Senior Project CIS 4911-U01

Multi-Touch and Mid-Air Framework

Visualizer

Feasibility Study and Project Plan

Team Members:

Richard A. Lopez

April 30, 2015

Mentor:

Francisco Ortega

Professor:

Masoud Sadjadi

# Abstract

This document outlines the feasibility of creating an API for the use of developers who want to create applications that use any type of input devices, focusing mainly on touch screens and mid-air inputs. Also covered on this document is the plan that we are to follow when developing this API and the necessary objects needed in place such as software or hardware objects.

This document will go over the current systems in place that are not meeting the expectations that our API will meet. It will describe the purpose of our software system and the alternative solutions that could be put in place to generate the expected functionality. The project plan will delineate the organization of the project, the team member that will take part in the development of the API and the hardware and software requirements for the system to be implemented.

# Table of Contents

[Abstract 2](#_Toc417890503)

[Table of Contents 3](#_Toc417890504)

[1. Introduction 4](#_Toc417890505)

[1.1 Problem definition 4](#_Toc417890506)

[1.2 Background 4](#_Toc417890507)

[1.3 Definitions, Acronyms and Abbreviations 5](#_Toc417890508)

[1.4 Overview of document 5](#_Toc417890509)

[2. Feasibility Study 6](#_Toc417890510)

[2.1 Description of Current System 6](#_Toc417890511)

[2.2 Purpose of New System 6](#_Toc417890512)

[2.3 High-level Definition of User Requirements (must include security/privacy) 7](#_Toc417890513)

[2.4 Alternative Solutions 7](#_Toc417890514)

[2.4.1 Description of Alternatives 7](#_Toc417890515)

[2.4.2 Selection Criteria 7](#_Toc417890516)

[2.4.3 Analysis of Alternatives 8](#_Toc417890517)

[2.5 Recommendations 9](#_Toc417890518)

[3. Project Plan 10](#_Toc417890519)

[3.1 Project Organization 10](#_Toc417890520)

[3.1.1 Project Personnel Organization 10](#_Toc417890521)

[3.1.2 Hardware and Software Resources 10](#_Toc417890522)

[3.2 Identification of Tasks, Milestones and Deliverables 11](#_Toc417890523)

[4. Appendix 15](#_Toc417890524)

[4. Appendix 16](#_Toc417890525)

[4.2 Appendix B - Feasibility Matrix 17](#_Toc417890526)

[4.4 Appendix D - Diary of Meetings 18](#_Toc417890527)

[5. References 24](#_Toc417890528)

# Introduction

Input devices have advanced at an outstanding rate in the last few decades. The introduction of the mouse was one of the factors that brought the personal computer out of the technical user niche and into the regular consumer. Now touch input is in almost every cellphone and readily available to the general public. Newer input forms have been developed since such as multi touch screen, 3D mouse, midair tracking, and others. Up until now developers have had to code their respective games and applications for the individual input devices which make it a time consuming task and it limits the utilization of more than one input device in use at a time.

The Touch Air Motion Framework is trying to create a more encompassing input device library such that developers can simply choose their input devices and code their apps without having to deal with the individuality of each device.

## 1.1 Problem definition

Today’s market is flooded with different input devices and it has become a common problem for app and game developers to code for the individual input devices aside that to code for their own application or game. Most of the time the plugins and internal software of the input device is available for the developers to search through, learn and be able to apply the necessary characteristics that the developers needs the input devise to express. In the worst cases the information is not open source or there is not enough information around hindering the use of these input devices.

It has become cumbersome for some software developers to decide which input devices they can use out of the list that they really want to use. A framework that contains most of the input devices in the market will make is so that more developers can code for more out of the box applications, making the computer science field a more user friendly environment. A centralized input device library can make it so that new generations of developers and students can code for the low cost input devices and create the next generation of human computer interactions.

