*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Final Deliverable

Project Title: TAM 4.0 - Multimodal Interactive Paint

**Team Members:**

Eric Aguiar

Jorge Nonell

Christopher Naranjo

Alexander Karpis

**Product Owner(s)**:

Francisco Ortega

**Mentor(s)**:

Francisco Ortega

**Instructor**: Masoud Sadjadi

The MIT License (MIT)

Copyright (c) *2016 Florida International University*

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

***Abstract***

*This document presents the design of our improvements to the Multi Modal Interactive Paint Application. As well as new features added in this version; such as, Microsoft Kinect integration, and early support for a separate Microsoft HoloLens application. Descriptions of the architecture, design, and system patterns that were chosen for both applications are in the sections to follow. Diagrams and testing methodologies that complement the understanding our chosen design are included for the benefit of future developers.*

*Finally, a breakdown of each development sprint, and the subsequent user stories and tasks worked on for Multi Modal Interactive paint are described in detail. Emphasis is placed on the work done during the sprints. And the team’s experience are detailed through the Sprint Review, Sprint Retrospective, and Sprint Planning meetings with the product owner.*

**Table of Contents**

**Introduction** ……………………………………………………………………………………………………………………………….. 5

Current System ………………………………………………………………………………………………………………………... 5

Purpose of New System ……………………………………………………………………………………………………………... 5

**User Stories**

Implemented User Stories ………………………………………………………………………………………………………….. 7

Pending User Stories …………………………………………………………………………………………………………..….. 10

**Project Plan**

Hardware and Software Resources ………………………………………………………………………………………….… 12

Sprints Plan …………………………………………………………………………………………………………………………. 13

*Sprint 1*  …………………………………………………………………………………………………………………………... 13

*Sprint 2*  …………………………………………………………………………………………………………………………... 13

*Sprint 3*  …………………………………………………………………………………………………………………………... 14

*Sprint 4*  …………………………………………………………………………………………………………………………... 15

*Sprint 5*  …………………………………………………………………………………………………………………………... 16

*Sprint 6*  …………………………………………………………………………………………………………………………... 17

*Sprint 7*  …………………………………………………………………………………………………………………………... 18

**System Design**

Architectural Patterns ………………………………………………………………………………………………………….. 20

System and Subsystem Decomposition ……………………………………………………………………………………….… 21

Deployment Diagram ………………………………………………………………………………………………………….…... 22

Design Patterns ……………………………………………………………………………………………………………….….... 22

**System Validation**  …………………………………………………………………………………………………………………….23

**Glossary**  ………………………………………………………………………………………………………………………………….37

**Appendix**  ………………………………………………………………………………………………………………………………….38

Appendix A - UML Diagrams ……………………………………………………………………………………………………. 38

*Static UML Diagrams*  ……………………………………………………………………………………………………….38

*Dynamic UML Diagrams*  …………………………………………………………………………………………………..40

Appendix B - User Interface Design ……………………………………………………………………………………….…... 52

Appendix C - Sprint Review Reports ……………………………………………………………………………………...…… 69

Appendix D - Sprint Retrospective Reports …………………………………………………………………….…………… 74

**References** ……………………………………………………………………………………………………………………...………...80

# Introduction

TAM 4.0 is a software solution to showcase the multi modal functionality of FIU’s OpenHID labs Smart Desk. The smart desk can be equipped with multiple devices that have different input methods. Currently the devices integrated are the Multi Touch Monitor, Tobii EyeX, Intel Real Sense Device, Leap Motion Controller, and Xbox One Kinect. Our goal was to create a painting application to showcase the various devices and their unique functionalities such as hand tracking, facial recognition, eye tracking, and touch tracking. The application experience was designed to be fun and interactive while allowing the users to become familiar with each device and their unique functions. This paint application will also provide smart desk developers with information regarding the various uses of the devices and how to develop an intuitive user interface for device features.

## 

## Current System

Currently there is no unique platform that can take such a range of input devices; which can display the unique functionalities the smart desk will provide with its multi modal device interactions. In order to progress the development of the smart desk, a software solution is needed to test how users may interact with the smart desk. The application also provides an entertaining experience where users can interact with the devices to draw in the paint application.

Additionally, new generations of computer interface devices are always being released. The *OpenHID* lab needed to explore the capabilities of these devices and potentially re-architect the current application to support them.

## 

## Purpose of New System

The purpose of the new application is to resolve the limitations of the previous version in regards to application performance and future maintainability. As well as adding support for Microsoft Kinect.

In order to provide application performance and maintainability, we had to analyze and implement appropriate design/architecture principles into the paint program. This would allow for easier understanding of the project, debugging and ability to add features quicker and more efficiently.

