

# Masters Program in **Geospatial Technologies**



## ***BENEFITS & DRAWBACKS OF DIFFERENT MEANS OF INTERACTION FOR PLACING OBJECTS ABOVE A VIDEO FOOTAGE***

Francisco Javier Morata Chillerón

Dissertation submitted in partial fulfilment of the requirements  
for the Degree of *Master of Science in Geospatial Technologies*

***BENEFITS & DRAWBACKS OF DIFFERENT  
MEANS OF INTERACTION FOR PLACE  
OBJECTS ABOVE A VIDEO FOOTAGE***

**Dissertation supervised by:**

PhD Christian Kray

PhD Óscar Belmonte

PhD Pedro Cabral

January 2015

## ACKNOWLEDGMENTS

I would like to thank my supervisors PhD Christian Kray, PhD Óscar Belmonte and PhD Pedro Cabral for the help received to write this dissertation. I would also like to express my gratitude to Morin Ostkamp, PhD student, for his time and support.

Thanks to all my classmates for the great moments that we have spent together learning from each other. Thanks to all of our teachers, as well as Dori Apanewicz and Karsten Höwelhans for making our stay easier.

At last, but not least, I would like to acknowledge all the received from my family, thanks to all of them.

# ***BENEFITS & DRAWBACKS OF DIFFERENT MEANS OF INTERACTION FOR PLACE OBJECTS ABOVE A VIDEO FOOTAGE***

## **ABSTRACT**

Public Display Systems (PDS) increasingly have a greater presence in our cities. These systems provide information and advertising specifically tailored to audiences in spaces such as airports, train stations, and shopping centers. A large number of public displays are also being deployed for entertainment reasons.

Sometimes designing and prototyping PDS come to be a laborious, complex and a costly task. This dissertation focuses on the design and evaluation of PDS at early development phases with the aim of facilitating low-effort, rapid design and the evaluation of interactive PDS. This study focuses on the IPED Toolkit. This tool proposes the design, prototype, and evaluation of public display systems, replicating real-world scenes in the lab.

This research aims at identifying benefits and drawbacks on the use of different means to place overlays/virtual displays above a panoramic video footage, recorded at real-world locations. The means of interaction studied in this work are on the one hand the keyboard and mouse, and on the other hand the tablet with two different techniques of use.

To carry out this study, an android application has been developed whose function is to allow users to interact with the IPED Toolkit using the tablet. Additionally, the toolkit has been modified and adapted to tablets by using different web technologies. Finally the users study makes a comparison about the different means of interaction.

## KEYWORDS

Public Display Systems

Rapid Prototyping

Immersive Environment

IPED Toolkit

Video Footage

Location Based Services

Human Computer Interaction

User Study

Gesture Recognition

Usability

Mobile Devices

Server / Client Side

Web Technologies

Android

## ACRONYMS

ADT – Android Development Tools

API – Application Programming Interface

CSS – Cascading Style Sheets

DOM – Document Object Model

FTP – File Transfer Protocol

GIS – Geographic Information System

HTML – HyperText Markup Language

HTTP – HyperText Transfer Protocol

IDE – Integrated Development Environment

IP – Internet Protocol

IPED – Immersive Public Display Evaluation and Design

JDK – Java Development Kit

JRE – Java Runtime Environment

JSON – JavaScript Object Notation

LBS – Location Based Services

MVC – Model-View-Controller

OOP – Object Oriented Programming

PDS – Public Display Systems

PD – Public Display

TCP – Transmission Control Protocol

SDK – Software Development Kit

SMTP – Simple Mail Transfer Protocol

SQL – Spatial Query Language

SSL – Secure Socket Layer

SUS – System Usability Scale

UI – User Interface

UMUX – Usability Metric User Experience

URI – Uniform Resource Identifier

URL – Uniform Resource Locator

WWW – World Wide Web

XML – Extensible Markup Language

# INDEX OF THE TEXT

<b>1. INTRODUCTION</b> .....	<b>1</b>
1.1. Overview .....	1
1.2. Objectives .....	2
1.3. Dissertation organization .....	3
<b>2. Literature Review</b> .....	<b>4</b>
<b>3. Concept</b> .....	<b>7</b>
<b>4. Methodology</b> .....	<b>9</b>
<b>5 Implementation</b> .....	<b>12</b>
<b>5.1. Android Application</b> .....	<b>12</b>
5.1.2. Identifying Gestures .....	12
5.1.2. Android Application .....	16
5.1.2.2. Handling Multi-Touch Gestures .....	17
5.1.2.3. Handling Sensors for Spatial Rotations .....	20
5.1.2.4. Creating a training scenario to test the gestures implementation.....	21
5.1.2.5. Sending commands to the IPED Toolkit .....	22
<b>5.2. IPED Toolkit Integration</b> .....	<b>24</b>
5.2.1. IPED Toolkit Server .....	24
5.2.2. IPED Toolkit Front-End .....	25
5.2.3. Overlay Plug-in .....	26
5.2.4. Creating Scenarios.....	27
<b>6. User Study</b> .....	<b>29</b>
<b>6.1. Introduction</b> .....	<b>29</b>
<b>6.2. Participants</b> .....	<b>29</b>
<b>6.3. Material</b> .....	<b>30</b>
<b>6.4. Tasks</b> .....	<b>31</b>
<b>6.5. Procedure</b> .....	<b>33</b>
<b>6.6. Questionnaires</b> .....	<b>36</b>
<b>7. Results</b> .....	<b>38</b>
<b>7.1. Usability Questionnaire</b> .....	<b>38</b>



<b>7.2. Comparative Questionnaire.....</b>	<b>39</b>
7.2.1. Immersiveness .....	39
7.2.2. Easy To Learn .....	39
7.2.3. Easy To Use.....	40
7.2.4. Speed of Use.....	41
7.2.5. Accuracy.....	41
7.2.6. Fun.....	42
7.2.7. Overall Preference .....	43
<b>7.3. Benefits and Drawbacks.....</b>	<b>43</b>
<b>8. Discussion .....</b>	<b>45</b>
<b>9. Conclusions .....</b>	<b>46</b>
<b>10. Further work .....</b>	<b>47</b>
<b>11. References .....</b>	<b>48</b>
<b>12. Appendix A. Questionnaires. ....</b>	<b>51</b>
<b>13. Appendix B. Participant Consent Form .....</b>	<b>59</b>

# INDEX OF FIGURES

Figure 1: Initial Scenario (1), Target Scenario (2).....	8
Figure 2: Methodology scheme .....	9
Figure 3: Gantt chart proposed for the study .....	11
Figure 4: Dragging one finger .....	13
Figure 5: Pinch and spread gestures .....	13
Figure 6: X, Y and Z axes .....	13
Figure 7: Horizontal drag of two fingers .....	13
Figure 8: Vertical drag of three fingers .....	14
Figure 9: Horizontal drag of three fingers .....	14
Figure 10: Rotating for Z-axis.....	14
Figure 11: Rotating for Y-axis .....	15
Figure 12: Rotating for X-axis .....	15
Figure 13: Tablet Multi-touch Technique (above). Tablet Mixed Technique (below).....	15
Figure 14: Simulating pixel matrix.....	18
Figure 15: Distance between two pointers.....	19
Figure 16: Tablet training scenario for multi-touch gestures.....	22
Figure 17: Neo4j example in IPED Toolkit.....	28
Figure 18: IPED Toolkit displaying a video footage.....	31
Figure 19: Scale example .....	32
Figure 20: Rotation Z-axis example .....	32
Figure 21: Rotation Y-axis example.....	32
Figure 22: Rotation X-axis example.....	33
Figure 23: Translation example.....	33
Figure 24: Training scenario .....	34
Figure 25: User study process .....	34
Figure 26: Comparative graph between the three means on Immersiveness feature .....	39
Figure 27: Comparative graph between the three means on Easy to Learn feature.....	40
Figure 28: Comparative graph between the three means on Easy to Use feature.....	40
Figure 29: Comparative graph between the three means on Speed of Use feature.....	41
Figure 30: Comparative graph between the three means on Accuracy feature.....	41
Figure 31: Comparative graph between the three means on Fun feature.....	42
Figure 32: Comparative graph between the three means depending on Overall preference.....	43

## INDEX OF TABLES

Table 1: Sensor availability by Android platform .....	20
Table 2: Creating and sending of JSON objects with SocketIO .....	24
Table 3: Order for run all tasks, for different means by user.....	36
Table 4 Items UMUX methodology .....	37
Table 5: Umux scores for each mean .....	38
Table 6: Benefits and drawbacks of each mean of interaction .....	44

# 1. INTRODUCTION

## 1.1. Overview

Nowadays, Public Display Systems (PDS) are present in many places of our cities. They are part of what is called ubiquitous computing, also known as pervasive computing. The goal of ubiquitous computing is the nonintrusive availability of computers throughout the physical environment, virtually, if not effectively, invisible to the user [Weiser 1993].

PDS are accessible to groups of people and are located in (semi-) public spaces. Usually, these displays provide information that is only relevant around where they are located. These are some of the reasons why public displays are linked to location based services. Jiang and Yao [Jiang and Yao 2006] affirm that there is no clear-cut boundaries of LBS and GIS, as many fundamental research issues of GIS science are those of LBS as well.

The development of PDS is costly and to identify the placement where they should be placed is not trivial. The idea is to use tools or techniques in order to reduce costs designing and evaluating PDS. One of the biggest problems in developing ubiquitous computing applications is the simulation of the required infrastructure and the environment in which they are to be deployed.

There are different methods of evaluation and design for public displays systems. We focus on the IPED toolkit [Ostkamp and Kray 2014]. It uses an immersive video environment technique that is inspired by the system presented by Singh [Singh et al. 2006]. Briefly this technique uses immersive video with surround sound and simulated infrastructure to create a realistic simulation of a ubiquitous environment in the software design and development office.

Until the completion of this work, the only way to interact with this toolkit for designing and prototyping PD is combining the keyboard and mouse means. We think that if one of the goals is to recreate as much as possible the environment of a real location, there are other means of interaction that would increase the feeling of be in this location.

The problem with using the keyboard and mouse is that the user should be away from the three back-projection screens. For the proper operation of keyboard and mouse, these devices must be on a table. So users do not have freedom of motion in the lab, reducing the “immersiveness”.

From our perspective, we think the user should have full freedom of movement in the laboratory. And take advantage of the ability of immersive of the three back-projection screens provided by IPED Toolkit. Users could then stay in between the screens increasing the perception of being physically present in the real location.

Therefore our purpose is to develop an alternative method to the keyboard and mouse for interacting with the toolkit to increase the sense of "being there". Next we carry out one user study in order to get information about the different means. Then we can analyze the data and determine and identify the benefits and drawbacks of the different means of interaction.

## **1.2. Objectives**

The main objective of this work is to increase the sense of reality during the design and prototyping of public displays using the IPED Toolkit. With this purpose we develop a different means of interaction with this toolkit. Our challenge is to prove that it is possible to increase the immersiveness through the use of other means maintaining a good usability degree at the same time.

This main objective can be divided into the following sub-objectives:

- To implement different interfaces for the tablet. These interfaces recognize different actions and gestures made by users.
- To integrate these tablet interfaces with the IPED Toolkit. That integration allows interaction between toolkit and tablet.
- To make a user study comparing tablet and keyboard + mouse means.
- To evaluate the results and determine the benefits and drawbacks from the different means.

### 1.3. Dissertation organization

The structure of this dissertation consists of eleven chapters: *Introduction*, *Literature Review*, *Concept*, *Methodology*, *Implementation*, *User Study*, *Results*, *Discussion*, *Conclusions*, *Further Work* and *References*. There are also two appendices: *Appendix A: Questionnaires*, and *Appendix B: Participant Consent Form*.

The *Literature Review* makes a revision of different tools for designing, prototyping of public displays systems and summarizes the most important points of a selection of literature references that have inspired this work.

The *Concept* chapter shows the different alternatives of means for interact with IPED Toolkit and why the tablet mean has been selected. *Methodology* chapter explains all the steps followed to carry out the study.

The *Implementation* of our research brings together all the aspects related with the implementation of the application. It defines the basics about web technology (necessary to understand the whole), the development of the tablet interfaces and the integration with the IPED Toolkit.

The *User Study* describes the participants of the study, the material needed, the tasks to run and the questionnaires used to obtain the results. The *Results* chapter analyzes the data obtained from the questionnaires.

The *Discussion* shows the limitations and difficulties found during the work. *Conclusions* section explains the achievements of this dissertation and *Further work* chapter makes some comments on things that have not been developed due to time restrictions and that will be interesting to consider in other future works.

*Appendix A* shows the questionnaires used during the user study. In the *Appendix B* there is a sample of the participant consent form.