## Background

The data that a developer needs to receive input from peripheral devices is encoded usually within the firmware on the device. To access this data the developer has to code the specific flags, and services, this is a time consuming process that becomes cumbersome whenever more than one input device is needed for their application. The ability to have a standard format to access the information to that that devices provide would solve this problem and would give developers a more open environment in which they could combine touch screens with virtual reality googles.

Creating an easy to use all encompassing framework with the ability to test the input devices while codding for them will create a new generation of software developers. The inexpensive devices available on the market today can create a whole new generation of young students that will create much better HCI programs and suits.

## 1.3 Definitions, Acronyms and Abbreviations

* API: Application Program Interface
* HCI: Human Computer Interaction
* TAM: Touch and Mid-Air Framework
* WIMP: Windows, Icons, Menus, Pointer

## 1.4 Overview of document

This document will go over the feasibility of implementing a framework for touch and mid-air input devices and the plan for such implementation. First we take a look at current system in place that attempt to solve the problem. We will propose the system that we want to implement and the requirements of such a system. We will take a look at alternative solutions for this problem, the criteria used to analyze the feasibility of this project and we will also present the analysis of the alternatives.

The second half of this project will take a look at the project plan, the organization of the tasks for this project and the hardware and software requirements. Finally we will identify the tasks deliverables and milestones for the development of this solution during this semester. Chapter four is the appendix were all of the supplemental images and tables are placed.

# 2. Feasibility Study

This feasibility study will go over the strengths and weaknesses of the proposed system, opportunities, threats, resources and the necessary elements for the success of the system.

## 2.1 Description of Current System

Currently there is no framework or API that provides an easy to use comprehensive collection of functions and control elements for multiple input devices. Some solutions have been created that allow coding for multiple input devices easily and fast, the problem is that most of this solutions are specific for some game engines or app engines, most are not open source therefore creating a blockade for those the biggest collection of developers which is the open source community. The only active open source solution available today is the Virtual Reality Peripheral Network (VRPN). The VRPN is “a set of classes within a library and a set of servers that are designed to implement a network-transparent interface between the application programs and a set of physical devices used in a virtual reality system.” (Taylor II)

The VRPN suffers from some problems that make it an unqualified candidate for solving the problem at hand. First as the name states this system uses a network in which one computer hosts a Virtual Reality (VR) station that controls the peripheral or input devices. This is problematic since many developers do not want to use a network connection and the speeds of data transfer between devices in a network are very slow when compared to the communications inside an individual computer system. A secondary problem is that the VRPN is codded in C language, this proposes a challenge to game developers that use different languages that provide better graphics and faster processing such as UNITY which is a C++ and C# gaming engine. It is easier to port a C++ framework to other languages such as JAVA and C than to go the other way around not to mention that you still have the versatility of many of the C low level performance.

## 2.2 Purpose of New System

The Touch Mid-Air Framework seeks to create an all-encompassing framework to allow game and app developers the freedom to choose between many input devices available in the market. With this framework the developer can simply select the type of input, the function they want to use to map that input to an output and recognizers to allow for specific motions to be mapped to events such as drawing a question mark to open the help menu.

The main purpose of the system is to shorten the time and get rid of the hassle of having to individually code the communication with input devices.

## 2.3 High-level Definition of User Requirements (must include security/privacy)

The TAM Framework is oriented for the use of software developers that need to utilize input devices and want a better way to connect to such devices. The developer needs to be able to use the following functions.

* Subscribe to the system services to receive input events.
* Test the events
* Obtain the information native to the input devices (axes)
* Control the mapping of the input to output
* Record input for further analysis
* Play back input to automate testing
* Control the functions that analyze input in search for pre stablished gestures

## 2.4 Alternative Solutions

This section describes the alternative solutions that deal with the problem at hand. We will compare and contrast the aspects of these alternative solutions against our proposed solution to try and determine the feasibility of our solution.

### 2.4.1 Description of Alternatives

An alternative to this framework would be to create a standard for all input devises, this standard would be followed by every manufacturer in order for app and gam developers to use these input devices in a simple manner, combine them, and eve generate new forms of input with these conventions.

