Developing an application presenting a showroom in Virtual Reality

Françoise Tith

Student ID number: H00199330
August 2015

School of Mathematical and Computer Sciences

Dissertation submitted as part of the requirements for the award of the degree of MSc. in Information Technology (Software Systems)
Declaration of Authorship

I, Françoise Titth, declare that this thesis titled, 'Developing an application presenting a showroom in Virtual Reality' and the work presented in it is my own. I confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed: __________________________________________

Date: __________________________________________
Abstract

Smart Electric Lyon (SEL) is a European experimentation project in the city of Lyon steered by EDF. It gathers 21 industrial partners and also researchers who develop smartgrid experimentations to test new forms of energy consumptions in the city. It is 4-years study whose goal is to establish a code of good practice based on sociological, statistical and behavioural analysis.

To present this programme to customers and partners, a showroom is available in Lyon. In order to reach a larger audience, the showroom should be digitally available.

My master project consists of developing an application presenting this showroom. A good way to captivate people’s attention is to use virtual reality, since that technology is fascinating, and became accessible to a large audience with the introduction of the cheap headset Google Cardboard. The use of sounds, videos, and games is also a good method to interest people.
Acknowledgements

Firstly, I would like to express my gratitude to my school supervisor, Dr Hans-Wolfgang Loidl for his guidance and help choose an appropriate placement, and his immense patience with the particular conditions that it implies.

Besides my school supervisor, my deepest thanks goes to my company supervisors: Stephane Rouilly and Geoffroy Deleury for their guidance, ideas and knowledge during the placement. It was really rewarding and a pleasure to work with accomplished engineers.

In parallel, I would like to thank Smart Electric Lyon team, in particular Nam NGUYEN, for their proposition and their feedbacks on the project.

My sincere thanks also goes to the educational team of Heriot-Watt University, in particular Dr Hamish Taylor; and ESIEE Paris for making the placement possible.

Finally, I would like to thank the whole staff of EDF for their warm welcome, the interesting discussions and good ambiance.
# Contents

- Declaration of Authorship ........................................................................................................... ii
- Abstract ........................................................................................................................................ iii
- Acknowledgements ..................................................................................................................... iv
- Contents ......................................................................................................................................... v
- List of Figures ............................................................................................................................... viii
- List of Tables ................................................................................................................................. x
- Abbreviations ................................................................................................................................. xi

1 Introduction .................................................................................................................................... 1

1.1 Context ........................................................................................................................................ 1
1.2 Aims ............................................................................................................................................. 2
1.3 My participation .......................................................................................................................... 3

2 Literature review ......................................................................................................................... 4

2.1 Virtual reality ............................................................................................................................... 4

2.1.1 Interfaces ................................................................................................................................. 4
2.1.2 Purposes and application ......................................................................................................... 6
2.1.3 Hardwares ............................................................................................................................... 7
2.1.3.1 2015 situation of head-mounted-display (HMD) ................................................................. 7
2.1.3.2 Oculus Rift ......................................................................................................................... 9
2.1.3.3 Google Cardboard .............................................................................................................. 9

2.2 Existing softwares for virtual reality .......................................................................................... 11

2.2.1 Oculus SDK .......................................................................................................................... 11
2.2.2 Google cardboard SDK .......................................................................................................... 12
2.2.3 Web VR solution .................................................................................................................... 12

2.3 Remote control .......................................................................................................................... 15

2.4 Stream video feedback .............................................................................................................. 16

2.4.1 Motion .................................................................................................................................... 16
2.4.2 mjpg-streamer ......................................................................................................................... 16
2.4.3 FFMpeg ................................................................................................................................... 16
2.4.4 VLC ............................................. 17

3 Methodology ................................ 18
  3.1 Baseline evaluation of possible technologies ....................... 18
  3.2 Metrics evaluation of 3D VR .................................. 18

4 Requirement analysis .............................. 21
  4.1 Mandatory and optional requirements ............................. 21
    4.1.1 Mandatory requirements .................................. 21
    4.1.2 Optional requirements .................................... 21
  4.2 User requirements ......................................... 22
  4.3 Software and technology requirements ........................... 22

5 Implementation ..................................... 24
  5.1 General organisation ....................................... 24
    5.1.1 Development versions .................................... 24
    5.1.2 Files and folders ....................................... 24
    5.1.3 Virtual Tour ........................................... 25
      5.1.3.1 XML Structure ...................................... 27
      5.1.3.2 JavaScript and kpano ................................. 29
  5.2 Functionalities .......................................... 29
    5.2.1 Virtual tour ........................................... 29
      5.2.1.1 Architecture / first version ......................... 29
      5.2.1.2 Title in every image ................................ 29
      5.2.1.3 Triggers hotspots .................................. 31
      5.2.1.4 Animation of triggers hotspots ...................... 34
      5.2.1.5 Testers’ quotes: verbatims .......................... 35
    5.2.2 Stereoscopy ............................................ 36
    5.2.3 Sound records .......................................... 37
    5.2.4 Game .................................................. 38
    5.2.5 Remote control: Stereolab control ........................ 42
      5.2.5.1 General connection .................................. 43
      5.2.5.2 Raspberry Pi setup .................................. 44
      5.2.5.3 Physical control ..................................... 45
      5.2.5.4 Virtual Control ..................................... 46
    5.2.6 Video feedback ......................................... 48
  5.3 Deliverables versions ...................................... 49
    5.3.1 Web version ........................................... 49
    5.3.2 App versions ........................................... 49
      5.3.2.1 Apache Cordova ..................................... 50
      5.3.2.2 Publication on the store ............................. 51
      5.3.2.3 Import from desktop version to App version ........ 51
  5.4 Additional work ......................................... 51

6 Results and reflections .............................. 53
  6.1 Deliverables ........................................... 53
  6.2 Stakeholders and their needs ................................ 55
  6.3 Project Schedule: Gantt Chart .............................. 57
6.4 Qualitative evaluation of possible technologies . . . . . . . . . . . . . . . . 58
  6.4.1 Software for virtual tour . . . . . . . . . . . . . . . . . . . . . . . . 58
  6.4.2 Electronic board . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
  6.4.3 Camera control . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
6.5 Metrics evaluation of 3D VR . . . . . . . . . . . . . . . . . . . . . . . . . 60
  6.5.1 Raw results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
  6.5.2 Hypothesis 1: User experience is related to stereoscopic view . . . . . 60
    6.5.2.1 Descriptive Statistics . . . . . . . . . . . . . . . . . . . . . . . 60
    6.5.2.2 Statistically significant test . . . . . . . . . . . . . . . . . . 61
  6.5.3 Hypothesis 2: Time of use is related to stereoscopic view . . . . . . . 62
  6.5.4 Testing conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
6.6 Improvements and evolution . . . . . . . . . . . . . . . . . . . . . . . . . . 63

7 Professional, Legal, Ethical and Social considerations 65
  7.1 Legal issues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
    7.1.1 Disclaimer and instruction of use . . . . . . . . . . . . . . . . . . . 65
    7.1.2 Licensing . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
      7.1.2.1 krPano/Panotour . . . . . . . . . . . . . . . . . . . . . . . . 66
      7.1.2.2 Web server for remote control . . . . . . . . . . . . . . . . 66
      7.1.2.3 Motion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
      7.1.2.4 Google Cardboard . . . . . . . . . . . . . . . . . . . . . . . 66
    7.1.3 Trademark . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66
    7.1.4 Copyright . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
  7.2 Professional issues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
    7.2.1 British Computer Society (BCS) Code of Practice . . . . . . . . . . 67
  7.3 Ethical issues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
  7.4 Social issues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 68
    7.4.1 Virtual reality risk . . . . . . . . . . . . . . . . . . . . . . . . . . 68
    7.4.2 Possible misconceptions . . . . . . . . . . . . . . . . . . . . . . . 68

8 Conclusion 69
  8.1 General outcomes of the project and technical issues . . . . . . . . . . . . . 69
  8.2 User results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
  8.3 Experience acquired during the placement . . . . . . . . . . . . . . . . . . . 71

Bibliography 72

A Stereolab prototype photos 74

B Questionnaire 77

C Scripts 79
  C.1 Import script from desktop to app . . . . . . . . . . . . . . . . . . . . . . 79
## List of Figures

2.1 Workbench of two screens ........................................ 5  
2.2 Spheric cave ......................................................... 6  
2.3 Oculus Rift, Morpheus, Vive ...................................... 7  
2.4 Samsung Gear VR and Google Cardboard ......................... 8  
2.5 C1-GLASS .......................................................... 9  
2.6 krpano droplet ....................................................... 13  
2.7 Panotour Interface .................................................. 14  
3.1 Decision tree for test .............................................. 20  
5.1 Hotspot .............................................................. 26  
5.2 More information hotspot .......................................... 31  
5.3 Visual triggers ....................................................... 34  
5.4 Records ............................................................... 34  
5.5 Stereoscopy .......................................................... 36  
5.6 Menu ................................................................. 36  
5.7 Wall of testers ....................................................... 38  
5.8 9-images-wall ....................................................... 39  
5.9 Game on ............................................................. 40  
5.10 Stereolab Modules .................................................. 43
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.11</td>
<td>Eletronic Scheme</td>
<td>44</td>
</tr>
<tr>
<td>5.12</td>
<td>Stereolab Box</td>
<td>46</td>
</tr>
<tr>
<td>5.13</td>
<td>Stereolab devices</td>
<td>47</td>
</tr>
<tr>
<td>5.14</td>
<td>Stereolab video</td>
<td>49</td>
</tr>
<tr>
<td>6.1</td>
<td>Sticker distributed during ADEME presentation</td>
<td>54</td>
</tr>
<tr>
<td>6.2</td>
<td>Comparison Raspberry 1 and 2 for Motion</td>
<td>59</td>
</tr>
<tr>
<td>A.1</td>
<td>Stereolab</td>
<td>74</td>
</tr>
<tr>
<td>A.2</td>
<td>Front Stereolab</td>
<td>75</td>
</tr>
<tr>
<td>A.3</td>
<td>Back Stereolab</td>
<td>75</td>
</tr>
<tr>
<td>A.4</td>
<td>Side Stereolab</td>
<td>76</td>
</tr>
<tr>
<td>B.1</td>
<td>Questionnaire</td>
<td>78</td>
</tr>
</tbody>
</table>
List of Tables