## 2. Literature Review

Designing and prototyping public display systems becomes challenging when a system consists of many displays in different locations. Recently different methods or toolkits have appeared to reduce costs during deployment of PDS. In this section we present a selection of some interesting tools related to the development of PDS.

One of these methods is the Proximity Toolkit [Marquardt 2011]. This toolkit allows rapid prototyping and researching of newfangled interfaces that are based on a theory of proxemic interaction. This method offers three contributions: the design of a toolkit architecture, which simplifies access to proxemic information; interpretation and representations of higher-level proxemic concepts from low-level information; and the design of complementary visual tools that allow developers to explore proxemic relationships between entities in space without coding. On the other hand, this method lacks of contextual factors, for instance, it does not consider display contents.

Other technique is to make a replica of the location where the public display should be installed. Researchers [Hamhoum and Kray 2011] recreated a real scenario inside the lab. This possible scenario would be locations with dense population with the aim of install PDS to support the navigation in that place. On the one hand, this system calculates the relative density of displays thanks to physical simulation and supports locomotion (although simulating user's locomotion is still difficult). On the other hand, much effort is required to recreate the physical small-scale model. In addition, for each location, a new recreation should be made.

A Stochastic model [Harrison and Massink 2010] is proposed as a tool of prototyping and evaluating ubiquitous systems during early phases of development. They affirm that it is possible to compare different assumptions about volumes, rates and design parameters rapidly with good degree of accuracy. But, the techniques used are complex and sophisticated.

An alternative system is a hybrid prototyping [Nakanishi 2012] using both virtual and miniature simulation. The proper positioning of input and output devices are investigated in virtual space. In miniature simulation they clarify issues coming from differences between virtual and real space.

Modelling and designing user assistance in intelligent environments [A. Geven, J. Schrammel 2006] is a combination of narrative interviews, a scenario-based approach and a Wizard-of-Oz prototyping. An integrated scenario facilitates designers focus on users, a better understanding of users in early phase of development is achieved by using narrative questionnaires and Wizard-of-Oz prototyping allow designers to test early in the design of intelligent environments.

Another approach for the design of intelligent environments [Stahl and Hauptert 2006] is using a geometric 3D of the environment. The environment can be represented with this method through sensor devices and public displays. Then they show real content to the virtual displays, and using an avatar as a positioning loopback device for location-aware user assistance applications. This makes difficult to represent different scenes at the same location.

Other tool for prototyping public displays and in which we are going to focus is on the Immersive Display Evaluation and Design (IPED) Toolkit [Ostkamp and Kray 2014]. This research uses an immersive video environment technique that is inspired by Singh [Singh et al. 2006]. Briefly this technique uses immersive video with surround sound and simulated infrastructure to create a realistic simulation of a ubiquitous environment in the software design and development office. IPED Toolkit supports the design and evaluation of PDS at early development stages by combining panoramic images or video footage with a light-weight, graph-based model to simulate public displays in an immersive video environment. This toolkit reduces the time expended to design and evaluate PDS with relatively little effort. One of the objectives of this approach is to increase the degree of 'situatedness' inside the lab. Three back-projection screens display a panoramic video footage recorded at real-world locations. During prototyping, users can overlay a virtual public display over the video footage using keyboard and mouse.

From our point of view, using alternative means during prototyping could increase even more the perception of being physically in the real location.

There are many different means of interaction in immersive environments. A study [Stefani and Karaseitanidis 2004] describes five input devices for magnetic tracking in virtual immersive environments. On the one hand, they present OmniControl, Wedge and ErgoMike as input devices based on the clench grip. On the other hand, Spoon



and StyleBall are based on the pinch grip. All of them lack of freedom, because are connected by cable.

Smart phones and tablets incorporate a touch screen, which is used as the main input device. The most important use of the touch screen is like mouse substitute of a desktop computer. In this way we can select, drag and drop any item on the screen easily. However the use of the touch screen is broader. Often it is used to replace the keyboard on devices that do not have physical keyboard.

A study [Kim et al. 2009] proposes iPhone and Ipod touch as input devices in immersive virtual environments. The interaction technique implemented was for interaction tasks in a virtual environment. They had some limitations due to the reduced screen size.

Tablets have bigger screen and the possibility of implementing different gesture techniques. An investigation [Kurdyukova et al. 2012] identifies three gesture modalities: multi-touch gestures, performed by manipulating iPad in 3D space, and direct contact gestures, involving the physical contact of iPad and another device. Furthermore gestural interaction is comfortable, fast execution, and intuitive [Kray et al. 2010].

Another method for virtual-environment interaction is based on proprioception [Mine et al. 1997] . This means interaction depends on the position of the body from users. But the precise manipulation of moving objects is still harder in virtual spaces than in real space.

### 3. Concept

This paper intends to find out other means of interaction with the IPED toolkit for prototyping public displays. So the aim of this paper is to find a mean which increases the sensation of reality, and at the same time maintain a high level of usability.

To carry out this study there are different alternatives. One possibility is to interact through tablets. A second option could be through body gestures, using a Kinect camera for gesture recognition. Another alternative would be using a microphone for implement voice recognition. We discuss which of these means of interaction could be selected in order to integrate it with the IPED Toolkit and compare it with the current method of interaction. The current mean implemented is the one combining keyboard and mouse.

Finally the chosen mean is the tablet. The reasons for this choice are set out below:

- Tablets are mobile devices that are growing in popularity. Nowadays more and more people have a tablet at home.
- The way to interact with the tablet is very similar to that of smart phones. So users are familiar with such gestures.
- Tablets allow the implementation of different methods of user interaction. The next section details the different gesture techniques that Kurdyukova [Kurdyukova et al. 2012] identifies in his study with iPads. This is a very good reason, because using a single device we can develop different interaction techniques.

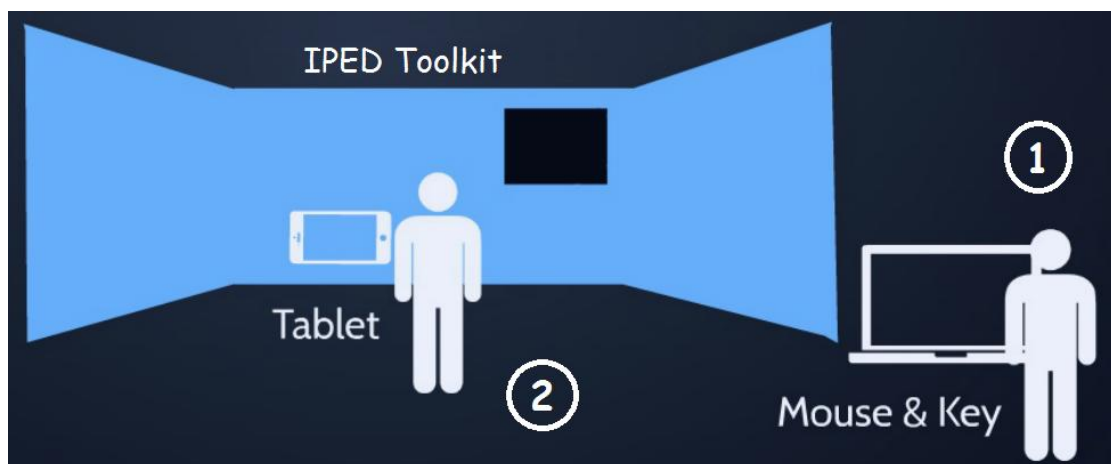
Two different techniques are going to be developed for object motion on a 3D web scenario using the tablet. Three gesture types were identified for iPad [Kurdyukova et al. 2012]:

1. Multi-touch gestures are performed on the iPad screen.
2. Spatial gestures are performed by rotating, tilting or panning the iPad body.
3. Direct contact gestures imply physical contact between the iPad body and the body of another device.

We have chosen the first technique Multi-touch gesture on the tablet screen. For the second one, we combine the two first gesture types identified before (point number 1 and 2). Then the tablet will implement two techniques of interaction:

- Tablet with Multi-touch technique: The interaction is only run when users touch the screen.
- Tablet with Mixed technique: The interaction is performed when users touch the screen and also when the body of the tablet is rotated.

The concept of this work is to implement the means of interaction mentioned, using the tablet device and integrate them to the IPED Toolkit. The goal is to achieve the target scenario shown in the following figure.



**Figure 1: Initial Scenario (1), Target Scenario (2)**

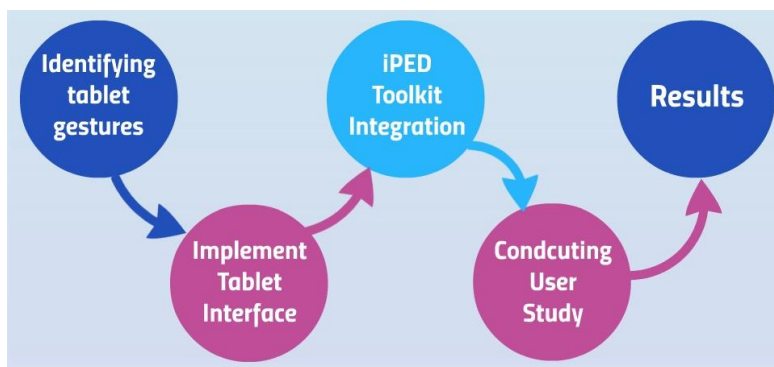
After running a user study and analyzing the results, the paper proposes to answer if there are alternative means of interaction with IPED Toolkit during prototyping public displays, which increase the feeling of reality and maintain a good grade of usability. Next section details how the study is carried out.

## 4. Methodology

The focus of this research is on the design and execution of a user study to identify one usable mean to overlay a virtual display above panoramic video footage. In other words, to perform a comprehensive comparison between different devices implemented in the IPED toolkit to carry out the task mentioned before. We will follow the instructions given by Olsen in which complex systems can be compared and evaluated [Jr 2007].

At this moment, IPED toolkit has implemented one mean to overlay the virtual display on the scene; this is achieved by using both keyboard and mouse. The intention in this research is to integrate a tablet during prototyping PDS with IPED toolkit, and compare the usability with the mouse and keyboard means. Therefore we are going to make a prototype for the tablet.

Next figure schematized the steps followed to complete this study.



**Figure 2: Methodology scheme**

The first step is to identify the different gestures that users can perform on a tablet in order to interact with it. We have to identify gestures for the tablet multi-touch technique and for the mixed technique. In gestures section of the implementation chapter we define all the gestures.

The second step is to implement an android application which recognizes the gestures identified previously. Users should be able to choose the technique of interaction that they are going to use.

After developing the android application, we have to integrate it with the IPED Toolkit. When users perform a gesture, the application should inform the IPED Toolkit about the action that the user wants to run. On the other hand, the IPED Toolkit must understand the message received from the android application in order to apply these changes in the polygon (which represents the virtual display).

Once we have implemented and integrated the two interfaces for the tablet, there will be three means to analyze and compare:

1. Keyboard + Mouse
2. Tablet Multi-touch Technique
3. Tablet Mixed Technique

Now we explain the procedure to be performed for the comparison between different means. This study compares first the grade of usability for each mean. For this purpose a questionnaire is created following the UMUX methodology [Kraig Finstad n.d.]. Furthermore other questionnaire compares different features for each mean. This second questionnaire focuses on ranking the three means in features such as immersiveness, easy to use, fun, and accuracy among others. These questionnaires are shown in detail during the user study section. Finally we analyze and discuss the results before the conclusions.

For the development of this study we follow the Gantt chart shown below. The Gantt chart shows the tasks identified in this study and the time spent in each of these tasks.