For all manufacturers to comply to a certain framework it would take a revolution in the hardware and firmware field. Even if this solution was to be implemented it would create a problem for independent creators such as the creators of the OCULUS RIFT VR system. It is important that a solution to the problem at hand be an open source solution to achieve the greatest audience of developers which is the open source community.

### 2.4.2 Selection Criteria

The selection criteria that will be used in the analysis of the feasibility of our software solution are as follows:

Operational Feasibility

Functionality: A description of how well the system would work.

Technical Feasibility

Technology: An assessment of the maturity, availability (or ability to acquire), and desirability of the computer technology needed.

Expertise: An assessment to the technical expertise needed to develop, operate, and maintain the candidate system.

Economic Feasibility

Cost to develop: The cost to implement the solution.

Schedule Feasibility: An assessment of how long the solution will take to design and implement.

### 2.4.3 Analysis of Alternatives

Operational Feasibility:

The TAM framework is oriented for developers, the simplicity of a visualization interface that allows immediate feedback from the input device makes this framework a great solution. Since the implementation of a standard on input devices would cause the industry to revolutionize and change to the new system it is not feasible to create a simple visualizer for every device. If this was to be implemented we will still fall onto the problem of the small differences between de devices as the intellectual properties would create them.

Technical Feasibility:

To implement a standard on input devices would be a challenge to try and convert all of the existing devices to this standard, and it might create many incompatible systems. The change would also take a longer time to take as we know that the USB standard was. This solution deprecates most old input devices which is not something that we want to happen. The TAM framework will work with the most popular input devices in the market, allowing for improvement of a module for devices that follow instead of an overhaul of the entire framework.

Economic Feasibility:

Since the TAM project is open source and it is receiving support via a grant it makes it cheap to create and deploy. The maintenance of the system can easily be done by the open source community making the maintenance cheap or even free. If the alternative solution were to be used it would fall onto each input device manufacturer and creator to pay out of pocket for the change to this new standard. For an independent developer creating a new input device the cost to follow a certain standard may be a block and cause the device to never see the light of day.

Schedule Feasibility:

The TAM framework could be finished in months if the number of people involved is around 5 to 10. The major time factor would be the research into the input devices. This is a key point because that research would need to be done once for every device, the app developer which this software is intended for would not have to do this research which is the main quality of the TAM framework. The acceptance of a standard for input devices would take a few years to implement fully; this is basically a chokepoint on the creative process and the advancement of HCI.

See Appendix C for the feasibility matrix.

## 2.5 Recommendations

The TAM framework is the most suitable solution for the problem at hand; it encompasses all of the needs that app and game developers are facing when it comes to input device manipulation. It is our believe that this framework can be implemented in a short amount of time and with a great deal of expandability to allow for the framework to become part of operating systems in the future and adopted as an indispensable tool for application developers.

# 3. Project Plan

In this section we will go over the project plan, the organization of the tasks that will be needed to be completed for the TAM framework to be developed. The focus this semester is the visualization system, this system is an essential part of the TAM framework, allowing application developers to test the input devices in real time, record the inputs and replay them later on. We will also go over the people involved in this project as well as their roles, we will also go over the requirements to develop this system be them software or hardware. Finally we will go over the milestones, deliverables and tasks that will need to be completed this semester.

## 3.1 Project Organization

This section outlines the management and organization aspect of this project. The implementation of this project is broken down into various requirements depending on the features that this product will exhibit. Each group member was assigned a task along with the necessary deadlines needed to complete the project.

### 3.1.1 Project Personnel Organization

This semester the only team member is Richard A Lopez. He will be responsible for the development of the visualization interface that allows the developer to test touch event in a multi touch screen, replay the events, save them for future use mapping functions.

The software development methodology used for this semester will be Agile Software Development. This will be divided into five Scrums which is an Agile method that focuses on the organizational aspects of the software engineering process. It is currently one of the most popular Agile methods. (Sen. Proj. Class). A detailed schedule can be seen in Appendix A.

