_Wi.php) [cited 10 April 2015]. xiii, 16
- [Dal] Art and dalvik [online]. Available from: <http://source.android.com/devices/tech/dalvik/index.html> [cited 15 May 2015]. 6
- [Gra14] Graz University of Technology. Media/press [online]. 2014. Available from: [https://handheldar.icg.tugraz.at/media\\_press.php](https://handheldar.icg.tugraz.at/media_press.php) [cited 10 April 2015]. xiii, 16
- [Jaw14] Jawaid, Zehra. Tours come alive with augmented reality [online]. 3 January 2014. Available from: <http://www.epluxinsights.com/tours-come-alive-with-augmented-reality/> [cited 15 May 2015]. xiii, 21, 22
- [Joh15] John, Shiju P. Android architecture [online]. 3 February 2015. Available from: <http://www.eazytutz.com/android/android-architecture/> [cited 15 May 2015]. xiii, 4
- [Kat15] Katsumata, Masashi. Tutorial for windows [online]. 2014-2015. Available from: <https://github.com/wf9a5m75/phonegap-googlemaps-plugin/wiki/Tutorial-for-Windows> [cited 10 April 2015]. 12
- [Kim14] Kim, Bryan [online]. 22 July 2014. Available from: <http://www.hostingreviewbox.com/5-amazing-augmented-reality-ar-apps-android-ios-device/>. xiii, 21
- [KK14] Young-geun Kim and Won-jung Kim. Implementation of augmented reality system for smartphone advertisements. *International Journal of Multimedia and Ubiquitous Engineering*, 9(2):385–392, 2014. 15, 16

- [MTUK94] Paul Milgram, Haruo Takemura, Akira Utsumi?, and Fumio Kishino. Augmented reality: A class of displays on the reality-virtuality continuum. *Telem Manipulator and Telepresence Technologies*, 2351(2-3):282–292, 1994. xiii, 15
- [Nes13] Nesloney, Todd. Augmented reality brings new dimensions to learning [online]. 2013. Available from: <http://www.edutopia.org/blog/augmented-reality-new-dimensions-learning-drew-minock> [cited 21 April 2015]. 20, 21
- [OB14] Ricardo da Silva Ogliari and Robison Cris Brito. *Android - Do Básico ao Avançado*. Ciência Moderna Ltda., 2014. 3, 6
- [Out14] Outplay Entertainment. Monstar truck [online]. 2013-2014. Available from: <http://outplay.com/games/monstar-truck> [cited 21 April 2015]. xiii, 19
- [PM06] M. Pressigout and É Marchand. Hybrid tracking algorithms for planar and non-planar structures subject to illumination changes. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2006)*. IEEE Computer Society, 2006. 16, 17
- [Pri14] Prigg, Mark. Google glass for war: The us military funded smart helmet that can beam information to soldiers on the battlefield [online]. 2014. Available from: <http://www.dailymail.co.uk/sciencetech/article-2640869/Google-glass-war-US-military-reveals-augmented-reality-soldiers.html> [cited 21 April 2015]. xiii, 20
- [Que] Queens University Perk Lab. Augmented reality image overlay guidance for musculoskeletal procedures [online]. Available from: <http://perk.cs.queensu.ca/contents/augmented-reality-image-overlay-guidance-musculoskeletal-procedures> [cited 20 April 2015]. xiii, 18
- [Rab07] Ramon Ribeiro Rabello. *Android: um novo paradigma de desenvolvimento móvel*, 2007. 4, 5
- [Rob13] Peter Roberts. *Augmented reality interactive marketing applications for the hospitality industry*, 2013. 17, 18
- [Son15] Sony Computer Entertainment America LLC. *Wonderbook: Book of spells* [online]. 2015. Available from: <https://www.playstation.com/en-us/games/wonderbook-book-of-spells-ps3/> [cited 21 April 2015]. xiii, 19
- [Tay15] Taylor. *Memorable moments in mobile history* [online]. 19 March 2015. Available from: <http://appinstitute.co.uk/blog/memorable-moments-in-mobile-history/> [cited 15 May 2015]. xiii, 3
- [The13a] The Apache Software Foundation. *About apache cordova* [online]. 2012-2013. Available from: <https://cordova.apache.org/> [cited 10 April 2015]. 9
- [The13b] The Apache Software Foundation. *The command-line interface* [online]. 2012-2013. Available from: [http://cordova.apache.org/docs/en/5.0.0/guide\\_cli\\_index.md.html](http://cordova.apache.org/docs/en/5.0.0/guide_cli_index.md.html) [cited 10 April 2015]. 9, 10, 11
- [vKP10] D.W.F. van Krevelen and R Poelman. A survey of augmented reality technologies, applications and limitations. *The International Journal of Virtual Reality*, 9(2):1–20, 2010. 15

## MARA - Mobile Augmented Reality for Advertising (Discover Covilhã)

- [Web13] Webcam Social Shopper. The shopper experience [online]. 2009-2013. Available from: [http://webcamsocialshopper.com/wp-content/uploads/2013/02/Header\\_Virtual\\_Dressing\\_Room-703x200.jpg](http://webcamsocialshopper.com/wp-content/uploads/2013/02/Header_Virtual_Dressing_Room-703x200.jpg) [cited 20 April 2015]. xiii, 18
- [Wik15a] Wikitude GmbH. Best practice for target images [online]. 2014-2015. Available from: <http://www.wikitude.com/developer/documentation/android> [cited 29 May 2015]. 28
- [Wik15b] Wikitude GmbH. Setup guide cordova cli [online]. 2014-2015. Available from: <http://www.wikitude.com/developer/documentation/phonegap> [cited 10 April 2015]. 13
- [Wik15c] Wikitude GmbH. Supported android devices [online]. 2014-2015. Available from: <http://www.wikitude.com/developer/documentation/android> [cited 29 May 2015]. 28, 33
- [ZDB08] Feng Zhou, Henry Duh, and Mark Billinghurst. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR 2008)*. IEEE Computer Society, 2008. Available from: <http://www.computer.org/portal/web/csdl/doi/10.1109/ISMAR.2008.4637362>. 15, 17