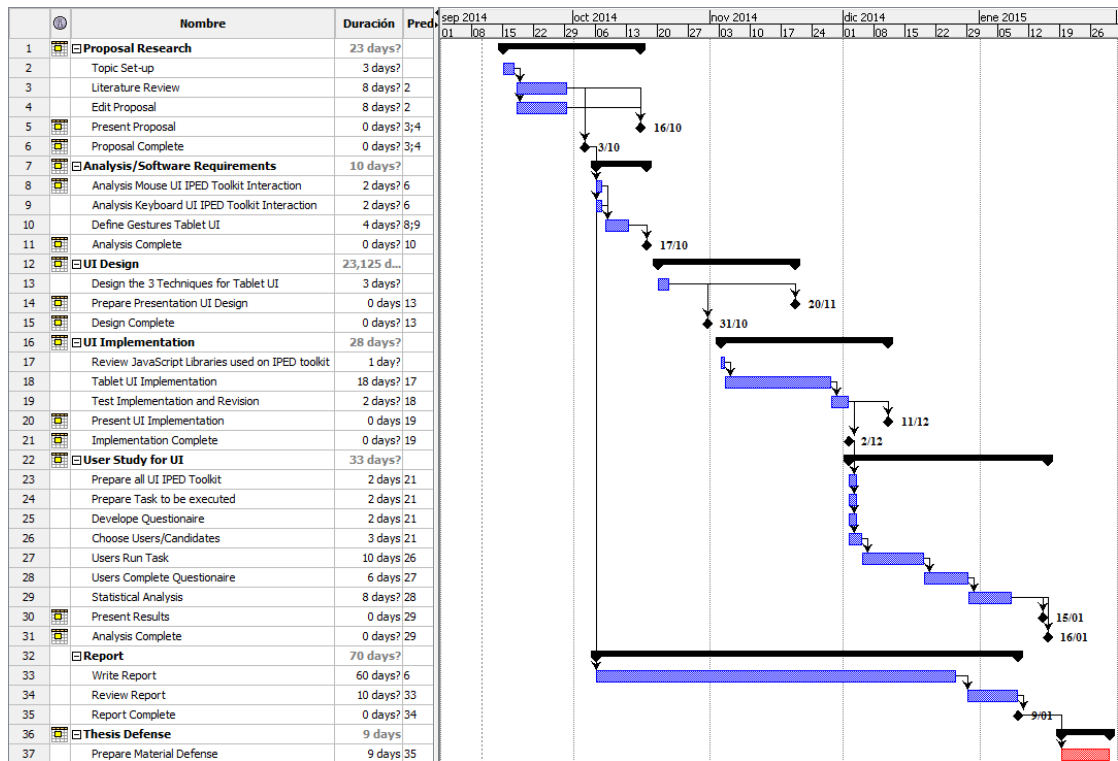


Figure 3: Gantt chart proposed for the study

## **5 Implementation**

This section explains how we develop the different means of interaction with the IPED Toolkit. First we have to deploy the android application for the interaction with the tablet, which recognizes user gestures. Then we complete the integration between the tablet and the toolkit.

### **5.1. Android Application**

In this section we will explain the steps and technologies followed to perform the tablet interface. First we need to identify the user gestures for the tablet. Then, developing the application to recognize the gestures identified. Finally we have to communicate the toolkit about the gesture performed by the user.

#### **5.1.2. Identifying Gestures**

This section explains the gestures that we define to interact with the tablet before we start with the implementation of the application. There are different techniques of interaction with the tablet. For instance, touching the screen and rotating the body of the tablet.

Touching the screen with one finger is known as single-touch. A multi-touch gesture is performed when multiple fingers (pointers) touch the screen at the same time. It is a technology that offers new ways of interaction compared to traditional input devices like keyboard and mouse. Users can manipulate objects or execute commands by means of their hands.

Those are the gestures we define for interacting using a single-touch and multi-touch gestures.

Translate: it is the action to move the polygon which represents the virtual display on the scene. To execute these action users should move their fingertip over the screen of the tablet without losing contact. See figure 4 below.



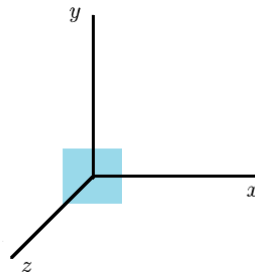
**Figure 4: Dragging one finger**

Scale: it is the action to change the size of the polygon. The possible actions are to increase or decrease the size. To decrease size, users must touch with two fingers and bring them closer. To increase size, users must touch the screen with two fingers and move them apart. Figure 5 shows these gestures.



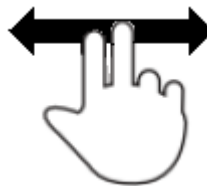
**Figure 5: Pinch and spread gestures**

Rotate Z-axis: it is the action of rotating the object in one of the three axes. Depending on the distance travelled by dragging fingers, the angle should be bigger or smaller. Next figure shows the three axes:



**Figure 6: X, Y and Z axes**

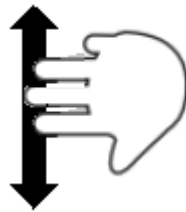
To complete the action of rotating a number of degrees in the Z-axis, users have to drag two fingers in horizontal way. See figure 7.



**Figure 7: Horizontal drag of two fingers**



Rotate X-axis: it is used to complete the action of rotating a number of degrees in the X-axis; users should drag three fingers in vertical manner as shown in figure 8.



**Figure 8: Vertical drag of three fingers**

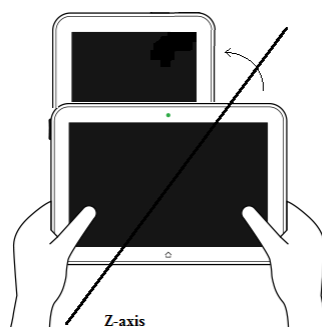
Rotate Y-axis: dragging three fingers over the screen in horizontal sense, the polygon is rotated in the Y-axis.



**Figure 9: Horizontal drag of three fingers**

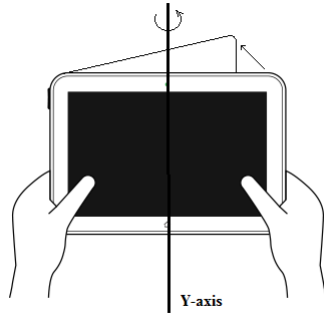
Other way of interaction is performed by rotating the body of the tablet. The three gestures that users can perform are the following:

Rotate Z-axis: the effect on the polygon is the same as rotating in the same axis with Multi-touch gestures. In order to rotate the virtual display in the Z-axis, the tablet should be turned like a wheel. See figure 10.



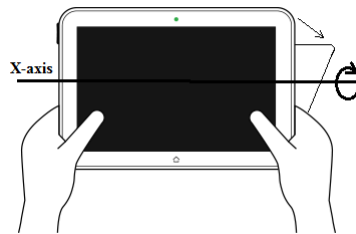
**Figure 10: Rotating for Z-axis**

Rotate Y-axis: Now users have to rotate the body of the tablet like rotating the Y-axis. See figure below:



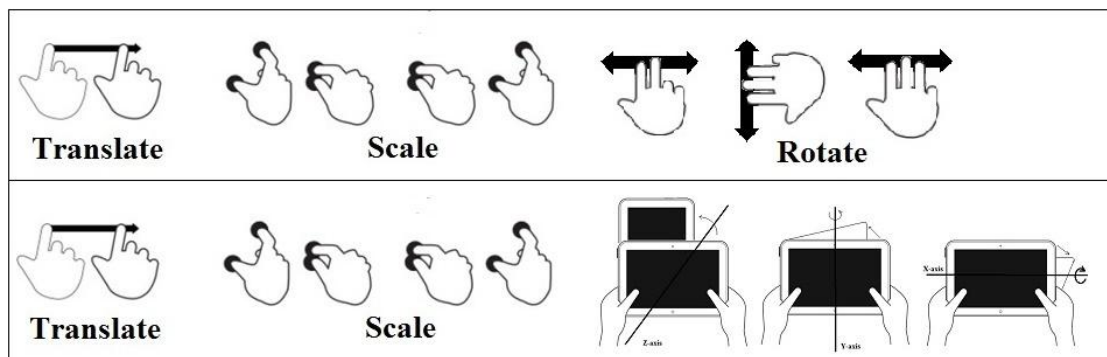
**Figure 11: Rotating for Y-axis**

Rotate X-axis: it is used to rotate the virtual display in the X-axis, users should push the tablet in back sense:



**Figure 12: Rotating for X-axis**

Once we have defined all possible gestures of interaction using the tablet, the next step is to develop the interface. Two techniques of interaction are implemented: multi-touch gesture and mixed gesture technique. The first technique used is just single-touch and multi-touch gestures aforementioned. The mixed gesture technique is a combination of touching and rotating the screen. For translating and scaling actions the gestures used are same as before and for rotating actions the users should rotate the tablet. Figures 13 summarize both techniques:



**Figure 13: Tablet Multi-touch Technique (above). Tablet Mixed Technique (below)**

Next section details how to detect all these gestures using the Android API<sup>1</sup>.

## 5.1.2. Android Application

### 5.1.2.1 Android Overview

Android is an operating system based on Linux kernel. It is designed primarily for a wide range of mobile devices such as smart phones or tablets. Android is an open-source software stack and it is led by Google. One of the keys to the popularity of Android is that, like Linux, is an open source platform, allowing manufacturers, operators and developers give more functionality to their smart phones.

Anyone who can program can create new applications, widgets<sup>2</sup>, or even modify the operating system itself, so knowing Java programming language is very easy to start programming on this platform. The Android application development is very simple and only requires a basic knowledge of Java and the software development kit provided by Google. This kit can be downloaded for free<sup>3</sup>.

The programming language used to implement the android application is Java. Java is a general-purpose computer programming language and it is an object oriented language. Java is used to deploy native android applications. The first step to deploy an android application is to install the Java Development Kit (JDK). It is very important to differentiate between JDK and JRE (Java Runtime Environment). JRE contains the Java virtual machine, and is needed to run applications written in Java. But, for programmers, the necessary is the development kit, the JDK, which also includes the JRE itself.

It is important to know the definition of classes in java. The classes are the basis of Object Oriented Programming (OOP). A class is a structure of data and operations common to set of objects. In other words, it is an abstraction that defines an object specifying which properties (attributes) and available operations have.

---

<sup>1</sup> Application Programming Interface (API) is a set of routines, protocols, and tools for building software applications.

<sup>2</sup> A widget is a small application that provides access to common functions

<sup>3</sup> <http://developer.android.com/develop/index.html> [accessed: November 20th, 2014]

### 5.1.2.2. Handling Multi-Touch Gestures

Android provides a rich application framework that allows build different kind of applications for mobile devices.

The `onTouchEvent()` method is used to intercept the touch events in an Activity or a View class. An Activity class is the base of any Android application with user interface (UI) and Activities use one or more Views to present the user the UI elements. An `onTouchMethod()` returns a parameter, an object of the `MotionEvent` class.

The `MotionEvent` class has 2 methods that can be used to get the action being performed:

- `getAction()` – Returns the kind of action being performed as a packed integer that contains the action code/constant (like `ACTION_MOVE`) and pointer index (which finger for multi-touch). For single touches it is safe to use this function as there is only one finger involved, hence no need to store the pointer index.
- `getActionMasked()` – Returns the action performed without the pointer index information unlike `getAction()`. The associated pointer index can be fetched by calling `getActionIndex()`.

The `getX()` and `getY()` methods are used to get the X/Y coordinates of the pulsation on the screen. With these values we can identify the movements of the fingers on the screen.

When multiple pointers touch the screen at the same time, the system generates the following touch events:

When one or more than one fingers (pointers) touch the screen at the same time, Android system generates the following touch events<sup>4</sup>:

- `ACTION_DOW`: For the first pointer that touches the screen. This starts the gesture. The pointer data for this pointer is always at index 0 in the `MotionEvent`.

---

<sup>4</sup> <http://developer.android.com/training/gestures/multi.html> [accessed: October 10th, 2014]

- ACTION\_POINTER\_DOW: For extra pointers that enter the screen beyond the first. The pointer data for this pointer is at the index returned by `getActionIndex()`.
- ACTION\_MOVE: A change has happened during a press gesture.
- ACTION\_POINTER\_UP: Sent when a non-primary pointer goes up.
- ACTION\_UP: Sent when the last pointer leaves the screen.

We use these events to handle the gestures defined previously for the multi-touch technique on the tablet interface. Let's see how we use the main methods and events to recognize the multi-touch gestures.

With `event.getPointerCount()` function we obtain the number of fingers that are touching the screen at the same moment:

```
//Number of touches
int pointerCount = event.getPointerCount();
```

With `pointerCount` variable we can know the fingers touching the screen at a given time. And depending on that, we can apply different rules to identify the gestures.

### Translate

With one finger touching the screen, user tries to translate the object. See the following example which describes how our application identifies this action.

	X-1,Y-1	X,Y-1	X, Y-1	
	X-1, Y	X,Y	X+1,Y	
	X-1, Y+1	X,Y+1	X+1,Y+1	

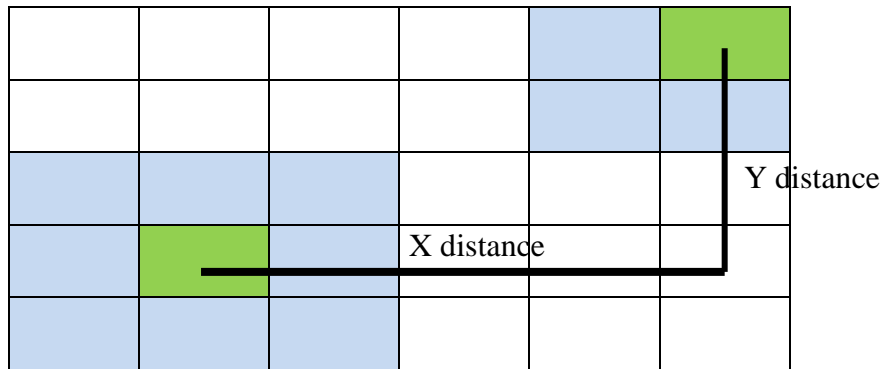
**Figure 14: Simulating pixel matrix**

This matrix represents the pixel matrix of the tablet screen (`pointerCount = 1`). We suppose that user is touching with one finger the green position. So we call `getX()` and `getY()` functions to store the coordinates (X,Y). When user drag the finger, will touch one of the blue positions. If user drag to the right, with `getX()` and `getY()` we obtain

the new coordinates (X+1,Y). Then we compare the current coordinates with the previous coordinates. In our example X-coordinate increases and Y-coordinate remains the same. The application should send a message to the IPED Toolkit with the action (translate) and the new position (increase in 1 unit the X-coordinate and leave the same value for the Y-coordinate).