### 3.1.2 Hardware and Software Resources

The implementation of this version of the TAM framework has the following requirements:

Hardware:

Touch screen capable computer

Multi touch screen capable computer

1 G of RAM

512 MB Video card

Software:

Windows Operating System 8.1 or greater

Visual Studio 2013 or greater

Qt plugin for Visual studio 5.4 or greater

## 3.2 Identification of Tasks, Milestones and Deliverables

The tasks for this semester are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Week #** | **Date** | **Sprint #** | **Week Day #** | **Class Session Agenda** | **Tasks** |
| **1** | 1/12/2015 |  | **1** | Course Introduction & Proposed Projects | Preparation Tasks |
|  | 1/13/2015 |  | **2** |  | Study Agile Software Engineering |
|  | 1/14/2015 |  | **3** |  | Study Mingle |
|  | 1/15/2015 |  | **4** |  | Study the Proposed Projects |
|  | 1/16/2015 |  | **5** |  | Rank the Proposed Projects |
| **2** | 1/19/2015 |  | **1** | ***MLK (University closed)*** | Accept Project Assignment |
|  | 1/20/2015 |  | **2** |  | Assign Scrum Roles |
|  | 1/21/2015 |  | **3** |  | Collect User Stories |
|  | 1/22/2015 |  | **4** |  | Requirement Analysis |
|  | 1/23/2015 |  | **5** |  | Create Product Backlog |
| **3** | 1/26/2015 |  | **1** | Lecture on Agile & Version Control | Setup Development Environment |
|  | 1/27/2015 |  | **2** |  | Initial Feasibility Study |
|  | 1/28/2015 |  | **3** |  | Initial Project Plan |
|  | 1/29/2015 |  | **4** |  | Initial System Design |
|  | 1/30/2015 |  | **5** |  | Initial Object Design |
| **4** | 2/2/2015 | **1** | **1** | Project Introduction (Session 1) | Sprint 1 planning |
|  | 2/3/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 2/4/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 2/5/2015 |  | **4** |  | Daily Scrum; Implementation and testing. |
|  | 2/6/2015 |  | **5** |  | Daily Scrum; Implementation and testing. |
| **5** | 2/9/2015 |  | **1** | Project Introduction (Session 2) | Daily Scrum; Implementation and testing. |
|  | 2/10/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 2/11/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 2/12/2015 |  | **4** |  | Daily Scrum; Integration and testing. |
|  | 2/13/2015 |  | **5** |  | Sprint 1 review |
| **6** | 2/16/2015 | **2** | **1** | Sprint 1 Report (Session 1) | Sprint 2 planning |
|  | 2/17/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 2/18/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 2/19/2015 |  | **4** |  | Daily Scrum; Implementation and testing. |
|  | 2/20/2015 |  | **5** |  | Daily Scrum; Implementation and testing. |
| **7** | 2/23/2015 |  | **1** | Sprint 1 Report (Session 2) | Daily Scrum; Implementation and testing. |
|  | 2/24/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 2/25/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 2/26/2015 |  | **4** |  | Daily Scrum; Integration and testing. |
|  | 2/27/2015 |  | **5** |  | Sprint 2 review |
| **8** | 3/2/2015 | **3** | **1** | Sprint 2 Report (Session 1) | Sprint 3 planning |