6.1 Questionnaire raw results ............................................. 59
6.1 Questionnaire raw results ............................................. 60
6.2 Scores given the display view ........................................ 60
6.3 User experience given the display view ............................ 61
6.4 Chi Square distribution table for DF = 2 .......................... 61
6.5 Expected values for chi square ................................. 62
## Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADB</td>
<td>Android Debug Bridge</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>C-R-T</td>
<td>Competence-Responsibility-Trust</td>
</tr>
<tr>
<td>CORS</td>
<td>Cross-Origin Resource Sharing</td>
</tr>
<tr>
<td>DV</td>
<td>Dependent Variable</td>
</tr>
<tr>
<td>DF</td>
<td>Degree of Freedom</td>
</tr>
<tr>
<td>EDF</td>
<td>Électricité de France</td>
</tr>
<tr>
<td>FPS</td>
<td>Frame Per Second</td>
</tr>
<tr>
<td>GDC</td>
<td>Game Developers Conference</td>
</tr>
<tr>
<td>HF&amp;E</td>
<td>Human factors and ergonomics</td>
</tr>
<tr>
<td>HMD</td>
<td>Head Mounted Display</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technology</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated/Interactive Development Environment</td>
</tr>
<tr>
<td>IPD</td>
<td>Interpupillary Distance</td>
</tr>
<tr>
<td>IV</td>
<td>Independent Variable</td>
</tr>
<tr>
<td>JS</td>
<td>JavaScript</td>
</tr>
<tr>
<td>NDA</td>
<td>Non-Disclosure Agreement</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>PLES(i)</td>
<td>Professional Legal Ethical Social (Issue)</td>
</tr>
<tr>
<td>POC</td>
<td>Proof Of Concept</td>
</tr>
<tr>
<td>SDK</td>
<td>Software Development Kit</td>
</tr>
<tr>
<td>SEL</td>
<td>Smart Electric Lyon</td>
</tr>
<tr>
<td>VR</td>
<td>Virtual Reality</td>
</tr>
<tr>
<td>WBS</td>
<td>Work Breakdown Structure</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Context

For my master project, I have chosen to work in a company. The placement takes place at the R&D direction of EDF (Électricité de France), at ICAME department (Innovation Commerciale, Analyse des Marchés et de leur Environnement, in English: Commercial Innovation and Commercial Analysis of Market and their Environment). ICAME department is composed of 100 engineers. The projects led by this department are developed with engineers, developers, designers, economists, statisticians and sociologists.

The team where I am working is composed of 3 persons. The application background is a partnership between stakeholders in energy, telecommunication, construction industry and universities to set up and present smartgrid experimentations at Lyon. Every steps of the program have to be validated by the person in charge of the communication in the partnership Smart Electric Lyon (SEL), that is to say, Gilles Cerardi. The goal of the general project is to present the smartgrid experimentations thanks to a showroom. This showroom is in Lyon and the aim is to make it accessible through a virtual visit.

I was supposed to work on the representation of different kind of POCs (Proof Of Concept) concerning smartgrids. Unfortunately, since those POCs are not clearly defined yet the idea was postponed. Thus, my supervisors proposed to SEL a virtual showroom. They accepted and I was entrusted to participate to the development of that project.


1.2 Aims

As explained in the previous section, the aims of the project are to present the showroom virtually in order to

- make people want to visit the showroom
- make it available to people who are geographically far away
- help people who visited the showroom to see what they missed.

Since the department wants the most generic solution, we decided to use cross-platform technologies, especially web technologies. To fulfil the main purpose, we made the decision to develop a virtual tour with those objectives:

- display 360° images in a sphere
- add immersive elements inside the virtual tour, such as
  - stereoscopic view to have a 3D effect, which is more appealing than flat view
  - sounds of JavaScript API/libraries or default HTML5 solutions
  - videos of JavaScript API/libraries or default HTML5 solutions
  - interactions, such as zooming, or " displacement" between images.
  - Mini-games can be included to add the fun effect to captivate the attention of audience.
- let the visitor of the virtual showroom control real devices in the showroom to increase the impression of presence. This part can be realised using an actual electronic board controlling relays which will switch on and off real devices. The commands will be sent thanks to request by web services (Ajax, Python and JavaScript). There is no meaning to control a device if the user can not see the result. Thus, he/she would be able to have the visual feedback, ideally in 3D, particularly if he/she is in a 3D space.

The reader is invited to refer to Chapter 5 to see how I achieved the different objectives.
1.3 My participation

The team working on the project is composed of 3 persons. The photos of the showroom were already taken by the other two persons of the team. I mainly worked on the architecture of the virtual tour with another person of the team, then by myself on the JavaScript functionalities to remotely control devices, mini-games in JS, python server setting on raspberry.

In this thesis, I will refer myself using "I" when I did the work on my own. To describe a task done collectively, I will use "we".

The details of my participation will be detailed in the section 5.
Chapter 2

Literature review

2.1 Virtual reality

The Virtual Reality is a virtual world, created by technology, where the user can be immersed. The user will perceive the imaginary universe through stimuli which affect all ours senses: sight, taste, smell, touch, and sound. The more realistic those stimuli are, the more the player will be immersed.

2.1.1 Interfaces

The user can interact with this virtual reality using different interfaces. According to Wikipedia (2015), current interfaces are:

- **workbench**: tablet graphic system, composed of one or two screens. It is focused on the object
• immersive room (spheric or cubic): composed of back-projected or stereoscopic synchronized screens disposed in a cubic or spheric pattern, where the user can be immersed. The most known system are CAVE and SAS Cube. It is rather focused on the user, unlike the workbench.

Several researchers with the University of North Carolina invented redirecting walking technique to expand artificially the size of a CAVE (Strickland, 2015).
2.1.2 Purposes and application

Virtual Reality can be used for many purposes (Fuchs Philippe, 2006):

- simulation training
- medical application (phobia treatment, relaxation, etc.)
- numerical art
- scientific visualization
- video game
- commercial fitting
- tourism and visit
Chapter 2. Literature review

- reconstitution of ruined or damaged site
- home automation and architecture

It is particularly well adapted for teaching and simulate thanks to its immersive context.

2.1.3 Hardwares

2.1.3.1 2015 situation of head-mounted-display (HMD)

In March 2015, game market of headset is composed of “powerful” engines using PC or consoles’ processor, and mobile-based devices. The former is monopolised by

- **Oculus Rift** by Oculus; presented during the Consumer Electronic Show in January 2013
- **Morpheus** by Sony; presented during the Game Developer Conference in March 2014
- **Vive** by Valve and HTC; presented during the Game Developer Conference in March 2015

![Figure 2.3: Oculus Rift at the top left, Morpheus at the right, Vive at the bottom left](image)
Chapter 2. Literature review

The later is run by **Samsung Gear VR** (Oculus and Samsung), powered by a Samsung Galaxy Note 4; **Google Cardboard** project and **Zeiss VR One**. Both of the last devices use every smartphones (between 4.7 and 5.2 inches) having a gyroscope, an accelerometer and a magnetometer. Compatible devices are enlisted in section 2.1.3.3.

![Samsung Gear VR and Google Cardboard](image)

**Figure 2.4:** Samsung Gear VR at the left and Google Cardboard at the right

The project will mainly focus on the **Google cardboard project** because the goal is to let the user bring the game to his home and Google Cardboard is the cheapest solution at present state.

During the Google I/O of the end of May, Google presents a new version of Cardboard. I will not work on this version since the blueprint of the Cardboard is not released yet.

My supervisors has brought back C1-GLASS (ref: 2.5) by Goggle Tech from **Augmented World Expo** at Santa Clara. Those glasses are quite useful for developers because unlike Google Cardboard, the developer can see what the experimenter actually see. Besides it can be easily used by people with glasses.
2.1.3.2 Oculus Rift

The Oculus Rift is a wearable virtual reality headset, which lets the user have a stereoscopic 3D view. It embeds a gyroscope, an accelerometer and a magnetometer.

2.1.3.3 Google Cardboard

Google Cardboard is a project which leans on the principle of the Virtual reality experience in a simple, fun, and inexpensive way (Google, 2015b).
Equipement

A google cardboard is composed of:

- Google cardboard template
- two lenses
- magnets. They are used to ”click” on mobile. When the user moves the magnet, the smartphone’s magnetometer will detect a variation and the application will interprets that variation as a click.

It can be completed with a NFC (Near field communication) chip.

Compatible devices

Normally, Google Cardboard is compatible with any smartphone equipped with a gyroscope (to recognize the head’s movement) and a a magnetometer (magnet clicking). Smartphones, which simulate the behaviour of a gyroscope based on the accelerometer data, do not work with Cardboard’s application. According to ZURICH (2015) website, currently the compatible devices are:

- Apple iPhone 4, 4S, 5, 5C + 5S
- Google/LG Nexus 4 + 5
- HTC Eco 3D, One (Mini, S, X, X+), Sensation, Sensation XE, Velocity 4G
- Huawei Ascend G 615 + P1
- iOcean X7
- LG G2, Optimus 3D Max (P720), Optimus 4X HD (P880), Optimus G (E975), Optimus G Pro + P940 Prada 3
- Samsung Aktiv S, Galaxy (Beam, S2, S3, S3 Mini), S4 (Active, Mini)
- Sony Xperia S, SP, T + Z1
- Wiko Highway
Analysis of mobile devices sensors

The recent smartphones are equipped with sensors. Those sensors may help the user connect to the virtual reality:

- accelerometer
- gyroscope
- magnetometer
- gravity sensor
- linear acceleration
- rotation sensor
- ambient temperature
- proximity sensor
- light sensor
- pressure sensor
- relative humidity.