**Scale**

The application identifies the scale gesture when user touches the screen with two fingers and the distance between the two fingers pointing the screen increases or decreases. As before, we store the coordinates when user touches the screen with two fingers, and we calculate the distance between the two fingers in the X-axis and in the Y-axis. It means that if the coordinates with finger one are (4, 5) and the coordinates with finger two are (8, 8), the distance in the X-axis is 4 and in the Y-axis is 3.



**Figure 15: Distance between two pointers**

If the user drags one or both fingers, we obtain the new coordinates and we calculate again the distance in both axes. Then we compare the distance with the previous calculations. If the distance increase in the X-axis and Y-axis the application send a message informing that the polygon should increase the width and the height respectively.

**Rotate (2 fingers)**

When user touches the screen with two fingers the application store the coordinates of both pointers and the distance between them. If the distance remains the same, and both fingers are moved to the right (X-coordinate of both touches increase), the polygon is rotated on Z-axis in clockwise. If both fingers are moved to the left (X-coordinate of both touches decrease), the polygon is rotated on the same axis in anticlockwise.

In brief, it must meet three conditions. Two fingers touching the screen (pointerCount=2). The distance between the two pointers should remain the same. And the X-coordinate from pointer 1 and X-coordinate from pointer 2 should vary.

### **Rotate (3 fingers)**

For rotating the polygon on X and Y axes, the user should touch the screen with three fingers. If the three fingers are moved in horizontal sense (X-coordinates of the three pointers increase or decrease), we rotate the polygon on the Y-axis. If the fingers are moved in vertical sense, the application identifies a rotation in the X-axis.

### **5.1.2.3. Handling Sensors for Spatial Rotations**

Sensors in Android are defined as a set of devices that can obtain information from the outside world. With them we can implement attractive forms of user interaction.

Android provides access to the internal sensors of the device through the Sensor, SensorEvent, SensorManager classes, and SensorEventListener interface, from the android.hardware package. The sensors available on Android are identified in table 1:

Sensor	Android 4.0 (API Level 14)	Android 2.3 (API Level 9)	Android 2.2 (API Level 8)	Android 1.5 (API Level 3)
<a href="#"><u>TYPE_ACCELEROMETER</u></a>	Yes	Yes	Yes	Yes
<a href="#"><u>TYPE_AMBIENT_TEMPERATURE</u></a>	Yes	n/a	n/a	n/a
<a href="#"><u>TYPE_GRAVITY</u></a>	Yes	Yes	n/a	n/a
<a href="#"><u>TYPE_GYROSCOPE</u></a>	Yes	Yes	n/a	n/a
<a href="#"><u>TYPE_LIGHT</u></a>	Yes	Yes	Yes	Yes
<a href="#"><u>TYPE_LINEAR_ACCELERATION</u></a>	Yes	Yes	n/a	n/a
<a href="#"><u>TYPE_MAGNETIC_FIELD</u></a>	Yes	Yes	Yes	Yes
<a href="#"><u>TYPE_ORIENTATION</u></a>	Yes	Yes	Yes	Yes
<a href="#"><u>TYPE_PRESSURE</u></a>	Yes	Yes	n/a	n/a
<a href="#"><u>TYPE_PROXIMITY</u></a>	Yes	Yes	Yes	Yes
<a href="#"><u>TYPE_RELATIVE_HUMIDITY</u></a>	Yes	n/a	n/a	n/a
<a href="#"><u>TYPE_ROTATION_VECTOR</u></a>	Yes	Yes	n/a	n/a
<a href="#"><u>TYPE_TEMPERATURE</u></a>	Yes	Yes	Yes	Yes

**Table 1: Sensor availability by Android platform**

This does not mean that all devices have all sensors. The sensor contained in each device depends on the specifications of the mobile manufacturer.

To handle the sensors in our android application, we have created a new class which implements the `SensorEventListener` interface. This class is called `Orientation` and uses the `Sensor.TYPE_ROTATION_VECTOR`. The rotation vector represents the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle  $\theta$  around an axis (x, y, or z).

`Sensor.TYPE_ROTATION_VECTOR` is not a physical sensor as such, but a combination of the following physical sensors: accelerometer, gyroscope and magnetic. The information acquired from this sensor is all we need to identify the rotation of the tablet when users interact with it.

The `Orientation` class implemented returns three values: pitch, roll and azimuth. When pitch value changes, it means that the user rotates the tablet on X-axis, and then we should rotate the polygon on the X-axis. When roll value changes, it means that the tablet has been rotated on the Z-axis, and then the virtual display should be rotated on the Z-axis. The same for the azimuth attribute, on the Y-axis.

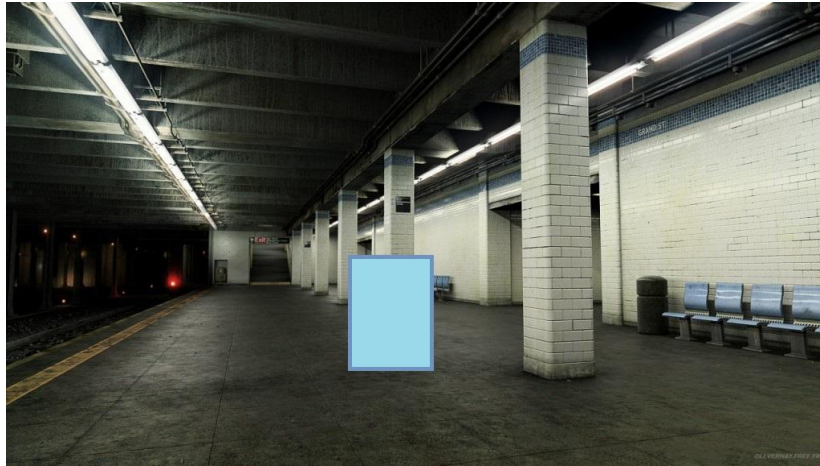
#### **5.1.2.4. Creating a training scenario to test the gestures implementation**

OpenGL API is a standard specification that is used to produce 3D graphics. Android OpenGL works using the same theory to represent 3D graphics on the Android platform.

We have implemented a test scenario for the android application using OpenGL. The idea is to simulate a scenario with a polygon centred on the screen. At the same time that we implemented the algorithms to recognize the gestures, we transform the polygon using the OpenGL library, according to the gestures identified. The purpose of this is to check simultaneously the result of the multi-touch gesture implementation with the polygon transformation. Because otherwise, we could not check the result until we finish the integration between the tablet interfaces with IPED Toolkit.

In other words, we have developed a training scenario using OpenGL for Android, with the aim of testing the different gestures on the tablet. Figure 16 depicts the scenario used during multi-touch gestures detection:





**Figure 16: Tablet training scenario for multi-touch gestures**

To perform this scenario three java classes have been created. The Square.java class draws the polygon which represents the overlay or virtual public display. The GLRendererEx.java class is responsible to execute the gestures recognized. For instance, if scale gesture is recognized then this class runs the glScalef command on the Square. The different actions are glTranslatef(), glScalef() and glRotatef(). The last class is called MyGLSurfaceView in which we extend<sup>5</sup> the GLSurfaceView class from OpenGL. In this view we render the polygon.

All the code used to implement the training scenario is commented in the last version of the application, because we do not want to display any scenario on the screen for not mislead the user. Thereby the user focuses on the three back-projection screens.

#### **5.1.2.5. Sending commands to the IPED Toolkit**

Once the android application is capable to recognize the gestures, it should send the message to the IPED Toolkit. To achieve this purpose, we have to connect the tablet with the IPED Toolkit. Using a hotspot, we connect the tablet with the toolkit by Wi-Fi.

The technology used to establish the communication is using sockets. The reason is because the IPED Toolkit uses this technology for communication between backend and frontend, so it will be easier to accomplish the integration.

---

<sup>5</sup> **Extend** a class mean create a new class that inherits the attributes and methods of another.

The application uses the Socket.IO v1.x Client for Java, which is simply ported from the JavaScript client. It is developed by Naoyuki Kanezawa<sup>6</sup> and it is possible to find the source code on his GitHub account.

The purpose of this library is to allow two-way real-time communication between client and server. This library is essential to establish communication between the tablet (client) and the IPED Toolkit (Server) because the toolkit is implemented with Socket.IO for JavaScript, and we need at least the same version for a proper communication.

The android application developed establishes the communication with the server. We create a socket connexion with the server IP address. The command that allows the connexion is similar to:

```
socket = IO.socket("http://192.168.1.167:8080");
socket.connect();
```

As we can see, first we create the socket with the server IP address, and then we try the connexion. The second step is when a gesture has been recognized. Then the application prepares the message. There are three possible messages for send to the server:

```
socket.emit("translateOverlay", obj);
socket.emit("scaleOverlay", obj);
socket.emit("rotateOverlay", obj);
```

The first parameter on the function "socket.emit" indicates one of the three different actions to apply on the polygon (virtual display). The second one is a JSON<sup>7</sup> object. It is used to exchange information between web services and REST APIs. Its simplicity and easy to implement give a great performance. We use this object to store the information that we want send to the server. For each action, we store distinct information. Next table shows the different JSON objects:

<b>Translate</b>	<pre><b>JSONObject obj = new JSONObject(); obj.put("x", sentX); // translate sentX units obj.put("y", sentY); // translate sentY units</b></pre>
------------------	--

---

<sup>6</sup> <https://github.com/nkzawa/socket.io-client.java> [accessed: October 10th, 2014]

<sup>7</sup> JSON: JavaScript Object Notation is an open standard that uses plain text to encode information in the form attribute: value. It is defined in the European Computer Manufacturers Association as standard ECMA-404 [ECMA 2013].

	<code>socket.emit("translateOverlay", obj);</code>
<b>Scale</b>	<code>JSONObject obj = new JSONObject(); obj.put("x", sentX); obj.put("y", sentY); socket.emit("translateOverlay", obj);</code>
<b>Rotate X</b>	<code>JSONObject obj = new JSONObject(); obj.put("axis", "X"); obj.put("angle", rx); // rx = number of degrees socket.emit("translateOverlay", obj);</code>
<b>Rotate Y</b>	<code>JSONObject obj = new JSONObject(); obj.put("axis", "Y"); obj.put("angle", ry); // ry = number of degrees socket.emit("translateOverlay", obj);</code>
<b>Rotate Z</b>	<code>JSONObject obj = new JSONObject(); obj.put("axis", "Z"); obj.put("angle", rz); // rz = number of degrees socket.emit("translateOverlay", obj);</code>

**Table 2: Creating and sending of JSON objects with SocketIO**

## 5.2. IPED Toolkit Integration

This section explains the main modifications on the source code of the IPED Toolkit for enabling and understanding communications with the tablet mean.

### 5.2.1. IPED Toolkit Server

The server is implemented using Nodejs and Socket.io<sup>8</sup> technologies. Nodejs is a server-side Javascript interpreter that changes the notion of how a server should work. Their goal is to allow a developer to build highly scalable applications and write code that handles tens of thousands of simultaneous connections in a just a physical machine. And Socket.io is a library in JavaScript for Node.js that enables bidirectional real-time communication between client and server. This tool is important for handle inputs and outputs from the server side (IPED Toolkit Backend) to the client side (IPED Toolkit Frontend or Tablet Interface) and vice versa.

The main changes in the server code are aimed at understanding communication with the tablet. Basically the server is listening connexions, and when identifies a message from the tablet, it sends to the frontend the data received.

```
// Socket.io packages
var io = socketio.listen(httpServer);
```

<sup>8</sup> <http://socket.io/> [accessed: October 15th, 2014]

```

io.on('connection', function(socket) {
  console.log('socket.io[connection]: New connection');

  // Translation
  socket.on('translateOverlay', function(data) {
    console.log('socket.io[translate]:' + JSON.stringify(data)); // data.x , data.y
    io.emit('translateOverlay', data);
  });

  // Scale
  socket.on('scaleOverlay', function(data) {
    console.log('socket.io[scale]: ' + JSON.stringify(data));
    io.emit('scaleOverlay', data);
  });

  // Rotate Multi-Touch
  socket.on('rotateOverlay', function(data) {
    console.log('socket.io[rotate]: ' + JSON.stringify(data));
    io.emit('rotateOverlay', data);
  });

  // Rotate Mixed
  socket.on('rotateOverlayMixed', function(data) {
    console.log('socket.io[rotateMixed]: ' + JSON.stringify(data));
    io.emit('rotateOverlayMixed', data);
  });
});

```

The code above shows how the server is listening connexions and sending to the frontend the data in case the connexion comes from the tablet.