|  | 3/3/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 3/4/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 3/5/2015 |  | **4** |  | Daily Scrum; Implementation and testing. |
|  | 3/6/2015 |  | **5** |  | Daily Scrum; Implementation and testing. |
| **9** | ***Spring Break*** | | | | |
| **10** | 3/16/2015 |  | **1** | Sprint 2 Report (Session 2) | Daily Scrum; Implementation and testing. |
|  | 3/17/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 3/18/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 3/19/2015 |  | **4** |  | Daily Scrum; Integration and testing. |
|  | 3/20/2015 |  | **5** |  | Sprint 3 review |
| **11** | 3/23/2015 | **4** | **1** | Sprint 3 Report (Session 1) | Sprint 4 planning |
|  | 3/24/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 3/25/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 3/26/2015 |  | **4** |  | Daily Scrum; Implementation and testing. |
|  | 3/27/2015 |  | **5** |  | Daily Scrum; Implementation and testing. |
| **12** | 3/30/2015 |  | **1** | Sprint 3 Report (Session 2) | Daily Scrum; Implementation and testing. |
|  | 3/31/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 4/1/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 4/2/2015 |  | **4** |  | Daily Scrum; Integration and testing. |
|  | 4/3/2015 |  | **5** |  | Sprint 4 review |
| **13** | 4/6/2015 | **5** | **1** | Sprint 4 Report (Session 1) | Sprint 5 planning |
|  | 4/7/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 4/8/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 4/9/2015 |  | **4** |  | Daily Scrum; Implementation and testing. |
|  | 4/10/2015 |  | **5** |  | Daily Scrum; Implementation and testing. |
| **14** | 4/13/2015 |  | **1** | Sprint 4 Report (Session 2) | Daily Scrum; Implementation and testing. |
|  | 4/14/2015 |  | **2** |  | Daily Scrum; Implementation and testing. |
|  | 4/15/2015 |  | **3** |  | Daily Scrum; Implementation and testing. |
|  | 4/16/2015 |  | **4** |  | Daily Scrum; Integration and testing. |
|  | 4/17/2015 |  | **5** |  | Sprint 5 review |
| **15** | 4/20/2015 |  | **1** | Sprint 5 Report (Session 1) | Finalize Feasibility Study & Project Planning Document |
|  | 4/21/2015 |  | **2** |  | Finalize Requirement Document |
|  | 4/22/2015 |  | **3** |  | Finalize Design Document |
|  | 4/23/2015 |  | **4** |  | Finalize the "Final Document" |
|  | 4/24/2015 |  | **5** |  | Develop Your Poster |
| **16** | 4/27/2015 |  | **1** | Sprint 5 Report (Session 2) | Develop an Introductory Video |
|  | 4/28/2015 |  | **2** |  | Develop a User Guide Video |
|  | 4/29/2015 |  | **3** |  | Develop an Installation & Maintenance Video |
|  | 4/30/2015 |  | **4** |  | Develop a Shortcoming & Wishlist Video |
|  | 5/1/2015 |  |  |  | Include a link to all the docs and videos |
|  | 5/2/2015 |  |  |  | Submit Your Final Deliverable Zip File |
|  | **5/1/2015** |  | **5** | Final Presenation & Showcase | Present Your Project |
|  |  |  |  |  | Fill Out Surveys |
|  |  |  |  |  | Demo Your Project |