Most of the smartphones are equipped with accelerometer and magnetometer.

2.2 Existing softwares for virtual reality

2.2.1 Oculus SDK

The Oculus company provides numerous SDK (Software Development Kit) to the developers, including for Unity 4. A complete documentation is available on Oculus VR website.
2.2.2 Google cardboard SDK

Google company provides two SDK: one for Android and the other one for Unity. It contains examples to get started quickly. The documentation is available on Google developers website, cardboard section.

2.2.3 Web VR solution

There are different libraries to reconstitute 360° panoramic images. Most of them are still not stable versions and are under development. The main point in our project was not just to reconstitute the image, but also to "move" virtually inside the showroom.

We were interested in a library allowing us to "click" without any physical interaction (touch or "magnet click" -cf: 2.1.3.3). The reasons are:

- the magnet does not work on every smartphone. They may not have a magnetometer which is indispensable for the magnet click to work or their magnetometer is too far from the magnet (due to the architecture of the cardboard).

- Google Cardboards are more expensive with a magnet, thus the distribution costs would be a lot higher for the company.

Ideally, we also wanted for API which provides stereoscopic view and hotspot to "move" between different panoramic images. We found KrPano and PanoTour API, which meets the requirements.

KrPano exists since 2008. It is developed by krpano Gesellschaft mbH company which counts only one employee: Klaus Reinfeld. It is the most complete API we have found and it continues to be actively and well developed. The community is quite active around the solution. According to krpano GmbH (2015b), its features are:

- Flash and HTML5
- Wide-range of supported panoramic formats and image file-formats
- Multi-Resolution
- 3D Projections
• Virtual Tours / Hotspots / Scripting

• Droplets to generate the tour (cf: image at 2.6)

• VR solution since February 14th 2015

The drawbacks of **krpano** is that there is only one person which develops the functionalities so it can be slow. Moreover, if Klaus Reinfeld decides to stop that activity, since all the source code are encrypted, there is no possibility to develop on newer version of the project.

**Panotour** is developed by **Kolor** company which exists since 2004. It partly uses **krpano** libraries and provides

• a cross-platform application

• presets of different style

• hotspot editor

• a clear user interface (cf: image at 2.7)
Chapter 2. Literature review

- standalone and offline virtual tour
- Maps, Floor Plans and Compass
- a stereoscopic/VR version with their Beta 2.5 version since July 2015.

Since it is partly based on the work of krpano, if Klaus Reinfeld decides to stop his activities, panotour would suffer as well. Nevertheless, since Kolor is quite a big company, they would surely find another solution on krpano part.

![Figure 2.7: Panotour clear user interface](image)

They are both closed-source softwares which means that they are protected by copyright and the source code can not be modified or reversed-engineered. Panotour has a partnership with krpano to use its source code.

Besides proprietary softwares, I took a look at open-source libraries, which are still unstable and then unusable for massive production.

krPano for instance is based on WebVR project (mozvr, 2015) with additional components. It is unstable JavaScript API developed by Vladimir Vukicevic (Mozilla) and Brandon Jones (Google). It lets the developer access to the VR device hosting the browser which displays the content, as long as VR properties such as Velocity, Acceleration, etc. It also provides VR attributes, such as the field of view, IPD (interpupillary distance).
Since those solutions are open-source, it may imply that what we develop will be open-source as well (copyleft license for instance).

2.3 Remote control

To remotely control devices, we need:

- a physical server to host a software server which receives requests and send electric signals to a board
- a board which can receive input signals and send orders to the relays board
- a relays board which can take input signals and ”controls” devices.

Numerous boards can handle those 3 requirements, we decided to opt for one of these existing boards instead of setting up a whole complex system.

The more well-known boards are

- Arduino (RAM: 2KB)
- Raspberry pi 1 (RAM: 512MB)
- Raspberry pi 2 (RAM: 1GB)
- BeagleBone Black (RAM: 512MB)

As detailed in Klosowski (2013) article, Arduino is rather for electronic project, Raspberry Pi is for for Complex, Multimedia, or Linux-Based Projects, BeagleBone Black is suited for Projects with External Sensors or Networking

Raspberry pi may be more suitable because of its support and active community.

Less-known boards but dedicated to remote control, such as Onion omega and Waio, are also possibilities.
2.4 Stream video feedback

The video API selection is induced by the choice of the board and the OS (Operating System) we are working on. Since we have decided to use Raspberry pi technology, we have to choose between different streaming video API:

- Motion
- mjjpg-streamer
- ffmpeg
- VLC

2.4.1 Motion

*Motion is a software that monitors the video signal from cameras. It is able to detect if a significant part of the picture has changed; in other words, it can detect motion.* [Cook (2015)](#) The software is developed in C language, and exists since 2010. It lets the user to set a lot of different configurations such as time-lapse, frequency, framerate, MySQL server setup, rotation, text, input option, etc.

2.4.2 mjjpg-streamer

mjjpg-streamer has to be downloaded from sourceforge website, then compiled on the raspberry.

*MJPG-streamer takes JPGs from Linux-UVC compatible webcams, filesystem or other input plugins and streams them as M-JPEG via HTTP to web-browsers, VLC and other software.*

2.4.3 FFMpeg

FFMpeg is a Linux package which can capture video stream.
It either streams to a some “other server”, which re-streams for it, or it can stream via UDP/TCP directly to some destination receiver, or alternatively directly to a multicast destination.

2.4.4 VLC

VLC is more well-known as a media player, but it can be also a streaming video server. It is written by VideoLAN. There is a stream option where the user can indicates a device as an input. The stream can be then displayed on a server with a protocol selected (HTTP, RTSP, etc.).
Chapter 3

Methodology

3.1 Baseline evaluation of possible technologies

Since the project takes place in Research and Development department, the goal for any subject is to develop a prototype quickly. So we evaluated and try various technologies and hardware to determine which one suits better our objectives and requirements.

Technically, I can qualitatively compare:

- hardware to host the server (remote interface and video stream) and control devices (for instance Raspberry Pi or Arduino)
- software used to get the virtual tour (krpano library/SDK cardboard/Web solutions)

I could compare software used to get a stereoscopic stream video but Lavirotte (2015) has already compared the existing APIs for stream video available on Linux.

3.2 Metrics evaluation of 3D VR

In a second phase, I want to determine if 3D VR makes people use our software longer than a flat solution and if they prefer or not the 3D vision. The Independent Variable (IV) here is the mode of display (stereoscopic or not), Dependent Variables (DV) are
Chapter 3. Methodology

the time, the reason why they stopped and user satisfaction score (1 to 5). I keep approximately constant the age of the testers.

Concerning the experimental design, I have no choice than asking independent subject. Indeed, the content will be the same so the subject under the second condition would be quickly bored because of the redundant information.

I will test for that purpose at least 30 persons: 15 in VR mode and 15 in flat mode. I will determine the time they are staying in the virtual reality and why they stopped (boring, feeling seek, etc.); and if they like the experience on a scale from 0 to 5. The questionnaire is in Appendix B, in figure B.1.

The conditions of experimental setup are:

- phone: Samsung Galaxy S5 which has a 5-inches screen.
- HDM: C1-GLASS by Goggle Tech. I used those glasses because it is easier to see what the tester see.
- I was present during the test

That questionnaire will confirm or deny

- if user experience is related to stereoscopic view
- if the time of use is related to stereoscopic view

Note: blue is dependent variable, red is independent variable

As seen in Pr Lemon (2015)’s lecture, I can use the tree decision of Andy Field (2005) to determine which method I should perform:
Thus Chi-square test should be relevant for those 2 hypothesis since my IV is always ordinal and my DVs are Nominal/Ordinal.

It will help us understand if 3D is suitable for marketing and/or presenting a project or product (which our case). Furthermore, we will know if we should add more elements in the virtual tour and which ones.

Besides, since an eventual patent is in process, I could not use my work on the “Stereolab” for my tests.
Chapter 4

Requirement analysis

4.1 Mandatory and optional requirements

4.1.1 Mandatory requirements

The mandatory requirements to fulfil the objectives (detailed in the 1.2 section) are:

- take in hand krpano and Panotour to produce a virtual tour of the showroom
- activate webVR plugins to have stereoscopic display
- learning how to use PhoneGap Cordova in order to adapt an app version for iOS and Android
- use HTML5 audio tag and implementing script in JavaScript to display sounds in the virtual tour.
- learn to develop on Rapsberry Pi and understand the functioning of web server in Python.
- how to deal with Google Cardboards’ producer and resellers

4.1.2 Optional requirements

The optional requirements are:
• adding videos

• adding interactions:
  – remotely control electric appliances (web front-end + web server on a raspberry pi)
  – stream video on the result using motion

• adding mini-game. One possible implementation is to display random logos of various SEL’s sponsors and other logos. The principle is to use eyes to hit sponsor’s logo. Once the player hits a logo, another one is displayed. He/She wins when all displayed logo are from SEL’s sponsors logos.

4.2 User requirements

The user wants to:

• immerse himself in a virtual world

• enjoy himself

4.3 Software and technology requirements

The ICAME department, where I will work, only uses cross-platform technology, such as Air, Titanium, Phonegap/Cordova, web technologies (JavaScript/HTML/CSS/Ajax/CORS) or Unity3D.

For the virtual tour development, I need to choose the quickest and most stable method to generate a tour given 360° photos between Panotour Pro, krpano library, and generic JavaScript libraries.

The problem concerning those VR libraries is that it is handle by one single person. If that person decides to stop any development, unless it is sold to another company, the software will not have any more updates.