### 5.2.2. IPED Toolkit Front-End

The front-end<sup>9</sup> of IPED Toolkit displays a polygon representing the virtual display above the video footage in a web browser. The changes implemented on the frontend for integrating with tablet, are recognizing and running commands coming from the server. This commands change the position, size or rotation of the polygon.

The code modification in the frontend is short. When the application activates the web sockets we have to handle the sockets coming from tablet. Once the socket is identified, we call the corresponding function for transform the polygon.

```

// -----
// Code for integrate Tablet UI -- Javi Morata
// -----

```

---

<sup>9</sup> The Front-End is the part of software that interacts with the user or users

```

// Translate
this.socket.on('translateOverlay', function(data) {
  // Tell three.js to rotate overlay
  thiz.translateOverlay(data);
});

// Scale      thiz.setLocationId(data);

this.socket.on('scaleOverlay', function(data) {
  // Tell three.js to rotate overlay
  thiz.scaleOverlay(data);
});

// Rotate
this.socket.on('rotateOverlay', function(data) {
  // Tell three.js to rotate overlay
  thiz.rotateOverlay(data);
});

// Rotate Mixed
this.socket.on('rotateOverlayMixed', function(data) {
  // Tell three.js to rotate overlay
  thiz.rotateOverlayMixed(data);
});

```

The functions which handle the polygon transformations are implemented in the overlayPlugin file. The goal of this plugin is explained in the next section.

### 5.2.3. Overlay Plug-in

The overlay plug-in<sup>10</sup> is the part of the code which is responsible for performing the transformations of the polygon. We use Three.js<sup>11</sup> library to display the polygon (which represents the public display) above the video footage on the browser. Also, it manages the different actions, such as translate, scale or rotate, when users interact with the toolkit by different means (keyboard-mouse, and tablet).

The methods implemented in this plug-in are:

---

<sup>10</sup> Plug-in is a software component that adds a specific feature to an existing software application.

<sup>11</sup> Threejs is a JavaScript library that makes WebGL (standard specification developed to display 3D graphics) very easy. Three.js does an excellent job of abstracting away many of the details of WebGL. Three.js is quite light and very efficient library for generating and animating 3D graphics within the browser.

- *translateOverlay()*: This function receives two parameters, the increment for the x-coordinate and the increment for the y-coordinate and sets the new position for the centre of the polygon. When user moves the finger to the right on the screen, the current position of the x-coordinate of the centre of the polygon increases in one unit (each time the app detects moving right). If the movement is to the left then decreases in one unit. Also, when user moves the finger down, the y-coordinate of the centre of the polygon increases in one unit and decreases when user moves the finger up.
- *scaleOverlay()*: Similar as before, the function receives two parameters. The increment of the width and the increment of the height. Each time the app detects increasing of the distance between the two pointers (the two fingers touching the screen) in the x-axis, the width increases in one unit. If the distance decreases, the width is reduced. Alike the distance increases in the y-axis, the height increases. If the distance decreases, the height is reduced again.
- *rotateOverlay()*: This method receives two parameters as well. In this case the axis to be rotated, and the increase of the angle. Inside the method, the angle value of the selected axis is updated depending on the angle parameter.
- *rotateOverlayMixed()*: This method is similar to the previous one. But, in this case is called when the app detects a rotation using the mixed technique.

#### 5.2.4. Creating Scenarios

The backend of the IPED Toolkit allows user create locations and assign a video footage and overlays (polygons or virtual displays) to them. For this purpose, the backend displays a map of the city. The technology used for displaying the map is Leaflet<sup>12</sup>.

For creating training and test scenarios we use the backend of the toolkit. We select different locations and different video footage. Then we create an overlay (polygon)

---

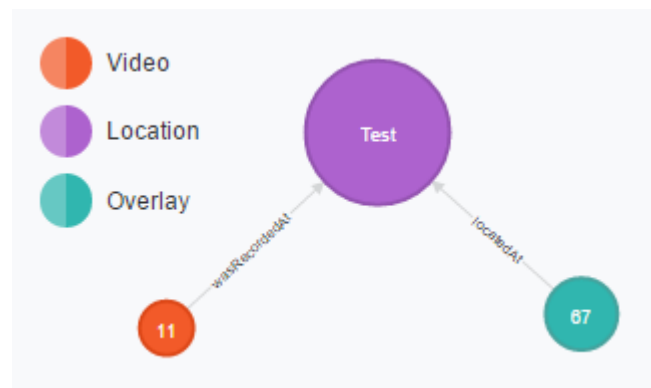
<sup>12</sup> Leaflet is an open source JavaScript library to create interactive maps. <http://leafletjs.com/> [accessed: October 15th, 2014]

for each scenario. The toolkit uses Neo4j<sup>13</sup> for store as a graphs, locations, video footage and overlays.

Formally, a graph is a collection of vertices and edges, or nodes and relationships that connect them. Graphs have a number of simple features:

- Consist of nodes and relationships.
- The nodes contain properties (such as key-value).
- Relationships have a name and always go from one source node to destination node.
- Relationships may also have properties.

An example of the graph obtained when creating our scenarios is:



**Figure 17: Neo4j example in IPED Toolkit**

This graph shows three different kinds of nodes: locations, videos, and overlays. The relationships between them can be defined as:

- A video -- was recorded at -- a location.
- An overlay -- is located at -- a location.

---

<sup>13</sup> Neo4j is the leading graph database. <http://neo4j.com/> [accessed: October 15th, 2014]

## **6. User Study**

### **6.1. Introduction**

The purpose of this study was to make a comparison between different interfaces to place an object in a specified position above the video footage using the iPED Toolkit.

This section describes the steps followed to carry out the user study in order to find out the benefits and drawbacks of different means of interaction with the iPED Toolkit. Participants had to execute 5 tasks using 3 different interfaces. After run the tasks for each interface, the users completed a questionnaire. With this questionnaire was possible to get a measurement of perceived usability for each interface following the UMUX methodology. Once finished the experiment, users filled a comparative questionnaire. The goal of this last questionnaire was to identify the benefits and drawbacks of the different means.

### **6.2. Participants**

Studies with large number of participants are difficult and expensive to run. Each participant involves substantial costs for recruiting, enrolling, conducting the study and managing data. On the other hand, studies with one or few participants have very real limitations. These studies rarely produce statistically significant results; the conclusions obtained from these small studies are extremely limited.

Empirical studies or controlled experiments require a sample group of participants large enough to produce statistically significant results. A research study needs at least between 15 and 20 participants [Lazar et al. 2010]. The inclusion of more than 20 users gives more statistical power. But, as each participant comes with costs in time and other expenses, there are always good arguments in favor of limiting the size of the study. This study has been performed with the number of 18 participants.

The IPED Toolkit proposes a new prototyping and evaluation method for public display systems (PDS). The final users of this toolkit are engineers or designers of PDS. So, we should have collected participants with strong experience on PDS. But, it



was very difficult to get around 20 experts or designers of PDS. To counteract this disadvantage, participants had at least a similar level of education. Master students, PhD students or graduates could fit this profile. All participants that are involved in the study had a requirement. All of them were familiar with computers (ability to use keyboard and mouse) and Tablets or Smartphones (ability to interact with gestures).

### **6.3. Material**

The following list shows all the materials used to run the study:

*Mouse and Keyboard:* It is the first means implemented for overlaying virtual displays with the IPED Toolkit. Users can interact pressing a combination of keys and dragging the mouse. The actions of the different keys are:

“I”: Start the interaction

“W”: Translate virtual display (object or polygon).

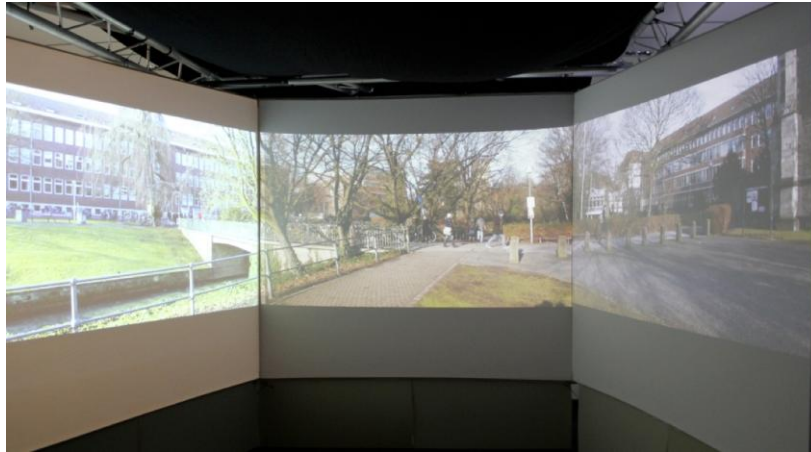
“E”: Rotate virtual display (object or polygon).

“R”: Scale virtual display (object or polygon).

*Tablet:* with the tablet users can interact with the toolkit in two different ways. One of the techniques is touching the screen of the tablet, called multi-touch technique. The second technique is combining two technologies multi-touch and spatial gestures.

*Video footage:* the study plays four panoramic video footage recorded at real-world locations. Two of them are recorded in indoor environments and the other two in outdoor environments. The meaning to have different video footage is just to avoid possible interferences on the results. Sometimes to overlay an object in a video footage is easier than other video footage.

*IPED Toolkit:* Three back-projection screens (200 cm x 150 cm) are arranged in a semi-circular manner, spanning a viewing angle of about 114°. The panoramic video footages are played back on the browser in order to immerse users into these settings.



**Figure 18: IPED Toolkit displaying a video footage**

*Tasks Document:* The document printed with the order of all tasks to be executed for each user. This document represents the table performed in the procedure section. Table 3 shows that tasks document.

*Questionnaires:* users completed three questionnaires. First one was about personal information. The second was following the UMUX methodology (explained in questionnaire section) for each interface. Last one was a comparative questionnaire, where the users ranked the three interfaces for different features. See all questionnaires in the Appendix A.

## **6.4. Tasks**

There were three different actions that users performed. Move the virtual public display from an initial position to a final position (translate). Scale the size of the overlay to a specified size (scale). Rotate the polygon that represents the virtual display in three different axes (rotate).

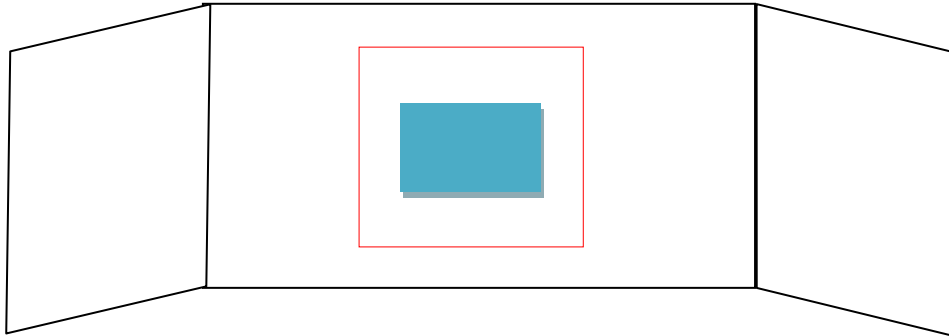
The following five tasks were identified from these three different actions:

- Translation (TR): users have to move the object on the scene (figure 23).
- Scale (SC): users have to scale the object in a specified size (figure 19).
- Rotation X-axis (RX): users have to rotate the object in the X-axis (figure 22).
- Rotation Y-axis (RY): users have to rotate the object in the Y-axis (figure 21).
- Rotation Z-axis (RZ): users have to rotate the object in the Z-axis (figure 20).

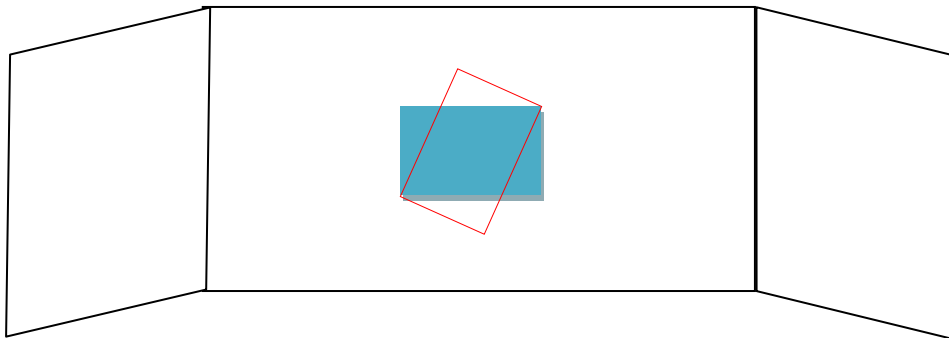
Participants completed each task for all different means. They had to run all the tasks three times, one time for each mean.