Table 3.2.1

# 4. Appendix

Here are the diagrams, images and chart that are supplementary to the Feasibility Study and Project Plan.

# 4. Appendix

Here are the diagrams, images and chart that are supplementary to the Feasibility Study and Project Plan.

**4.1 Appendix A - Project Schedule (Gantt or PERT chart)**

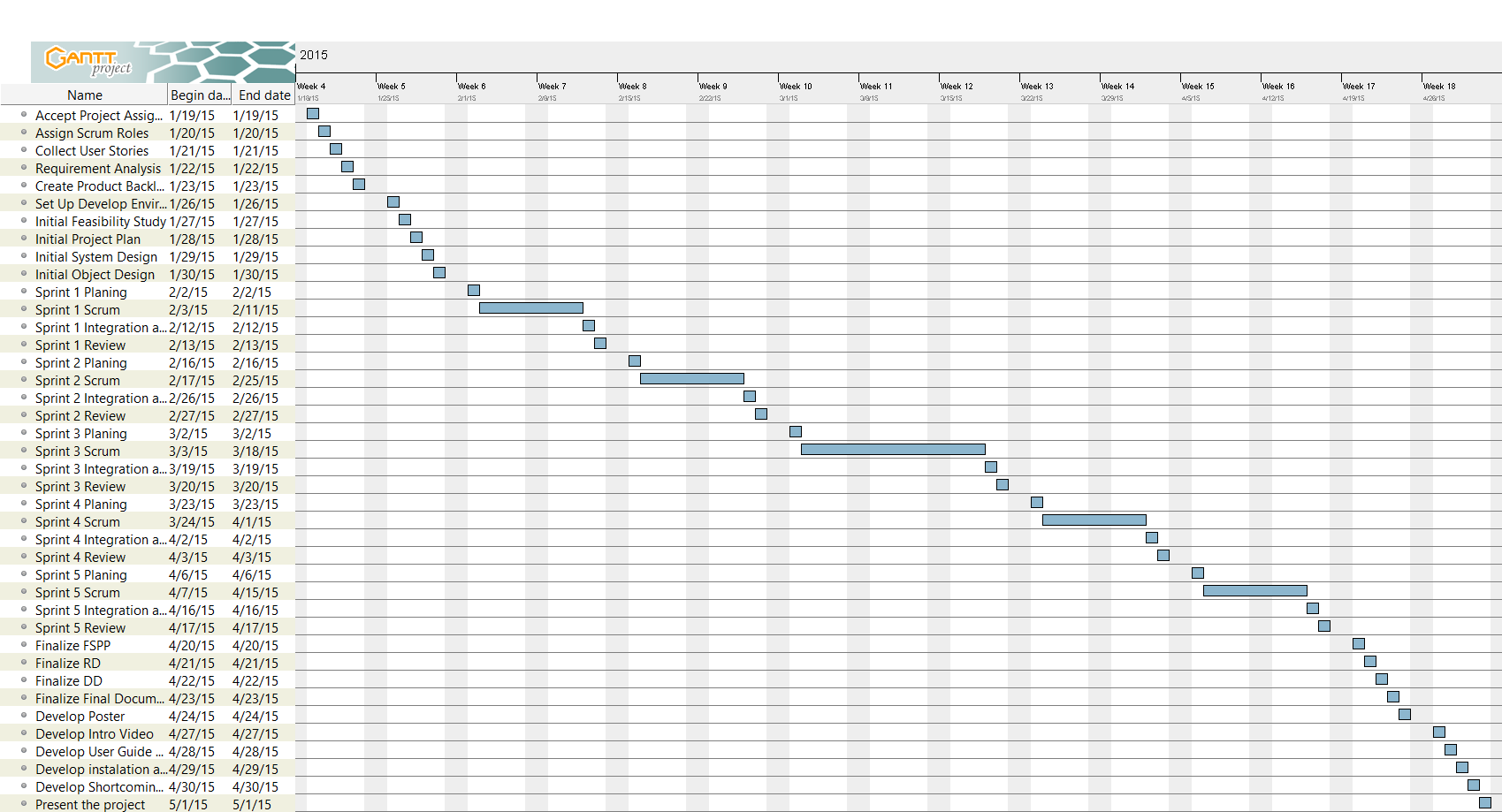


Fig. 4.1.1

## 4.2 Appendix B - Feasibility Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| Feasibility Criteria | Wt. | TAM | Factory Standard |
| Operational Feasibility | **30%** | Oriented for developers, the simplicity of a visualization interface that allows immediate feedback from the input device makes this framework a great solution  **Score: 25** | Cause the industry to revolutionize and change to the new system, small differences may cause compounding problems.  **Score: 10** |
| Technical Feasibility | **30%** | Will work with the most popular input devices in the market, allowing for improvement of a module for devices that follow instead of an overhaul of the entire framework  **Score: 27** | A challenge to try and convert all of the existing devices to this standard, and it might create many incompatible systems.  **Score: 5** |
| Economic Feasibility | **30%** | Open source and it is receiving support via a grant it makes it cheap to create and deploy.  **Score: 30** | Each input device manufacturer and creator to pay out of pocket for the change to this new standard  **Score: 5** |
| Schedule Feasibility | **10%** | Months if the number of people involved is around 5 to 10. The major time factor would be the research into the input devices.  **Score: 8** | A few years to implement fully due to reengineering.  **Score:5** |
| Ranking: | **100%** | **89%** | **25%** |