Evaluation of software choice after actual test are available in 6.4.1 section.
About hardware, we need cheap, easy to deploy and relatively compact devices.

Concerning the stream video API, I need to select one which has the best support, the best performances, and the possibility to access it through the Internet.

We use GIT as a version control.
Chapter 5

Implementation

5.1 General organisation

5.1.1 Development versions

During the development, we use 4 main versions:

- Production: stable version
- Build: unstable version in which we write and test new functionalities
- Stereolab: unstable version in which we try the experimental functionalities for the Stereolab
- Cordova: stable version adapted for app built, imported from production version.

The work flow of source code exportation is Build → Production → Cordova. In parallel, we work on Stereolab version that we will export into production when the whole hardware part will be set up in the actual showroom in Lyon.

5.1.2 Files and folders

The source code of the software is hosted in the following folders:

- **root folder** contains the .html files of the different versions of development
• js contains the .js script that we developed for the additional functionalities that is not supported by krPano, and general libraries, such as JQuery

• krpano contains all the libraries that are provided by krPano API, and configurations (.xml files) for the different plugins (webvr, gyro, etc.)

• panos contains the 360° photos in different qualities (desktop/mobile/tablet)

• ressources contains the different resources of the project, such as images for the hotspots, texts, sounds, and .psd files for the graphics sources, etc.)

• res (only in ”cordova-build” folder) contains solely the resources for the app version: icons and splash screens for iOS and Android.

• xml contains all the .xml files defining the virtual tour and functions. It contains the 3 versions of the XML and common folder containing the definition of common hotspots. In each version, there are those files and folders:
  – i18n contains the different versions of text translations
  – panos contains all the information of the different ”panos” (inclusion, hotspot, transition, etc.)
  – skin contains skin change we did for the layer menu
  – threejs contains source code of threejs library and objects
  – actions.xml contains the general actions generated by Panotour
  – panos.xml contains the inclusions of all .xml files in panos folder
  – SEL.actions.xml contains the functions we wrote
  – welcome_page.xml contains the menu at the beginning which is display

5.1.3 Virtual Tour

The main component of the project, the virtual tour, is composed of different photos in 360°, which are called panos. Thanks to panotour, they are converted into various quality of images. The user agent of the browser (devices attribute) determines the quality of medias: mobile devices will get the lowest quality whereas desktop will have the best quality.

The panos are linked together thanks to hotspots.
There are two ways to generate the first version of the tour: **Panotour** by Kolor, a software with graphical user interface, or **krPano** by Klaus Reinfeld, which provides a droplet to generate the tour given 360° photos. They are both based on an xml ”language” that is interpreted by JavaScript libraries.

With both methods, the generated tour is composed of .html, xml and JavaScript files. The tour generated by **krPano** is much cleaner than the one generated by **Panotour** but less complete in terms of functionalities. I mixed the two versions generated to have a complete and clean tour. Both of them provides a VR mode that is to say a stereoscopic vision of the tour so that a user, equipped with **Google Cardboard** or an equivalent, can see the tour in 3D.
Chapter 5. Implementation

It was better to work with krPano due to its complete documentation rather than Panotour. This software quickly shows the limit of the functionalities since it is defined by what is graphically implemented and does not provide any documentation for the source code.

5.1.3.1 XML Structure

This part is a quick explanation of the XML structure in order to help you understand the functionalities that I developed.

The root element of XML files are krpano DOM object. Various subtrees can inherit from that root element.

In term of representation, zorder attribute represents the superposition order of the elements, while depth represents the "distance feeling".

Scene tag

Every images are determined by a "scene" tag. The source of the image is defined by "thumburl" attribute. Inside scene tag, subtrees tag can be defined, in particular the view, hotspot, layer, action, image.

Functions

Functions are defined in action tag, and their names are defined by the name attribute as following:

1. `<action name="name_of_the_function">`
2. `</action>`

To call them, a reference to their name is enough. Parameters can be passed by adding parenthesis "name_of_the_function(parameter1, parameter2, ...)". Inside the function, parameters can be accessed using percentage symbol (%) following by the number of the parameter starting at 1 (0 is the name of the function).
Chapter 5. Implementation

Hotspots

The **hotspots** are defined using *hotspot* tag and can have various attributes, such as:

- *ath*, the x coordinate
- *atv*, the y coordinate
- **style** - optional, attribute which takes the name of *style* tag
- *url*, the URL of the image for the hotspot
- *crop* - optional, the crop of the image if it is a sprite. The value is defined as following: "x-position|y-position|width|height".
- *distorted* - optional, option to activate in 3D space. Mandatory to be visible for VR version.

Since 3D text in virtual reality are not well integrated because of the depth problem, we have used image hotspots to display them.

Layers

Layer are structures which are on the top of what it is represented and cannot be selected in VR mode. They are defined as following:

```
1 <layer name="name_of_layer">
2 </layer>
```

Furthermore, they can have the following attributes:

- *width*
- *height*
- *bgcolor*
- *bgalpha*

They can also have child layers which will be represented on the top of the layer.
5.1.3.2 JavaScript and krpano

To use JavaScript directly in krpano, I have to include .js files from .html files and call a JS function using \( \text{get(name\_of\_function)} \).

To pass JavaScript variable to .xml files, in .js file, I have to access the \text{krpanoSWFObject} element using \text{getElementById()} function:

```javascript
var krpano = document.getElementById("krpanoSWFObject");
```

Afterwards, I can use \text{set()} function to name a variable and set its content:

```javascript
krpano.set("name\_of\_variable", "content");
```

Finally I can use that variable from a xml file using the \text{get()} function.

5.2 Functionalities

I will detailed in this part the functionalities that I developed or generated. Other functionalities developed by my colleagues or existing functions from libraries will not be explained because it is irrelevant in this context, unless explicit reference.

5.2.1 Virtual tour

5.2.1.1 Architecture / first version

To generate the raw virtual tour, I used \text{Panotour Pro}, I imported the 360° images into the software. Afterwards, I added hotspots in every image, then generated the very first version of the tour.

The main point of that version is to have the global architecture of a virtual tour using \text{krpano/pantour}.

5.2.1.2 Title in every image

I have added a title hotspot inside every scene tag:
All the texts are stored in a sprite image: texts-panotitle.png.

The style TextPanoTitle is defined in the file hotspot_style.xml in common folder:

```xml
<style name="TextPanoTitle"
    url="%HTMLPATH%/ressources/texts/texts-panotitle.png"
    enabled="false"
    capture="false"
    distorted="true"
    scale="1"
    ath="0" atv="10"
    zorder="100"
    zorder2="1"
    edge="center"
/>
```

By default, the hotspot is not visible. I created a SEL_Show_Title_Pano() action, which will make the title visible, and then hide it after 4 seconds:

```javascript
set(hotspot[text_TITLE_PANO].atv, 90);
set(hotspot[text_TITLE_PANO].alpha, 0.0);
set(hotspot[text_TITLE_PANO].visible, true);

callwith(hotspot[text_TITLE_PANO],
    tween(alpha|atv, 1|20, 2, easeOutQuad|easeOutQuint);)

delayedcall(4, SEL_Hide_Title_Pano());
```

The previous function is called by onnewpano() action, which is called at every generation of a pano. The SEL_Hide_Title_Pano() action will move the hotspot vertically until it is not visible by the user and set it to not visible.

```xml
<action name="SEL_Hide_Title_Pano" >
callwith(hotspot[text_TITLE_PANO],
    tween(alpha|atv, 0|-90, 2, easeOutQuad|easeOutQuint,)
```
5.2.1.3 Triggers hotspots

By clicking on the "more info" hotspot, it leads to a particular pano containing visual trigger:

![Image](image_url)

**Figure 5.2:** More information hotspot leading to the map on the left, and to the wall on the right

Those visual triggers (cf: figure 5.3) pops up a record (cf: figure 5.4)

I defined the triggers as following:

```xml
<hotspot name="temoignage-residentiel-03"
  category="residentiel"
  description="djoumer"
  fiche="fiche-temoignage-residentiel-03"
  style="temoignage"
  ath="-31.858"
  atv="4.456"
  url="%HTMLPATH%/ressources/spots/temoignages/3-02.png"
  onclick="SEL_ShowTemoignage_popup('ressources/sounds/SEL_Expe_Keltoum_Djoumer_H264_PAD.mp3')"/>
```

On the click of the trigger, it will call *SEL_ShowTemoignage_popup()* action thanks to the *temoignage* style:

```xml
<style name="temoignage"
  distorted="true"
  enabled="true"
  capture="true"
```
I defined that action as following:

```xml
<action name="SEL_ShowTemoignage_popup">
  hide_all_fiches();
  show_all_temoignages();
  js(playMySound('%1'));
  set(hotspot[hotspot_sound_load].alpha,1);

  set(hotspot[get(name)].visible, false);
  set( fiche , "fiche-" );
  txtadd( fiche , get( name ) );
  set(h_fiche, get(ath));
  set(v_fiche, get(atv));
  set(hotspot[get( fiche )].ath, get( h_fiche ));
  set(hotspot[get( fiche )].atv, get( v_fiche ));
  set(hotspot[get(fiche)].visible, true);

  <!-- Show hotspot sound load pano -->
  set(hotspot[hotspot_sound_load].ath, get(ath) );
  set(hotspot[hotspot_sound_load].atv, get(atv) );
  set(hotspot[hotspot_sound_load].visible, true);

  <!-- hide hotspot sound load pano -->
  delayedcall(3,
    callwith(hotspot[hotspot_sound_load],
      tween(alpha, 0,
        set(hotspot[hotspot_sound_load].visible, false)));
  );

  set( play , get(fiche));
  txtadd( play , "-play" );
  set(h_play, get(ath));
</action>
```
Basically, it

1. hides all record if the user clicks on another trigger without closing the records

2. shows all the triggers.