For each task, users placed the object inside a concrete mark in the scene. It was not necessary that the object fits exactly inside the mark; the task was completed when the object fit in an acceptable manner.

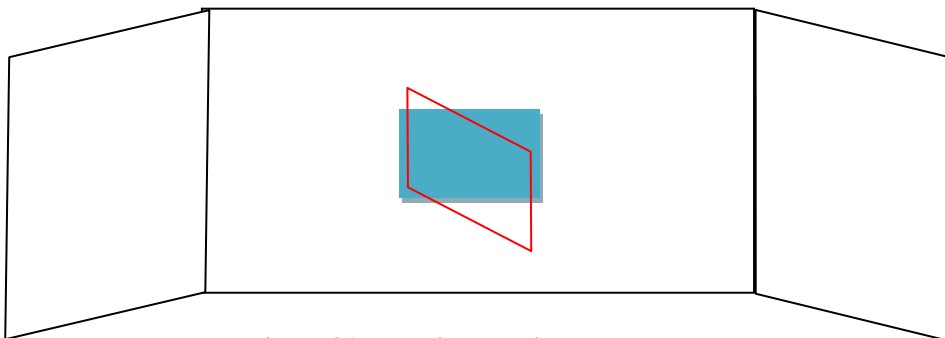
Examples:



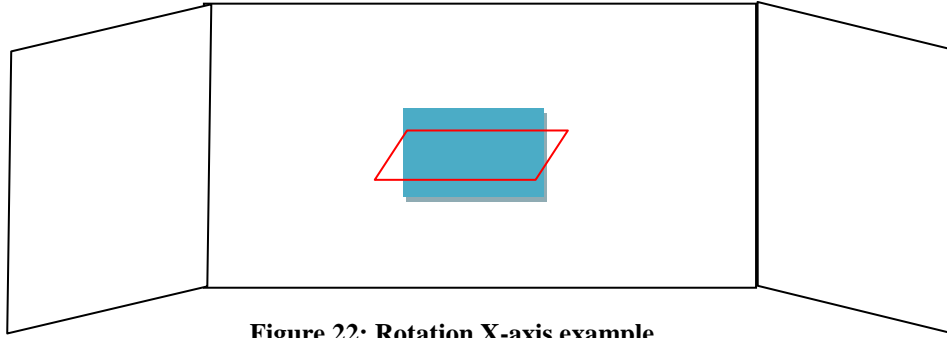
**Figure 19: Scale example**



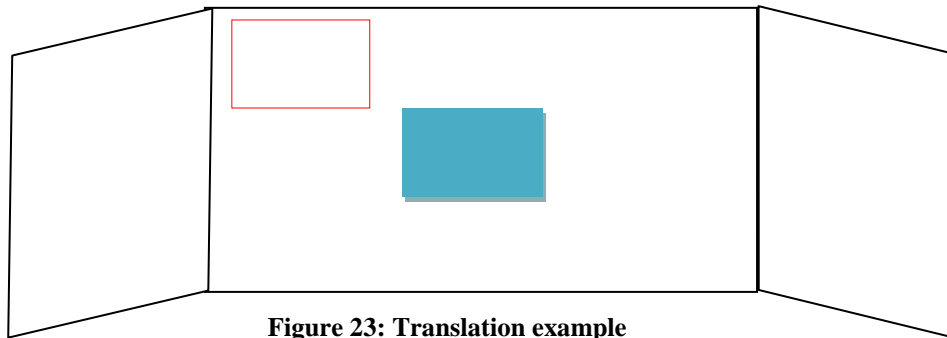
**Figure 20: Rotation Z-axis example**



**Figure 21: Rotation Y-axis example**



**Figure 22: Rotation X-axis example**



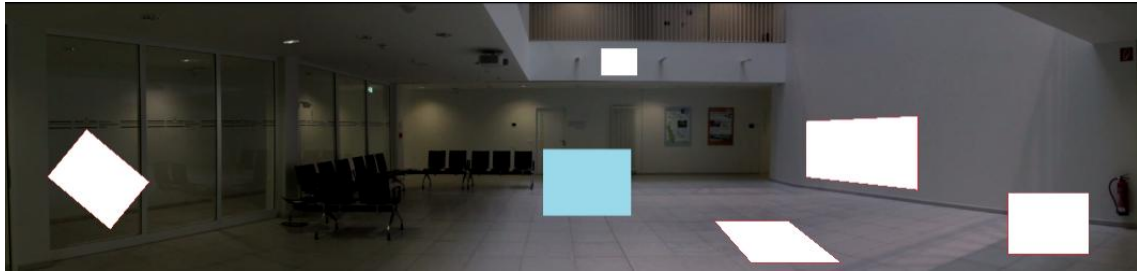
**Figure 23: Translation example**

## **6.5. Procedure**

Participants came to the lab one by one. This is because all subjects should start with the same conditions. If they come in groups, the last participant could have different initial conditions after he see other participants running the tasks.

When the subjects came to the lab, they were informed about the goal of the study. Then, participants should read and sign the participant consent form performed. In this document we informed to users about their privacy protection, their freedom to cancel the participation at any time of the process, their right to delete their data and their non-payment for participation among other things. On the appendix B there is a full copy of the participant consent form.

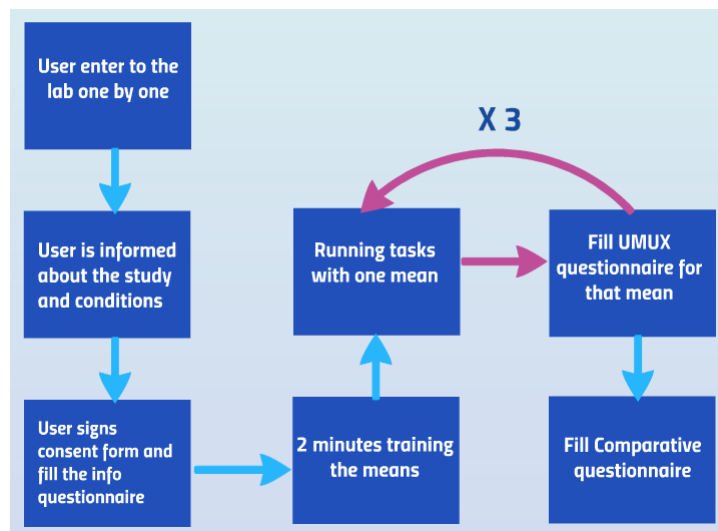
Participants had some time of training before start running the tasks. They had some minutes until they understood how to perform the different actions with each mean. A special scenario was prepared for this aim:



**Figure 24: Training scenario**

Figure 24 shows the scenario where the users could practice different actions with the interfaces. At the same time was useful for a better understanding of the tasks to do on the study.

Once users understand the study and accept to participate on it, they could start the user study. Users executed the tasks using one interface, after complete all the tasks with the interface, they filled a questionnaire. And so on for each mean. Finally participants answered a comparative questionnaire. This whole process is outlined in figure 25.



**Figure 25: User study process**

In order to avoid any kind of external influence on the results, tasks, video footage and means were combined in different order. There were 3 means of interaction (keyboard + mouse, tablet with 1° technique and tablet with 2° technique). Also, there were 5 possible actions (Translate, Scale, Rotate X-axis, Rotate Y-axis and Rotate Z-axis). Each user completed 15 tasks (3 means x 5 actions), and each task was numbered from the 1° until 15°. Also there were 4 different video footages for carry out the study.

To get a random combination of the tasks we used a research randomizer tool. It is a website designed for researchers and students who want a quick way to generate random numbers or assign participants to experimental conditions<sup>14</sup>.

We number from 1 to 5 the 5 tasks for the first interface, from 6 to 10 for the 5 tasks of the second interface and from 11 to 15 the 5 tasks of the third interface. Then, using the research randomizer tool, we randomize the three lists [1..5], [6..10] and [11..15] once for each participant. Also we number from 1 to 3 the three different interfaces. Again we randomize the list [1, 2, 3] one time for each subject.

Table 3 shows the order of the tasks for each participant. The acronyms for the different actions are:

TR: Translation task

SC: Scale task

RX: Rotation in X-axis

RY: Rotation in Y-axis

RZ: Rotation in Z-axis

User	Video Footage	Means														
		Mouse + Keyboard					Tablet Multi-Touch					Tablet Mixed Tech.				
		T	S	R	R	R	T	SC	R	R	R	T	SC	R	R	R
	R	C	X	Y	Z	R		X	Y	Z	R		X	Y	Z	
1	1	3	1	5	2	4	15	11	14	13	12	9	6	10	8	7
2	2	14	15	11	13	12	5	4	3	2	1	10	7	6	5	9
3	3	9	5	6	10	8	11	14	12	13	15	4	5	3	1	2
4	4	11	12	15	14	13	2	4	5	3	1	6	8	10	7	9
5	1	5	2	1	4	3	15	12	11	13	14	9	10	8	6	7
6	2	15	12	13	11	14	6	8	10	7	9	4	2	1	5	3
7	3	3	5	1	2	4	9	6	8	7	10	11	14	12	13	15
8	4	9	10	7	6	8	2	4	3	5	1	13	14	12	15	11
9	1	8	6	9	10	7	13	11	12	15	14	5	4	2	3	1
10	2	14	15	12	11	13	8	9	10	7	6	3	5	1	4	2
11	3	4	2	1	5	3	11	15	13	14	12	10	6	8	9	7
12	4	7	10	6	9	8	3	2	1	4	5	14	11	12	15	13

<sup>14</sup> Urbaniak, G. C., & Plous, S. (2013). Research Randomizer (Version 4.0) <http://www.randomizer.org> [accessed November 18th 2014]

<b>13</b>	1	4	2	5	1	3	14	15	13	12	11	8	7	9	6	10
<b>14</b>	2	11	12	15	13	14	8	10	6	7	9	1	2	4	3	5
<b>15</b>	3	7	6	9	8	10	1	2	4	3	5	13	11	12	14	15
<b>16</b>	4	2	5	1	4	3	13	14	12	11	15	6	8	7	9	10
<b>17</b>	1	2	4	3	5	1	8	10	7	6	9	12	11	14	15	13
<b>18</b>	2	13	14	15	11	12	1	4	5	2	3	10	9	8	7	6

**Table 3: Order for run all tasks, for different means by user.**

## 6.6. Questionnaires

A questionnaire is a tool for collecting and recording information. It is made up of a list of questions, but should also include clear instructions. Questionnaires should always have a definite purpose that is related to the objectives of the research, and it needs to be clear from the outset how the findings will be used.

The biggest single advantage is that a usability questionnaire gives you feedback from the point of view of the user. If the questionnaire is reliable, and you have used it according to the instructions, then this feedback is a trustworthy sample of what you (will) get from your whole user population [Kirakowski 2000].

The purpose of perform the questionnaire is to look at the basic attitudes/opinions of a group of people relating to a particular issue.

We performed three different questionnaires that participants should fill. The goal of the first questionnaire is to know information about the subjects. Age, gender, occupation, dominant hand...

Second questionnaire was made following the usability metric for user experience (UMUX) [Finstad 2010]. This is a relatively new standardized usability questionnaire designed to get a measurement of perceived usability consistent with the System Usability Scale (SUS), but using less items [Lewis et al. 2013]. SUS is a ten-item questionnaire with a scale of five points. Its purpose was to provide an easy test to complete (with few number of items), easy to score and that would allow for cross-product comparisons [Brooke 1996].

The UMUX methodology has four items and have seven scale steps from 1 (strongly disagree) to 7 (strongly agree). The score can range from 0 to 100. Participants answered the four items after to use each interface. Table 4 shows the items.

INTERFACE							
Questions	Strongly Disagree			Strongly Agree			
This system's capabilities meet my requirements.	1	2	3	4	5	6	7
Using this system is a frustrating experience.	1	2	3	4	5	6	7
This system is easy to use.	1	2	3	4	5	6	7
I have to spend too much time correcting things with this system.	1	2	3	4	5	6	7

**Table 4 Items UMUX methodology**

The last questionnaire is a comparative questionnaire. The purpose is to identify the benefits and drawbacks of the three different interfaces. We compare them among the following features: immersiveness, easy to learn, easy to use, speed of use, accuracy and fun. Participants should rank with best technique, second best technique and worst technique the three means for each feature.

*Immersiveness*: feeling to be in the real location.

*Ease to learn*: it is easy to learn to use it.

*Ease to use*: it is easy to use.

*Speed of use*: tasks completed faster.