Table 4.2.1

## 

## 4.4 Appendix D - Diary of Meetings

Meeting 1:

Date: 01/19/15

Participants: Anthony M Amador

Francisco Ortega

Richard A Lopez

Topics discussed:

* More in-depth explanation of the system by Francisco
* Detailed explanation on Francisco expectations on Anthony and Richard
* Background information obtained from Anthony and Richard
* Set up next meeting Tuesday January 20th

Meeting 2:

Date: 01/20/15

Participants: Anthony M Amador

Francisco Ortega

Richard A Lopez

Topics discussed:

* Assigned Scrum roles: Richard: Scrum Master

Anthony: Team Member

* Discussed possible use cases
* Set up next meeting Wednesday January 21st

Meeting 3:

Date: 01/21/15

Participants: Anthony M Amador

Francisco Ortega

Richard A Lopez

Topics discussed:

* Took down all the ideas that Francisco wanted for the system
* Molded the ideas into the format of user stories ”who, what, why”
* Collected all the user stories and placed the ones that seemed unfeasible at the moment
* Set up a face to face meeting to learn C++ for Monday January 26th

Meeting 4:

Date: 01/26/15

Participants: Anthony M Amador

Francisco Ortega

Richard A Lopez

Topics discussed:

* Francisco introduced C++
* Francisco introduced some code for the system
* Anthony and Richard set up reading from C++ book
* Set up next meeting for Wednesday January 28th

Meeting 5:

Date: 01/28/15

Participants: Anthony M Amador

Richard A Lopez

Topics discussed:

* Cleaned up user stories
* Decided on points to be given for all stories
* Decided on stories that could not be developed during this semester
* Split up development of documents and diagrams
* Set up net meeting for Saturday January 31st

Meeting 6:

Date: 01/31/15

Participants: Anthony M Amador

Francisco Ortega

Richard A Lopez

Topics discussed:

* Talked about the task that would be created for each story
* Decided on meeting time for sprint planning for Monday February 2nd
* Split up more work for presentation

Meeting 7:

Date: 02/02/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Talked about the task that would be created for each story
* Decided on the task to be implemented for sprint 1
* Talked about learning Qt
* Created the basic structure of the Git repository

Meeting 8:

Date: 02/13/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Reviewed the work done in sprint one
* Went over the user stories to refine them
* Sent a few user stories to the backlog
* Created a new user story replay
* Learned more C++ from Francisco
* Talked about the user stories for next sprint, all user stories will be carried over

Meeting 9:

Date: 02/27/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Reviewed the work done in sprint two
* Accepted most of the user stories for sprint two
* Choose the user stories for sprint three
* Clarified some questions on Qt and the touch interface
* Learned more C++ from Francisco

Meeting 10:

Date: 03/20/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Reviewed the work done in sprint three
* Accepted all user stories from sprint three
* Re-defined the user stories for the semester due to a teammate leaving
* Choose the user stories for sprint four
* Created new user stories to accommodate for previous defects
* Talked about the mapping and recognizer user stories

Meeting 11:

Date: 04/03/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Reviewed the work done in sprint four
* Accepted all user stories except saving to file from sprint four
* Choose the user stories for sprint five that encompasses all the needs for a visualizer in version one point zero.

Meeting 12:

Date: 04/17/15

Participants: Francisco Ortega

Richard A Lopez

Topics discussed:

* Reviewed the work done in sprint five
* Accepted the final user story
* Talked about the documents and the poster
* Went over the complete visualizer
* All done Now to work on the documents!!

# 5. References

“Agile tutorial for the Senior Project Class”. Web. 26 Apr. 2015. <https://moodle.cis.fiu.edu/v2.8/pluginfile.php/45767/mod\_resource/content/2/Agile\_Tutorial.pdf>

Taylor II, Russell. "Overview." *VRPN*. Web. 26 Apr. 2015. <http://www.cs.unc.edu/Research/vrpn/>.