3. plays the sounds sent in first parameters (cf: 5.2.3 section)

4. hides the call trigger

5. shows the record associate to it (same name but with "fiche-" before it)

6. temporarily shows a sound icon to indicate that a sound is played

7. shows "play" icon to let the user pause and resume the sound (same name but with ",play" after it)

8. shows "quit" icon to let the user quit the record and stop the sound (same name but with ",quit" after it)
If the visitor quits the pano, it will stop all sounds.

5.2.1.4 Animation of triggers hotspots

After a few tests, we were told that the sound hotspots and map hotspot (visual triggers) are not explicitly interactive (cf: figure 5.3).

That is why, we decided to animate them on the loading of the pano. For that purpose, in SEL_actions.XML file, I created a SEL_AnimateTrigger() function, which will get the default atv attribute, then set the hotspot’s atv to a high number, then progressively transfer it from that number to its default atv.

```xml
<action name="SEL_AnimateTrigger">
  set (hotspot[get(name)].scale,1);
  set (tmp_atv, get(atv));
  set (hotspot[get(name)].atv,-85);
  callwith(hotspot[get(name)],
            tween(atv, get(tmp_atv), 2 );
            tween(scale, 0.1, 2 );
  );
</action>
```
Then I added an `onloaded` attribute to every trigger hotspots. That attribute will call `SEL_AnimateTrigger()` function:

```xml
<style name="temoignage"
  distorted="true"
  enabled="true"
  capture="true"
  scale="0.1"
  alpha="1.0"
  depth="600"
  zorder="2"
  onover="tween(depth|zorder,300|3);"
  onout="tween(depth|zorder,600|0);"
  onloaded="SEL_AnimateTrigger"
/>
```

### 5.2.1.5 Testers’ quotes: verbatims

The verbatims are testers’ quotes. I defined them as following:

```xml
<hotspot name="verbatims-residentiel-01"
  category="residentiel"
  style="verbatim"
  ath="20.788"
  atv="10.054"
  url="%HTMLPATH%/ressources/spots/verbatims/v2-residentiel1.png"
/>
```

Since the visitors could not clearly read them. To correct that problem, I let them zoom in on the verbatims. The main issue was to not "click" on it in stereoscopic/VR mode since the `onover` attribute does not make any sense in this mode. In order to avoid that, I set the `handcursor` attribute to "false".
5.2.2 Stereoscopy

krPano provides a stereoscopic functionality to the virtual tour generated. The library code is stored in `webvr.js`, and configuration is defined in `webvr.xml`. A default `webvr.xml` is provided when a tour is generated.

`webvr.js` is encrypted, and is based on WebVR Browser API.

![Figure 5.5: Same image from a and non stereoscopic view](image)

In order to add the stereoscopic display into our virtual tour, I generated a tour as well with krpano droplets. I imported the WebVR.js libraries, the skin and other configurations into our project, made the correct inclusions.

After the first demonstration in front of Paris town hall, we could have determine what we could improve. The complexity of the menu is one approach. Indeed, screensize and IPD (Interpupillary distance) were unknown for the majority of the users and for the hosts. So we wanted a simpler menu. Since krPano provides only a complete menu, while Panotour provides also simple one, we decide to combine both.

![Figure 5.6: VR Menu: simple one on the left and advanced one on the right](image)
I will refer the stereoscopic functionality (dual screen) as the VR or WebVR version/option/mode later on.

5.2.3 Sound records

In the actual showroom, the visitor can listen to the records of testers. We decided to integrate this functionality into the virtual showroom.

I used the software audacity to extract the useful track from the video they have sent to us.

I firstly used the sound interface plugin of krPano. It works well in app version since it is pre-loaded, but in VR web version, it did not load the sound unless a touch is performed on the screen.

In order to avoid that problem, I defined a sound using HTML:

```html
  <audio id="sonTemoignage" preload="auto">
    <source src="ressources/sounds/rebond.mp3" />
  </audio>
```

The click on a portrait will call the function SEL_ShowTemoignage_popup(), which calls the playMySound() function. It will replace the url source of the blank sound by the parameter.

```javascript
  function playMySound(src){
    document.getElementById("sonTemoignage").src = src;
    document.getElementById("sonTemoignage").play();
  }
```

Additional animation is animation at the load of the panos is set to make people understand that portraits are "clickable".

User will zoom in verbatims (testers' quotes) on hover of the mouse or reticle. It can be done by setting handcursor attribute of the verbatims' hotspot to false and changing the scale to a bigger one. The change of zorder attribute makes the verbatim to the foreground.
Figure 5.7: Portraits in red are clickable, verbatims in green can be zoomed in.

5.2.4 Game

For Images wall, 9 screens are available, so we wanted to create a game with 9 graphic elements.
One idea of the game is to click on logo that does not belong to Smart Electric Lyon’s partners by clicking (in desktop mode) or targeting (in VR mode) them.

I create for that purpose 9 hotspots that superpose to the screens called "logo_game<no >". Each of them calls check_tile action and have a good attribute. By default, those 9 hotspots are hidden: their visible attribute are set to false.

**Start game**

On the click on the "Play" hotspot, it will call start_game action. That action

1. hides the "Play" hotspot
2. shows the rules
3. after five seconds, it hides the rules, shows the stop hotspots, called create_all_tiles() functions. That function, for every tiles will

   (a) call the randomGood() JavaScript function to get a true or false with a chance of 1/4 to have a true and store the result in the variable ” return_good” of krPano DOM Object:
function randomGood(){
    var krpano = document.getElementById("krpanoSWFObject");
    var test = Math.floor(Math.random() * 3);
    if(test>=1){
        krpano.set("return_good", "true");
    }else{
        krpano.set("return_good", "false");
    }
}

(b) given the result, it will set the good attribute to either true either false

(c) given the result, I determined a pseudo random crop (getCrop() function) for a real partner logo, or a specific crop ("0|20|25|24|0|135") which corresponds to the coordinates and size of the false logo.

(d) finally it set the visible attribute of the tile to true

getcrop() function returns a correct format the crop attribute of a hotspot, that is to say: "x|y|dx|dy", with x/y the "coordinates" and dx/dy the width and height of the image. The function calls randomXCrop() and randomYCrop() which return a multiple of the width and height respectively of a logo. Since I have 15 rows of logo and two columns of the global image of logo.

The result:

Figure 5.9: Logos tiles and stop hotspot are displayed
I am temporarily using a false logo because legally I cannot display an enterprise’s logo without their consent.

**Check of answer and win condition**

We wanted a visual element to indicate if the logo is good or not such as:

Thus I have created 18 hotspots, that superpose to the tiles, called "check<number>" and "uncheck<number>" whose visible attribute is set to false by default.

On the click or "target" of a logo, it will call the check_tile() function which

1. determines if the hotspot which calls the function is good or a false logo
2. given that information, sets the visibility (alpha) of "check" or "uncheck" hotspot to true
3. with a delay of 1 second set the opacity (alpha) of "check" or "uncheck" hotspot and the logo to 0.
4. 1.6 second later, I will "create" a new tile by changing the cropping of the caller hotspot to another one, using the create_tile() function. That function calls the win_condition() function which count the number of hotspots whose good attribute is set to true. If that count is bigger than the (number of tile - 1), that is to say 8, it will display the win message hotspot and call the stop_game() function.

**Stop game**

If the player want to stop the game, he can just click or "target" the stop hotspot. It will call the stop_game function which:
1. shows ”start” hotspot

2. hides ”stop” and the ”win text” hotspots.

3. calls `hide_all_tiles()` function which sets the visible attribute to false.

**Bug correction**

I started to create a different .png file for every logo and change the url to load a new logo, but it bugged a lot. So I decided to work with crop. I assumed that the loading of a new image every time is more resource consuming than just changing the crop. It bugs less, but there is still some errors at the generation of a new tile. So I decreased the size of the image and it bugs less after that manipulation.

**5.2.5 Remote control: Stereolab control**

To increase the feeling of immersion, we wanted to let the user control devices in the showroom through virtual showroom and have video feedback of the result.

That system is composed of 3 modules:

- Physical control and web server
- Virtual control through the virtual showroom
- Video feedback and video control

We decided for that part to use a Raspberry Pi for this part. Further explanation of the choice is available in 6.4.2 section. We choose a **Raspberry pi 2** to improve the experience for the video stream (cf: 6.4.3 section).
Figure 5.10: Stereolab is composed of 3 modules

5.2.5.1 General connection

The piFace has to be put on the top of the Raspberry Pi. It directly connects its outputs/inputs to the Raspberry Pi’s pins. The output pins 0 and 1 of the piFace are already connected to the relays which are already on the piFace, that is why we used the pins number 4 to 7. The outputs are connected to the relays’ inputs so that they can be
control from the **Raspberry pi**. The switches are connected to the **piFace**’s input so that they can be read.

The electric device are connected to relays’ outputs.

Webcams are connected to the USB ports of the **raspberry pi**.

![Electronic connection for Stereolab](image)

**Figure 5.11:** Electronic connection for Stereolab

### 5.2.5.2 Raspberry Pi setup

To use the Raspberry Pi, we first need to install an OS. Recommended OS for that kind of device is **Raspbian**, a linux distribution especially adapted to Raspberry Pi.

After downloading it, we could install it on the SD card using `dd` command line for OS X and Linux user; or **Win 32 Disk Manager** on Windows.

After setting up the system, we have to install the **piFace** interface and **motion**.

When Raspberry Pi is shutdown unexpectedly, the SD card containing the OS can be corrupted. In this case, there is no solution to get back the data from a corrupted SD
card for now. Thus it is better to clone the OS of the SD card using `dd` command line or Win 32 Disk Manager to create an image clone. If anything happens to the SD card, we could burn a new copy thanks to that image.

To avoid any unexpected shutdown and so a SD card corruption, I need to use:

```python
sudo halt
```

to shutdown correctly.