*Accuracy*: tasks completed with more precision.

*Fun*: it is fun to use it.

*Overall preference*: technique preferred in general terms.

The three questionnaires have been created using Google Forms. Google Forms can simply be described as free, collaborative survey software. It records all responses in Google Spreadsheets which is just one subset of the Google Docs applications and can be downloaded as a Microsoft Excel or Open Office file for further manipulation. Appendix B contains an example of each questionnaire.



## 7. Results

### 7.1. Usability Questionnaire

The purpose of this usability questionnaire is to have a value or score, indicating the level of usability in order to compare between them. To obtain the UMUX score, results were processed following the UMUX procedure.

Odd items were scored as [score – 1], and even items were scored as [7 – score]. To achieve parity with the 0–100 range provide, a participant’s UMUX score is the sum of the four items divided by 24, and then multiplied by 100. These scores across participants are then averaged to find a mean UMUX score. See table below.

MEAN	UMUX Score
<b>Keyboard + Mouse</b>	81.48
<b>Tablet Multi-touch Technique</b>	65.74
<b>Tablet Mixed Technique</b>	71.29

**Table 5: Umux scores for each mean**

After analyzing the results, the keyboard and mouse mean obtains a higher score. Between the tablet interfaces, the implementation with mixed technique is which has a best grade. And the worst grade is for the tablet with multi-touch technique.

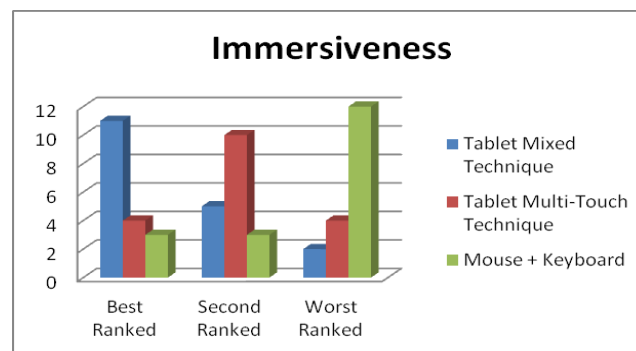
In addition, some participants helped giving feedback. For instance, a group of participants wrote that the tablet mean is too much sensitive. That difficulties precision tasks. Also five participants observed that when they try to rotate using multi-touch gesture, sometimes the polygon increases the size instead to rotate.

The following section detail some of these problems encountered and that may be the cause of a lower score usability in the tablet interface.

## 7.2. Comparative Questionnaire

### 7.2.1. Immersiveness

Immersiveness means the feel of reality that users can perceive. Analyzing the graph obtained from the answers (figure 26), we can identify easily that users have a higher feeling of reality using the tablet with the mixed technique. The second preferred mean is the tablet with multi-touch technique. And the worst interface for feeling the reality is the keyboard and mouse.

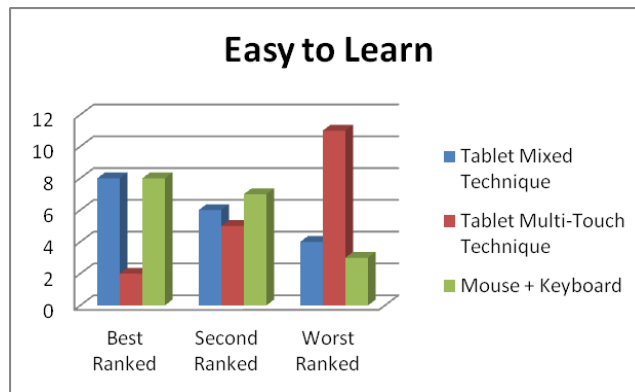


**Figure 26: Comparative graph between the three means on Immersiveness feature**

One of the reasons of this finding could be because with the tablet means, users can stay nearer from the three back-projection screens. When users use the keyboard and mouse means, they are in the desktop, far from the screens. And between the two tablet techniques, they choose the mixed technique as better, because the rotation gestures are more like reality.

### 7.2.2. Easy To Learn

In this case the results shown in figure 27 are not as clear as before. Keyboard and mouse, and tablet with mixed technique have roughly the same votes for the best and second best mean for “ease to learn” feature. The worst technique in easy to learn, is the tablet multi-touch, this could be because the rotation gestures are not intuitive.

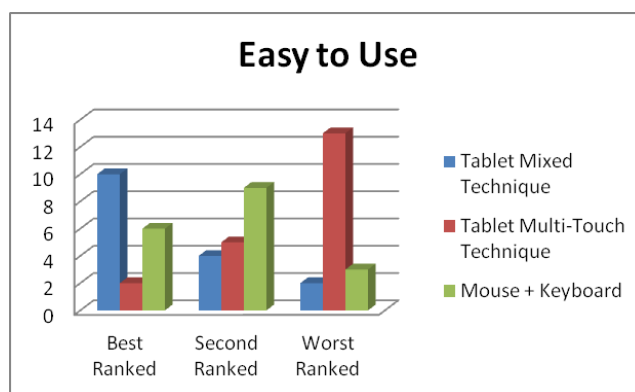


**Figure 27: Comparative graph between the three means on Easy to Learn feature**

It is harder to explain why some people choose as best keyboard and mouse, and others tablet with mixed technique. Maybe some of them prefer the intuitiveness of tablet mixed gestures. And the rest prefer the classical mean, keyboard and mouse.

### 7.2.3. Easy To Use

The results for “easy to use” are similar to the above. In figure 28 we can observe that the worst interface is the tablet with multi-touch technique again. But then, tablet with mixed technique seems that has a few more votes, relegating the keyboard and mouse to second place.

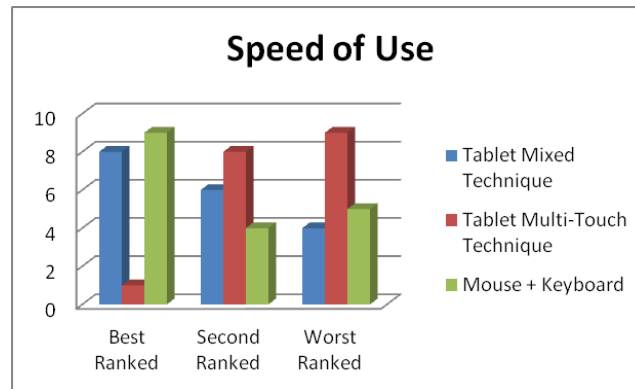


**Figure 28: Comparative graph between the three means on Easy to Use feature**

The problem detected using multi-touch technique was that sometimes users forgot the gestures for rotate the polygon. And when users try rotation gestures, in some cases, they rotated a wrong axis. The rotation gestures are not intuitive using this technique.

### 7.2.4. Speed of Use

For the "speed of use" the tablet multi-touch technique has been chosen the worst mean again. But, for the best mean there is no clear difference between keyboard and mouse, and tablet with mixed technique. Both of them got similar number of votes. See figure 29.

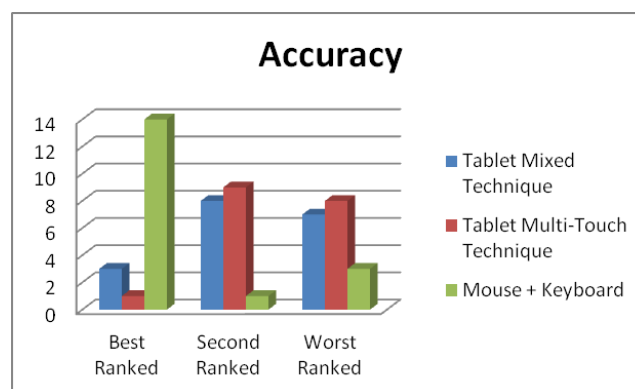


**Figure 29: Comparative graph between the three means on Speed of Use feature**

Whereas participants were running the tasks, we could observe how translation tasks were executed faster using keyboard and mouse. But, rotation tasks were executed faster using the tablet with mixed technique. So probably this is the argument why there is not clear mean for "speed of use".

### 7.2.5. Accuracy

This time, there is a clear best interface for accuracy. Figure 30 leaves no doubt that the best mean to achieve a better accuracy is the keyboard and mouse. Especially in translation and scale tasks.



**Figure 30: Comparative graph between the three means on Accuracy feature**

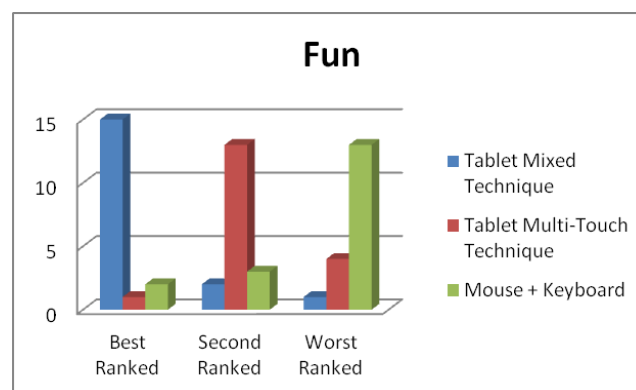
For accuracy feature, tablet techniques have some drawbacks. In the tablet multi-touch technique, when participants want to rotate the overlay in Z-axis, they can easily commit an error. If they do not move the two fingers at the same time in one sense, and the distance between the fingers increases, the algorithm identifies a gesture scale.

On the other hand, in the tablet mixed technique during translation and scale tasks, participants must keep the tablet in the same position, without moving it. Otherwise, the polygon rotates depending on the movement detected.

We could observe these two drawbacks during observation when running the tasks.

### 7.2.6. Fun

Users got more fun using the tablet with mixed technique. The second is the tablet with multi-touch technique. And the less funny is the keyboard and mouse. Figure 31 shows these results.



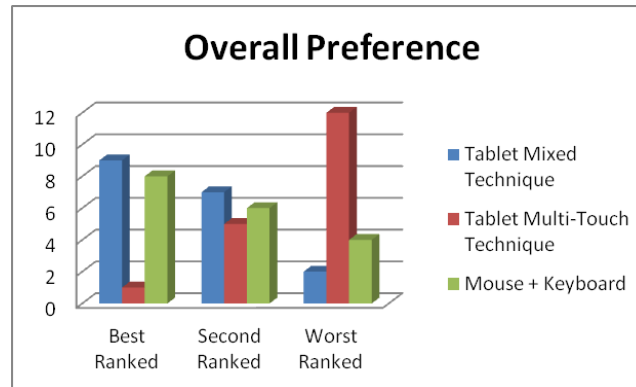
**Figure 31: Comparative graph between the three means on Fun feature**

Different causes of this may be because users have certain level of freedom with tablet. They can move around. However, if they use the keyboard and mouse mean, they must be seated.

On the other hand, the use of new technologies is usually more attractive, and obviously the tablet as a means of entry is newer than the keyboard and mouse.

### 7.2.7. Overall Preference

As overall preference, it is difficult to identify the best technique according to the results shown in the figure 32. Moreover, tablet with the multi-touch technique has been selected as the worst one in general terms.



**Figure 32: Comparative graph between the three means depending on Overall preference**

The less preferred technique is the tablet multi-touch technique, perhaps for the non intuitive interface for rotations.

On the other hand, there is no clear winner in general terms. The votes were distributed almost equally between interfaces, keyboard and mouse, and tablet with mixed technique. Between these two techniques (mouse + keyboard and tablet mixed technique) we have to distinguish different aspects. For translation and scale actions using mouse and keyboard, users fixed with more accuracy the virtual display on the empty space. For rotations tasks some users doubted when clicking the correct axis using the mouse. Instead this rotations using tablet were really intuitive using the mixed technique fo the tablet. One of the biggest advantages of the tablet is the increasing of freedom of motion and immersiveness. Next section shows the benefits and drawbacks of the three means.

### 7.3. Benefits and Drawbacks

Once we have analyzed the results of questionnaires UMUX, comparative questionnaire, reviewed the feedback from participants and collected the notes taken during the execution of tasks, we can identify the following benefits and drawbacks for each mean in table 6:

Mean	Benefits	Drawbacks
<b>Keyboard + Mouse</b>	Very good usability High level of accuracy High speed performing translation tasks Easy to learn Easy to use	Less sense of reality Less fun to use it
<b>Tablet Multi-touch Tech.</b>	Good usability High sense of reality It is fun to use it High speed performing rotation tasks Easy to learn and use	Less degree of accuracy Easy to make unwitting rotation
<b>Tablet Mixed Tech.</b>	High sense of reality It is fun to use it	Difficult to learn how to use it Difficult to use it Less degree of accuracy Slow performing tasks

**Table 6: Benefits and drawbacks of each mean of interaction**

## 8. Discussion

This section portrays a summary of limitations and difficulties identified during the elaboration of this work. The purpose of these interfaces is to help users during designing and prototyping of public displays. It would be interesting to carry out the user study by using participants with certain experience in that field, such as PDS engineers. Because of the difficulty to find around twenty public displays designers, we have replaced them for people with similar educational background. Although we received great comments from the users, we could obtain a wider feedback from experts.