### 5.2.5.3 Physical control

To physically control the device, we connected 4 switches to piFace’s inputs. Each switch has 3 pins: VCC (IC power-supply pin), ground (0V), and a led which toggle on the push. We added a resistance otherwise, there is still a bit electric tension which passes and the led was on even if the pin was not activated. The LED pin is connected to the output of piFace so that there is no additional software control needed.

I wrote a python script using piFace’s documentation to toggle the device given the physical buttons.

To develop on piFace interface, I need to import pifacedigitalio library, then create a PiFace Digital object. I can access directly the different elements of the board from this object:

```python
import pifacedigitalio

pfd = pifacedigitalio.PiFaceDigital()

def toggle_led4(event):
    event.chip.leds[4].toggle()

def toggle_led5(event):
    event.chip.leds[5].toggle()

def toggle_led6(event):
    event.chip.leds[6].toggle()

def toggle_led7(event):
    event.chip.leds[7].toggle()

listener = pifacedigitalio.InputEventListener(chip=pfd)
listener.register(4, pifacedigitalio.IODIR_FALLING_EDGE, toggle_led4)
```
To let the user control devices thanks to physical buttons, I add Event listener to the board’s inputs.

The script then can be run using python command:

```python
python3 /usr/stereolab_script/inputControl.py
```

You can see the 4 physical buttons on the following 3D model:

![3D model of Stereolab Box](image)

**Figure 5.12: Front of Stereolab Box, realised with SketchUp**

### 5.2.5.4 Virtual Control

The web server, used to control the devices at the distance, is from the example of piFace’s libraries: `simplewebcontrol.py`. The server is written in Python.

I added a specific `header` in order to make the request from the virtual tour using Ajax:

```python
self.send_header("Access-Control-Allow-Origin", "*");
```

Then the outputs of the board can be controlled from a browser using a URL request. The request is in hexadecimal format. The use of the server is detailed on piFace.org (2013)’s website:
You can set the output port of PiFace Digital (the LEDs) by adding the following line to the end of the address: /?output_port=0xaa So the full command will be: http://192.168.1.3:8000/?output_port=0xaa 0xaa is a representation of the output port values in hexadecimal. The PiFace Digital sees this in the binary form 10101010 with a 1 representing an LED on, and a 0 representing an LED off. 0xff would have all the LEDs turned on, and 0x00 would have them all off.

During the virtual tour visit, the user can trigger various devices:

The click on a hotspot will trigger a JavaScript function which will use Ajax to make a request to the web server on the Raspberry PI.

To let the user toggle device, I use a XOR (Exclusive OR) between the old output (pseudo json format) and new output (hexadecimal format).
5.2.6 Video feedback

In order to let the user see what he can do by triggering the switches, we set up a video module which is integrated in the virtual tour.

To realise that, there are a lot of different APIs(cf: Video feedback p.16). Since Motion provides a movement detection and a light server to visualise the stream video, we decided to choose that API.

Configuration files are in the following path:/etc/motion/. Compared to the default configuration, we changed:

- 2 threads (called thread1.conf and thread2.conf) to have 2 stream video: we have created one .conf file per webcam in addition of the general configuration file: motion.conf.

- rotation: physically, we got a problem during the building of the Stereolab and had to mounted a webcam on the reverse. That is why, I had to change the rotation of one thread/webcam of 180°.

- starting the server at boot of the raspberry pi

- decrease the framerate on raspberry pi 1.

- not saving the video

To access the raspberry pi, on Mac, I use directly the Mac console, and on Windows I use putty.exe using SSH protocol.

Locally, to access more easily the raspberry, we use the name of the device instead of its IP address thanks to Bonjour software. Bonjour software is an implementation of zeroconf which lets the user to access a host given its name rather than its IP address. A " .local" is needed after the name of the device.

The result of the stereoscopic video stream is:
5.3 Deliverables versions

5.3.1 Web version

This version is a complete application with the best quality of images. The visitor can zoom in (using the wheel of the mouse or using 2-fingers touch). It can be accessed via web on a mobile phone. Mobile phone user can use a Google Cardboard to have a stereoscopic view. An internet access is needed to visit the virtual showroom.

5.3.2 App versions

This version is a light version of the desktop version, there is no possibility to zoom in, and can only be installed on iOS or Android. This version does not need any internet access. The app is available on the App Store and Play Store.

To develop an app which is equivalent to the web software, there are multiple solutions:
• develop native apps for each mobile platform we want to exploit

• use Phonegap build

• use Apache Cordova

• use Appcelerator Titanium

Since redeveloping native apps from scratch requires more time and skills (Java Android SDK and Objective C) than just exporting the software using cross-platform application, we decided to focus on a cross-platform software. Since our department have already worked with phonegap/cordova, we decided to choose that technology.

Phonegap project split into two kinds of ”softwares”:

• Phonegap Build acquired by Adobe, it is an online tools which takes a .zip file containing the web source to generate compressed build files for each platform, such as .apk for Android or .api files for iOS.

• Apache Cordova, it is a command line tools, which can be install through npm interface. The SDK for each platform has to be installed before the build.

Since the Phonegap Build generation failed for our project, we decided to use Apache Cordova.

5.3.2.1 Apache Cordova

The apps are generated by Apache Cordova. We have focused on two mobile OS: iOS and Android.

For Android, I could directly generated the apk file using Cordova command line.

For iOS, the build of Cordova generates a xCode project. The debug can be directly launched from xCode. Apple provides a lot of different licences to their developers.

The difficult part is to adapt functionalities which are not exported well, or bugs which are not in desktop version.
5.3.2.2 Publication on the store

For Android and iOS version, the method to publish the app their respective store is different.

To publish it firstly on Google Play, I needed to generate a self-signed certificate using keytool, in Java tools. Then the code had to be aligned to decrease the size of the app using zipalign. Afterwards, I had to sign it using jarsigner. Finally, I could publish the app through Android developer interface on the web with EDF Android licence. The app will be publish after validation in a few hours following the upload of the apk.

To publish the app on the App Store, a distribution licence is needed. After setting all the licence in xCode, an archive can be generated and it will be upload and published through iTunes connect interface. The app will be publish after validation in a few weeks following the upload of the apk.

5.3.2.3 Import from desktop version to App version

When the app is too big, it often crashes. Thus for app, we removed the images for the desktop and only keep the mobile version. In the code, we need to remove the reference to the big images. To avoid the manual deletion at every import, I wrote a python script to only keep what we need when we define the images. Since it is XML structure, I used xml.etree.ElementTree library. (For complete code, refer to Appendix: C.1).

I firstly removed the image tag which has a level tag as subtree, and remove the attribute devices in the rest of image tag to generalise those images to every devices.

I could add those functionalities in native code in Objective-C for iOS and Java for Android:

- keeping the screen on
- keeping the orientation

5.4 Additional work

I have worked on Adobe Photoshop as well for the resources (hotspots, texts, etc.).
A non technical work that I had to do was to find a good supplier of **Google Cardboard** for the demonstration on the forecourt of Paris’ town hall on July 1st. Given the short notice time, we decided to ask Chinese people via Alibaba interface. I negotiated in Chinese to make sure that they have exactly what we needed.
Chapter 6

Results and reflections

6.1 Deliverables

The deliverables of the project are:

- first deliverables of the project. It was presented during Paris town hall event on July, 7th. This version is without the remote control of device and contains the virtual tour with images, displacement and sounds. The main concern for that version was to have an application which does not need any internet connection since the demonstration will take place outside. An iOS app was delivered on time. Cardboards were distributed along with stickers whose bit.ly address refers to the project’s page. The sticker design was done by a graphic designer working for SEL:
Chapter 6. Results and reflections

Figure 6.1: The sticker refers to the software website

The project’s page is on the Joomla based SEL’s website. At that time, it redirects towards the web-based software (cf: next point). After the demo, we have been reported that it was freezing randomly in stereoscopic/VR display. The problem was due to the webvr.js the library of krpano. It did not freeze on Android but since SEL only has iPhone 6, it was freezing unfortunately.

- the application as a web-based software: http://sandbox.e76.fr/vrsel/. This website displays the most advanced validated version. It has the best quality type of images. The user can zoom in without any quality lost. The drawbacks of that version are that the user need internet connection and the screen switch off after a while.

- the application as an ios app: https://itunes.apple.com/fr/app/showroom-vr/id1001163670?l=fr&ls=1&mt=8
  The advantages of that version are that it does not need any internet connection and the screen can be kept on. The drawbacks of that version are the random reboot, the lowest quality of images and long validation at every updates (about two to three weeks).
• the application as an Android app:
  The advantages of that version are that it does not need any internet connection,
  the screen can be kept on, there is no random reboot compared to the iOS version
  and quick validation of updates (about 3 days). The drawbacks of that version is
  the lowest quality of images.

• source code of the application to ICAME department.

• master thesis dissertation (covering user documentation, developer documentation,
  with project evaluation) to Heriot-Watt University.

• poster presenting the project

• Additionally to the website, iOS app, Android App available to be evaluated, a
demo will be performed in front of assessors who has signed the NDA.

6.2 Stakeholders and their needs

The direct stakeholders are:

• the customers and industrial partners

• the developers

• persons in charge of SEL project

• the supervisors of the developer (school supervisor and company supervisor)

• the managers of the developers

Indirect stakeholders relatives to softwares and hardwares used:

• Google Cardboard

• Homido

Additional stakeholders can be considered since it is a MSc. project at Heriot-Watt
University, i.e.:
• assessors of the project

• MACS department

• Heriot-Watt administration more generally.

Placement-based master project implies other additional stakeholders:

• ICAME department

• EDF Company

• SEL council
### 6.3 Project Schedule: Gantt Chart

<table>
<thead>
<tr>
<th>Month</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>August</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Start of the project**
- Virtual tour architecture
- Web VR functionalities
- Validation of version 1
- JS control devices+Server settings on RPy(v1)
- PhoneGap cordova export
- Git setup+debug+insomnia
- Sound
- Sponsors mini-game
- Paris town hall deliverable
- JS control devices+Server settings on RPy2(v2)
- Integration into the tour+video stream(v3)
- Mini-game development
- Debug/test/improvements
- Report writing
- Report deadline 20-08
- Poster preparation
- Poster presentation: 27-08
6.4 Qualitative evaluation of possible technologies

This part will explain the different tests I have done to determine why I used a given technologies for the project. The results are based on the actual test I did and the research I have done for the literature review.

6.4.1 Software for virtual tour

To generate the virtual tour, we had the choice between different libraries and API. We decided to use krpano because we can add functionalities, and use already implemented functionalities, and add 3D elements.

The drawbacks are that we depends on krpano development advancement. For instance, we have to wait for the new release to integrate video in 3D virtual tour. Furthermore since its libraries are encrypted, we can not change anything nor debug problems. For example, iOS reboot bug is induced by the webvr.js libraries.

6.4.2 Electronic board

As explained in 2.3 section, it was more appropriate to use a Raspberry pi rather than a Arduino or BeagleBone Black.

We still try firstly a dedicated to remote control solution: a Waio board for a quick first prototype. Since we could not make web request to the server, we decided to change technologies.

The team has a strong experience in raspberry pi. Furthermore that board has an active community and a good support. Lots of libraries have been developed for that board. For all those reasons, we decided to choose that solution.

In the following section, I will describe why we used a Raspberry Pi 2 instead of the first one.
6.4.3 Camera control

According to Lavirotte (2015), who had compared different technologies in video streaming, the solution ”ffmpeg, crtmpserver and lighttpd” seems to be the less resourceful.

Nonetheless, it lacks support and there is no easy way to set 2 inputs/outputs for stereoscopic vision.

That is why, we used motion, which provides very easy way to set up 2 or more webcams/cameras, we just need to add as many threads as camera/webcams.

At 20 framerate per second, by using top command, we got:

![Figure 6.2: Raspberry Pi 1 on the left and 2 at the right](image)

Given those printscreens, I can see that Raspberry Pi 2 uses **32.7% of CPU** and **1.5% of memory** while the 1 only uses **13.2% of CPU** but **4.1% of memory**. In theory, Raspberry 1 B+ model would be enough, unfortunately in practice, Motion

- could crash after starting it a long time on Raspberry Pi 1 despite the change of framerate, the movement detection and the non saving of the stream

- could have a long shift between what it was in input and what it is displayed.

So it was decided to use Raspberry Pi 2 rather than the 1 for the crashes and to avoid latency.
6.5 Metrics evaluation of 3D VR

6.5.1 Raw results

I carried an experience to get information concerning the software. The version used has all functionalities described in chapter 5 (p.24). Based on the questionnaire in the appendix B (B.1), I could gather the following information:
Table 6.1: Questionnaire raw results

<table>
<thead>
<tr>
<th>Time of use</th>
<th>Reason of stopping</th>
<th>Score (1-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14m</td>
<td>No more information</td>
<td>2</td>
</tr>
<tr>
<td>10m</td>
<td>No more information</td>
<td>1</td>
</tr>
<tr>
<td>12m</td>
<td>No more information</td>
<td>1</td>
</tr>
<tr>
<td>8m</td>
<td>No more information</td>
<td>2</td>
</tr>
<tr>
<td>10m</td>
<td>Not enough interaction</td>
<td>3</td>
</tr>
<tr>
<td>11m</td>
<td>Not enough interaction</td>
<td>2</td>
</tr>
<tr>
<td>13m</td>
<td>No more information</td>
<td>4</td>
</tr>
<tr>
<td>6m</td>
<td>Not enough interaction</td>
<td>2</td>
</tr>
<tr>
<td>8m</td>
<td>Not enough interaction</td>
<td>3</td>
</tr>
<tr>
<td>5m</td>
<td>Not enough interaction</td>
<td>3</td>
</tr>
<tr>
<td>15m</td>
<td>No more information</td>
<td>2</td>
</tr>
<tr>
<td>12m</td>
<td>No more information</td>
<td>3</td>
</tr>
<tr>
<td>13m</td>
<td>No more information</td>
<td>2</td>
</tr>
<tr>
<td>10m</td>
<td>No more information</td>
<td>4</td>
</tr>
<tr>
<td>11m</td>
<td>Not enough interaction</td>
<td>4</td>
</tr>
</tbody>
</table>

NON stereoscopic mode

<table>
<thead>
<tr>
<th>Time of use</th>
<th>Reason of stopping</th>
<th>Score (1-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8m</td>
<td>motion sickness</td>
<td>4</td>
</tr>
<tr>
<td>5m</td>
<td>motion sickness</td>
<td>3</td>
</tr>
<tr>
<td>3m</td>
<td>tiredness (eye accommodation)</td>
<td>3</td>
</tr>
<tr>
<td>12m</td>
<td>No more information</td>
<td>4</td>
</tr>
<tr>
<td>6m</td>
<td>motion sickness</td>
<td>4</td>
</tr>
<tr>
<td>10m</td>
<td>No more information</td>
<td>4</td>
</tr>
<tr>
<td>6m</td>
<td>motion sickness</td>
<td>3</td>
</tr>
<tr>
<td>4m</td>
<td>motion sickness</td>
<td>3</td>
</tr>
<tr>
<td>8m</td>
<td>No more information</td>
<td>3</td>
</tr>
<tr>
<td>5m</td>
<td>motion sickness</td>
<td>1</td>
</tr>
<tr>
<td>7m</td>
<td>No more information</td>
<td>5</td>
</tr>
<tr>
<td>4m</td>
<td>motion sickness</td>
<td>3</td>
</tr>
</tbody>
</table>

Stereoscopic mode

Table 6.1: Questionnaire raw results

<table>
<thead>
<tr>
<th>Time of use</th>
<th>Reason of stopping</th>
<th>Score(1-5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>9m</td>
<td>tiredness (eye accommodation)</td>
<td>2</td>
</tr>
<tr>
<td>6m</td>
<td>tiredness (eye accommodation)</td>
<td>2</td>
</tr>
<tr>
<td>5m</td>
<td>No more information</td>
<td>5</td>
</tr>
</tbody>
</table>

Notes:

- I was present during each test, to explain global information while the subject visits

- Most of people who stopped because they said they saw everything, did not actually. Some elements are not explicitly clickable so they did not think it was an interactive area

- I rounded the time to have a whole number of minutes because the experience does not need so much precision. If the difference between the two modes is based on second scale, it means that there is no difference.

6.5.2 Hypothesis 1: User experience is related to stereoscopic view

From the raw results, I can determine the following table:

Table 6.2: Scores given the display view

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>Non VR</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td>3</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>2</td>
<td>30</td>
</tr>
</tbody>
</table>

6.5.2.1 Descriptive Statistics

The mean is the average score.
The mean score for stereoscopic mode is: 3.33

The mean score for non stereoscopic mode is: 2.53

Since 3.33 < 2.53, those means seem to confirm the hypothesis that user enjoys more the virtual reality through 3D vision.

### 6.5.2.2 Statistically significant test

As I explained in chapter 3 (p.18), I should use Chi-square test to confirm what the means seems to confirm is to say user experience is related to stereoscopic view. To answer this question, I will assume that 1 and 2 are bad experiences whereas 4 and 5 are good experiences. The table becomes:

<table>
<thead>
<tr>
<th></th>
<th>Bad experience</th>
<th>Average experience</th>
<th>Good experience</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>VR</td>
<td>3</td>
<td>5</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>Non VR</td>
<td>8</td>
<td>4</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>9</td>
<td>10</td>
<td>30</td>
</tr>
</tbody>
</table>

1. **Null hypothesis**: user experience and display mode are independent

2. **Formulation of the analysis plan**:

   - **significance level** is predetermined and is 0.05
   - the **DF (degree of freedom)** is \((number \ of \ row - 1)(number \ of \ columns - 1) = (2 - 1)(3 - 1) = 2\)

   Based on the Chi Square distribution table (Ling, 2008) for DF = 2:

<table>
<thead>
<tr>
<th>DF</th>
<th>0.5</th>
<th>0.10</th>
<th>0.05</th>
<th>0.02</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1.386</td>
<td>4.605</td>
<td>5.991</td>
<td>7.824</td>
</tr>
</tbody>
</table>

   The \(\chi^2\) value should be between 5.991 and 7.824.
3. **Analyse sample data** To estimate the number of persons who liked the experience in stereoscopic mode, I take the number of persons I tested in stereoscopic mode over the total 15/30; and the number of people of liked the experience over the total 10/30 and multiply by the total. I should expect:

\[
\frac{15}{30} \times \frac{10}{30} \times 30 = 5
\]

I have done the same calculation to the rest of cells, which gave me:

<table>
<thead>
<tr>
<th>VR</th>
<th>Average experience</th>
<th>Good experience</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5</td>
<td>4.5</td>
<td>5</td>
</tr>
<tr>
<td>5.5</td>
<td>4.5</td>
<td>5</td>
</tr>
</tbody>
</table>

**Table 6.5: Expected values for chi square**

\[
\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}
\]

\[
\chi^2 = \frac{(3 - 5.5)^2}{5.5} + \frac{(5 - 4.5)^2}{4.5} + \frac{(7 - 5)^2}{5} + \frac{(8 - 5.5)^2}{5.5} + \frac{(4 - 4.5)^2}{4.5} + \frac{(3 - 5)^2}{5}
\]

\[
= \frac{6.25}{5.5} + \frac{0.25}{4.5} + \frac{4}{5} + \frac{6.25}{5.5} + \frac{2.25}{4.5} + \frac{4}{5}
\]

\[
\chi^2 \approx 3.97
\]

4. **Interpreting results**

Based on the results in the previous part, I have \(\chi^2 < 5.991\). So it rejects the null hypothesis so I can conclude that there is a relationship between the user experience and the mode of display.