Other limitation has been the time available to conduct this study, mainly for two reasons. First, we should have added another technique of interaction with the tablet in this study. This technique would be the implementation of a method of interaction without touching the tablet screen. Just with motion gestures such as tilt, shake, rotation, or swing. After analyzing the results, the tablet with multi-touch technique is the least desirable. And we found that certain features improved when motion gestures (for rotation actions) were added in mixed technique; as result of their usability. We leave for future work the implementation of this third technique (only motion gestures) as well as demonstrating if the performance can be improved.

The second reason is that we could update or modify the interfaces implementation depending on the feedback from participants. For instance, two participants suggest in their comments that multi-touch gestures should be less sensitive. This could lead to commit errors, mainly when users try to scale or rotate in Z-axis (using multi-touch technique). With more time, we could implement a second version of the software taking into account suggestions from users and even our observations.

One of the difficulties found in the laboratory is that there are several electronic components. And some of them contain magnets. These magnets distort the signal received by the magnetic sensor from the android device. Therefore, rotation tasks using the tablet with mixed technique were affected. Perhaps, carrying out the study in a laboratory with less electronic components our results could be different.



## 9. Conclusions

This work discusses on the increase of immersive technologies in our present society. Our approach aims at providing a greater degree of immersiveness in the world of public displays, specifically using the IPED Toolkit. The objective is to increase the immersive experience using the tablet mean during prototyping public displays inside the lab, maintaining certain usability.

By using the prototype designed for tablets with multi-touch techniques, we are able to increase the sense of reality when designing public displays. On the other hand it is necessary to acknowledge that it is difficult to learn how to use it. This technique is less accurate and performs the task more slowly. In general terms the degree of usability is low.

The tablet with the second interface developed (mixed technique) increases the sense of immersion more than with the multi-touch technique. However, this technique in terms of usability is acceptable. Its usage is easy to learn and fun. But it has two drawbacks. First, users can make unwitting rotations during translation or scale tasks if they do not hold well the tablet. Users could overcome this problem with previous training. Other option could be to implement a second version of the interface. The goal of this version would be to disable rotation sensors during the execution of these tasks. The second drawback concerns its accuracy. It is difficult to achieve the accuracy provided by the keyboard and mouse interface.

The main conclusion of this study is that if users want to increase the feeling of the real world inside the lab, they have the option to use a new input device, the tablet with mixed technique. This mean has a good degree of usability. But in terms of accuracy, keyboard and mouse are better.

So users must choose which interface to use according to their preference or type of tasks to execute. If they really need to run accurate tasks, they should use keyboard and mouse. But, if their need is to feel the environment and get immersed in a real scenario, then they have the option to use the tablet with mixed technique.

## 10. Further work

Following the same way of means of interaction with the IPED Toolkit, there are other ways of interaction that would be interesting to study:

- Tablet full spatial gestures: this is the technique that we have unable to develop due to time restrictions. It would be interesting to develop as it offers another way of interaction without touching the tablet screen, only using spatial gestures.
- Body gestures (Kinect): Another good option would be to develop an interface that recognizes body gestures. For this interaction method, users do not need any additional device. A Kinect camera would be responsible for carrying out the work.
- Voice command recognition: As previously, users do not need additional device. With their own voice and the help of a microphone could perform all necessary tasks.

## 11. References

- M. Tscheligi A. Geven, J. Schrammel. 2006. Modelling and Designing User Assistance in Intelligent Environments (MODIE 2006). In *MODIE Workshop at MobileHCI '06*. 1–6.
- John Brooke. 1996. SUS-A quick and dirty usability scale. *Usability Eval. Ind.* (1996).
- ECMA. 2013. ECMA-404 The JSON Data Interchange Format. , October (2013).
- Kraig Finstad. 2010. The Usability Metric for User Experience. *Interact. Comput.* 22, 5 (September 2010), 323–327.  
DOI:<http://dx.doi.org/10.1016/j.intcom.2010.04.004>
- Fathi Hamhoun and Christian Kray. 2011. Supporting pilgrims in navigating densely crowded religious sites. *Pers. Ubiquitous Comput.* 16, 8 (October 2011), 1013–1023. DOI:<http://dx.doi.org/10.1007/s00779-011-0461-6>
- Michael D. Harrison and Mieke Massink. 2010. Modelling Interactive Experience, Function and Performance in Ubiquitous Systems. *Electron. Notes Theor. Comput. Sci.* 261 (February 2010), 23–42.  
DOI:<http://dx.doi.org/10.1016/j.entcs.2010.01.004>
- Bin Jiang and Xiaobai Yao. 2006. Location-based services and GIS in perspective. *Comput. Environ. Urban Syst.* 30, 6 (November 2006), 712–725.  
DOI:<http://dx.doi.org/10.1016/j.compenvurbsys.2006.02.003>
- DR Olsen Jr. 2007. Evaluating user interface systems research. ... *20th Annu. ACM Symp. User interface ...* (2007).
- Ji-sun Kim, Virginia Tech, and Francis Quek. 2009. iPhone / iPod Touch as Input Devices for Navigation in Immersive Virtual Environments. , July 2007 (2009), 261–262.

- Jurek Kirakowski. 2000. Questionnaires in Usability Engineering FAQ. (2000). Retrieved November 20, 2014 from <http://www.ucc.ie/hfrg/resources/qfaq1.html>
- Kraig Finstad. The Usability Metric for User Experience. In *Interacting with Computers*, 22 (2010), 323-327.
- Christian Kray, Daniel Nesbitt, John Dawson, and Michael Rohs. 2010. User-defined gestures for connecting mobile phones, public displays, and tabletops. *Proc. 12th ...* (2010), 239–248.
- Ekaterina Kurdyukova, Matthias Redlin, and Elisabeth André. 2012. Studying user-defined iPad gestures for interaction in multi-display environment. *Proc. 2012 ACM Int. Conf. Intell. User Interfaces - IUI '12* (2012), 93.  
DOI:<http://dx.doi.org/10.1145/2166966.2166984>
- J. Lazar, JH Feng, and Harry Hochheiser. 2010. *Research methods in human-computer interaction*,
- James R. Lewis, Brian S. Utesch, and Deborah E. Maher. 2013. UMUX-LITE – When There 's No Time for the SUS. (2013), 2099–2102.
- Nicolai Marquardt. 2011. The Proximity Toolkit : Prototyping Proxemic Interactions in Ubiquitous Computing Ecologies. (2011), 315–325.
- Mark R. Mine, Frederick P. Brooks Jr, and The Problem. 1997. Moving Objects in Space : Exploiting Proprioception In Virtual-Environment Interaction 1  
MANIPULATION IN A VIRTUAL WORLD : WHY IS IT HARD ? (1997).
- Yasuto Nakanishi. 2012. Virtual prototyping using miniature model and visualization for interactive public displays. *Proc. Des. Interact. Syst. Conf. - DIS '12* (2012), 458. DOI:<http://dx.doi.org/10.1145/2317956.2318024>
- Morin Ostkamp and Christian Kray. 2014. Supporting design, prototyping, and evaluation of public display systems. *Proc. 2014 ACM SIGCHI Symp. Eng. Interact. Comput. Syst. - EICS '14* (2014), 263–272.  
DOI:<http://dx.doi.org/10.1145/2607023.2607035>

- Pushpendra Singh, HN Ha, Patrick Olivier, Christian Kray, and Zhiwen Kuang. 2006. Rapid prototyping and evaluation of intelligent environments using immersive video. In *MODIE Workshop at MobileHCI '06*. 36–41.
- Christoph Stahl and Jens Haupt. 2006. Simulating and Evaluating Public Situated Displays in Virtual Environment Models. In *MODIE Workshop at MobileHCI '06*. 32–35.
- O. Stefani and I. Karaseitanidis. 2004. Designing 3D input devices for immersive environments. *2004 IEEE Int. Conf. Syst. Man Cybern. (IEEE Cat. No.04CH37583)* 7 (2004), 6280–6285.  
DOI:<http://dx.doi.org/10.1109/ICSMC.2004.1401385>
- M. Weiser. 1993. Ubiquitous computing. *Computer (Long. Beach. Calif.)* , 804 (1993).

## 12. Appendix A. Questionnaires.

### Participant Information

1. Participant Number:

\_\_\_\_\_

2. Did you read, understand, and sign the informed consent form for participate in the study?

*Tick all that apply.*

Yes

No

3. Are you familiar with computers and smartphones or tablets?

*Tick all that apply.*

Yes

No

4. Gender

*Tick all that apply.*

Male

Female

NA

5. Age:

\_\_\_\_\_

6. Occupation:

\_\_\_\_\_

7. Highest completed education:

\_\_\_\_\_

8. If you have vision problems, please provide some details:

\_\_\_\_\_

9. Dominant hand:

*Tick all that apply.*

Left

Right

## User Study

You should answer the corresponding questionnaire after use each technique

\*Required

### Mouse + Keyboard User Experience

---

1. This system's capabilities meet my requirements.

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

2. Using this system is a frustrating experience.

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

3. This system is easy to use.

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

4. I have to spend too much time correcting things with this system.

Mark only one oval.

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

5. Comments

---

---

---

---

---

## Tablet Multi-Touch Technique User Experience

---

6. This system's capabilities meet my requirements.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

7. Using this system is a frustrating experience.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

8. This system is easy to use. \*

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

9. I have to spend too much time correcting things with this system.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

10. Comments

---

---

---

---

---



## Tablet Mixed Technique User Experience

---

11. This system's capabilities meet my requirements.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

12. Using this system is a frustrating experience.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

13. This system is easy to use.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

14. I have to spend too much time correcting things with this system.

*Mark only one oval.*

	1	2	3	4	5	6	7	
Strongly Disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Strongly Agree

15. Comments

---

---

---

---

---

## Comparative Questionnaire

Please rank the three techniques you experiences from along the following dimensions:

- Best technique
- Second best technique
- Worst technique

### Feeling of being in the real location

---

Which technique increases the feeling to be in the real location

**1. Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**2. Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**3. Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

### Easy to Learn

---

It is easy to learn to use it

**4. Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**5. Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**6. Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

## Easy To Use

---

It is easy to use it

**7. Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**8. Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**9. Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

## Speed of Use

---

I complete the tasks faster

**10. Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**11. Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**12. Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

## Accuracy

---

I complete the tasks with more precision

13. **Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

14. **Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

15. **Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

## Fun

---

It is fun to use it

16. **Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

17. **Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

18. **Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

## Select Your Overall Preference

---

I prefer in general terms

**19. Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**20. Second Best Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

**21. Worst Technique**

*Mark only one oval.*

- Mouse + Keyboard
- Tablet Multi-Touch Technique
- Tablet Mixed Technique

# 13. Appendix B. Participant Consent Form

## BENEFITS AND DRAWBACKS OF DIFFERENT MEANS OF INTERACTION FOR OVERLAY OBJECTS IN 3D WEB SCENARIO

### STUDY

The goal of this study is to make a comparison between different interfaces to place an object in a specified position above the video footage. Participants will use mouse and keyboard, and tablet with two manners of use to locate a polygon inside a frame. Is not necessary to get accuracy placing the object, we are not measuring precision and timing neither, just we want to know if the user is able to carry out the task and the effort spent.

After run the task with each mean, the participants will answer a questionnaire. At the end of all tasks, last comparative questionnaire must be filled.

### REQUIREMENTS

The main requirement for participate in this study is to be familiar with computers and smart phones or tablets.

### CONSENT TO PARTICIPATE IN THE USER STUDY

1. I am willing to run the task and answer the questionnaires.
2. I have been informed of the study goals and aim by Javi Morata, which is the responsible to carry out the user study. I feel that I have been adequately informed.
3. I have been informed:
  - a. That I am participating in user study, which has as its goal to compare different interfaces to place an object in a 3D web scenario.
  - b. Of the purpose of my participation.
  - c. That I am in no way obligated to answer the questions that are asked.
  - d. That my personal data are protected by the <Federal Data Protection Act> which means that my identity will not be disclosed to the public and that no conclusion can be drawn as to my identity from published data. Data provided by me will only be used in the iPED Toolkit project and for scientific purposes. Data concerning my identity will be deleted after termination of the iPED Toolkit project or stored in an anonymous way.
  - e. I am allowed to stop at any time without having to provide any rationale and without having to suffer any negative effects.
  - f. I have the right to have my data deleted at any time and a reference to data protection laws.
4. I can contact Javi Morata, whose email address is ataromivaj@gmail.com, if I have any questions about the project or my participation.
5. I have been informed that I will not receive any payment for my participation.

Münster \_\_\_\_ th December 2015

Name:

Signature

Javi Morata



Masters  
Program  
in **Geospatial  
Technologies**