6.5.3 **Hypothesis 2: Time of use is related to stereoscopic view**

The means time with stereoscopic view = 6,53
The means time with non stereoscopic view = 10.53

It indicates that people seem to use the software longer when it is non stereoscopic mode. The reason of stopping in VR mode is generally because of motion sickness or eyes tiredness. Those both reasons could explain why people do not stay a long time in the virtual reality.

6.5.4 Testing conclusion

As a conclusion, it seems that people usually spend more time when it is a non VR mode but the experience is far less enjoyable for the user. Furthermore they generally stop because of motion sickness or eyes tiredness.

Nonetheless, those results have to be interpreted with caution for those reasons

- the sample may be too small to be representative.
- the application is not finished, in particular concerning the content. We are still waiting for the content that the communication department of SEL will send us.
  Thus, the VR functionalities are not completely exploited
- the subjects are humans and the environmental parameters influence the results. For instance, light, bad/good mood, sight acuteness, etc.

6.6 Improvements and evolution

Thanks to the multiple tests carried out in the previous section, I can determine the following areas for improvement and evolution:

- add content but without adding too much time. For instance I can use sound for stereoscopic view
- decrease time for VR mode since 66% of stereoscopic subject has eyes tiredness or motion sickness
- improve interactive elements (make people understand that those elements are interactive)
• eventually add different content to each mode. For instance more texts and video into the non stereoscopic version and solely sound in the stereoscopic mode.

• decrease the motion sickness by improving the 3D experience (image quicker or gyroscope less sensitive)
Chapter 7

Professional, Legal, Ethical and Social considerations

To be a professional developer, I have to understand the context where my project is developed.

7.1 Legal issues

For this project two kinds of legal issues need to be highlighted. One concerns the game mechanics and the other one is the license part of the program.

7.1.1 Disclaimer and instruction of use

By using a virtual reality, the player can feel motion sickness or kinetosis. That is why, it is important to warn him/her and indicate a disclaimer:

Keep your hands on both sides of the cardboard to avoid the fall of your smartphone.

Beware of not hurting yourself. Remain seated while in use. Cardboard is not for use by children without adult supervision. EDF disclaim any responsibility for any accident, injury or damage suffered.
7.1.2 Licensing

By using softwares developed by other companies, a check of licences associated to them is necessary.

7.1.2.1 krPano/Panotour

krPano and Panotour are closed-source softwares that EDF bought. It means that:

Your krpano license grants you the right to sell and distribute projects containing your licensed version of krpano. (krpano GmbH, 2015a)

7.1.2.2 Web server for remote control

For the remote control, we used directly the example provides by piFace interface. The code has been written by Thomas Preston and is hosted on github.

7.1.2.3 Motion

We can read on the Motion (2015) that it is a open-source software:

Motion is published under the GNU Public License version 2 or later

7.1.2.4 Google Cardboard

Google cardboard project is completely open-source. Cardboard blueprint are available on Google website (Google, 2015a).

7.1.3 Trademark

In the game part (p.38), I evoked the use of logos. A trade mark is under Intellectual property law and Intellectual rights.

First of all, Smart Electric Lyon corporation can display the logo of their members on their website.
Secondly, it can be a case of fair use and there is no endorsement. So according to Chmielewski & Media (2005), I can use them but it is recommended to write a disclaimer

\[\text{that identifies the owner of the logo, and that the logo is not authorized by, sponsored by, or associated with the trademark owner}\]

7.1.4 Copyright

My contract states that what I developed during my internship belongs to the company. Thus, the application developed will be fully EDF’s property. The project is under confidential protocol. The trainee has signed a non-disclosure agreement.

Besides, every assessors who will assist to the demo of the MSc, will have to sign one as well.

The remote control part coupled with stereoscopic feedback is subject to a patent deposit.

7.2 Professional issues

The professional issues deal with CRT(Competence-Responsibility-Trust) problematic. In order to get the trust of the stakeholders, I have to follow guidelines in Quality, Safety and Security (Taylor, 2015). All the work (previous work, library, etc.) needs to be properly credited to their appropriate authors.

7.2.1 British Computer Society (BCS) Code of Practice

\[\text{The British Computer Society sets the professional standards of competence, conduct and ethical practice for computing in the United Kingdom.} \quad (McQuaker, 1992)\]

I commit to develop a game according to rules established by BCS Code of Practice.
7.3 Ethical issues

In the manner of the internet uninhibited people’s way of communication because they are not in front of each other when they communicate, virtual reality may desensitize people since they unconsciously know that this is not reality. Thus the ethical issue here is similar to the controversy of violence in game.

Nevertheless, in the showroom, any kind of violent will not be represented, so the risk is negligible.

7.4 Social issues

7.4.1 Virtual reality risk

There may be two social issues concerning the virtual reality context of the game. One is a potential addiction to the game and the other one is connected to the ethical issue of the differences between reality and fiction.

7.4.2 Possible misconceptions

Since Smart Electric Lyon has not sent us the audio content to explain the showroom in our tour, the project was not completely explicit for visitor.

To avoid misconception, in the case where the software is displayed in a show, a mediator will be present to explain grey area.
Chapter 8

Conclusion

8.1 General outcomes of the project and technical issues

The main aim of presenting the real showroom has been achieved. The virtual tour part could even have been released to the general public. Globally, the following objectives of the project have been successfully undertaken:

- showroom can be visited thanks to $360^\circ$ images. The gyroscope helps the visitor to see where he/she is looking at. Furthermore stereoscopic view makes him/her see in 3D. All that part has been realised thanks to kpano library.

- there are sounds of smartgrid testers’ records. It has been realised with HTML5 solution and JavaScript to handle the source.

- the visitor of the virtual showroom can control real devices in the showroom. It has been done thanks to a python server hosted on a Raspberry Pi controlling relays which will switch on and off actual devices. The commands are sent thanks to request by web services (Ajax, Python and JavaScript).

- the visitor of the virtual showroom can interact with virtual elements. For instance, he/she moves between the different $360^\circ$ images, activates sounds, zooms on images, thanks to actions developed in kpano’s XML language.
Concerning the difficulties, it was hard at first understand to take in hand krpano’s language but the documentation, active forum, Geoffroy’s help and detailed examples helped me to overcome the first impression.

I got toned results for:

- the video feedback handle by Motion on Raspberry Pi 2. Indeed, a stereoscopic video stream can be displayed in the virtual showroom. Nonetheless, the video stream is not actually integrated into the 3D space. It is displayed in krpano as a layer on the top of the rest. This way, the user cannot quit the stream using solely the eyes: he/she has to touch the screen to close it. Furthermore, we solely tested the program in local right now. So we are not still not sure that the video stream will work via the internet.

- I have only could developed one mini-game, and it is not particularly explicit for everyone (based on tests). It works, but it is a bit too easy since the wrong logo is everytime the same logo due to copyright on the trademarks

- the contents of the showroom. Visitors are still a bit lost in the virtual showroom since we are still lacking of the content explaining the purpose of the visit and SEL project.

Finally, I could not manage to integrated videos in a 3D space. Since krpano did not implemented this functionality yet, I could display 3D video into the virtual tour.

8.2 User results

It was difficult to gather the data, especially while a part of the project is confidential information. Besides those difficulties, on one hand it was useful to actually testing the project and having users’ feedback, on the other hand, I could determine if the functionalities we implemented in the solution are actually responding the aims of the project (cf: 6.5.4).
8.3 Experience acquired during the placement

This project was a really good experience. I have learnt a lot of different technologies in virtual reality domain, along with embedded systems. Since it was a placement in a company, my choice of technologies was partly imposed and I could not have a critical point of view. Nonetheless, it was beneficial to learn those technologies because they have been selected by the company for their stability and trustworthiness based on actual experience. Furthermore, in R&D unlike in production, the technologies used are not so old.

For the not imposed part, I could use my critical point of view to determine which hardware and technologies I should use to get better results. Furthermore, I have learnt what techniques/algorithms are better for given problems, for instance crop instead of loading new images.

I also experienced actual project in a company. It was demanding but an amazing experience. I could understand that some constraints determined the future of a whole project. For instance the obligations concerning the confidential protocol. Besides, the time constraints were particularly strong since I had to finish parts for actual deadlines.

I would have liked to implement the improvement areas describe in the 6 part after the questionnaire. Furthermore, I would have also like to add more interactive and fun elements. Since the placement is not finished yet, I could work on it in the future.
Bibliography


Klosowski, T. (2013). How to pick the right electronics board for your diy project. [online], available:http://lifehacker.com/how-to-pick-the-right-electronics-board-for-your-diy-pr-742869540[accessed 16 August 2015].


Motion (2015). Motion website. [online], available:http://www.lavrsen.dk/foswiki/bin/view/Motion/DownloadFiles [accessed 7 August 2015].


Appendix A

Stereolab prototype photos

Figure A.1: Stereolab prototype
Figure A.4: Side Stereolab
Appendix B

Questionnaire

<table>
<thead>
<tr>
<th>VERSION</th>
<th>VR</th>
<th>NON VR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td>Female</td>
<td>Male</td>
</tr>
<tr>
<td>Age</td>
<td>0-15</td>
<td>16-59</td>
</tr>
<tr>
<td>Time of use</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reason of stopping</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Did you like the experience?</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>
Appendix C

Scripts

C.1 Import script from desktop to app

```python
import xml.etree.ElementTree as ET
import glob, os

os.chdir("../prod/panos")
for file in glob.glob("*.xml"):
    tree = ET.parse(file)
    root = tree.getroot()
    print(root)
    for scene in root.findall('scene'):
        for img in scene.findall('image'):
            if img.find('level'):
                print("delete image with level")
                scene.remove(img)
            else:
                if img.get('devices'):
                    del img.attrib['devices']
                    print("delete 'devices' attribute")
                else:
                    print("Nothing to delete")

    tree.write(file)
```

79