Due to time limitations, we decided to not add support for Microsoft HoloLens into the main application. Instead, under the guidance of our product owner, we built a standalone painting application to explore the capabilities and features of the HoloLens. By leveraging the HoloLens’ sensors, the user will be to color and manipulate holograms in the real world with their sight, voice, and gestures. And finally, a simple development guide will be made to quickly get future developers up to speed with how to use sight, voice, and gesture capture with the HoloLens.

# User Stories

The following section provides the detailed user stories that were implemented in this iteration of the TAM 4.0 - Multimodal Interactive Paint project. These user stories served as the basis for the implementation of our improvements to the project’s features. This section also shows the user stories that are to be considered for future development.

## Implemented User Stories

Refer to Sprint planning for additional details such as actual user story, tasks, as well as acceptance criteria and a brief description.

### Eric Aguiar

**User Story # 770 Refactor UI in existing project code**

* **As a developer I** **would like to** Refactor and redesign previous project code **so that I** can easier understand and add features in the future

**User Story # 782 Refactor Device in existing project code**

* **As a developer I would like to** Refactor and redesign previous project code **so that I** can easier understand and add features in the future

**User Story # 801 Implement responsive application window**

* **As a User I would like to** have the ability to open the application into a windowed view **so that I** can choose the size of the window and whether to have the application in fullscreen.

**User Story # 824 Fix bugs with responsive application window**

* **As a User I would like to** not run into bugs while resizing the application window **so that I** can resize the app without any frustration

**User Story # 830 Begin redesigning UI to be object oriented**

* **As a Developer I would like to** have the ability to work with an object oriented UI **so that I** can easily create new UI features and fix UI bugs.

**User Story # 867 Redesign Layer Visualization Feature**

* **As a Developer I would like to** have the ability to work with an object oriented Layer Visualization Feature **so that I** can take advantage of reusable objects for future UI features and to easier debug any future bugs.

**User Story # 896 Redesign Layer Alpha Feature**

* **As a Developer I would like to** have the ability to work with an object oriented Layer Alpha Feature **so that I can** take advantage of reusable objects for future UI features and to easier debug any future bugs.

**User Story # 897 Fix bug with Layer Visualization Feature**

* **As a User I would like to** not run into bugs while using the Layer Visualization Menu **so that I can** see and toggle between layers without any frustration

### 

### Jorge Nonell

**User Story # 780 Refactor Drawing in existing project code**

* **As a developer I would like to** Refactor and redesign previous project code **so that I** can easier understand and add features in the future

**User Story # 781 Improve Build System**

* **As a developer I would like to** improve build system **so that** it takes less time and effort to build going forward as we work with the project.

**User Story # 812 Implement unified drawing system for devices**

* **As a Developer I would like to** be able to interpret input from devices and have the application draw that input without having to know how drawing is implemented in this application **so that I** focus solely on my device and this creates a separation of concerns in the application where all drawing is handle in one location and all input is handled in another.

**User Story # 847 Finalize New Drawing System**

* **As a developer I would like to** have the new drawing system fully work, **so that I can** use this system without bugs, or errors

**User Story # 848 Reimplemented Layering System to encapsulated framebuffer**

* **As a developer** who is simply handling input from a device **I would like to** not have to worry about how graphical rendering is handled in this application, **so that I can** focus on my job of handling input and code is more straightforward.

**User Story # 882 Reimplemented Menu system to add evening and callbacks**

* **As a developer I want to** be able to add menus that contain how they should be drawn and how they should behave inside of them **in order to** be able to separate out the responsibilities of each menu.

### Christopher Naranjo

**User Story # 771 Learn to use AR device**

* **As a developer I would** like to understand how AR devices work **so that I** can determine how they can enhance user interactions with the paint program.

**User Story # 773 Study NDK to integrate AR**

* **As a developer I want** to study NDK(C++) **so that I** can more easily integrate AR features into the main painting application

**User Story # 797 Setup Augmented Reality** **SDKs and development environment**

* **As a developer I want** to be able to developer for both the HoloLens and Moverio, so that I can create an effective AR UX

**User Story # 799 Add spatial mapping**

* **As a developer I would like** to use HoloLens’ spatial mapping capabilities to generate a 3d mesh that the user can draw and place holograms on.

**User Story # 800 Move Holograms with gaze and gestures**

* **As a user I would like** to be able able to interact with holograms, **so that I** can place them anywhere in the world

**User Story # 818 Add a gaze cursor**

* **As a user I wish to** have a cursor that follows my gaze **so that I** can accurately select objects

**User Story # 836 Create color picker**

* **As a developer I want** the user to be able to choose whatever color they want to paint with, **so that they** have a more natural user experience

**User Story # 837 Add basic holographic GUI**

* **As a user I want** a basic GUI **so that I** can control various functions within the application, and be informed of various information.

**User Story # 886 Add voice commands for HoloLens control**

* **As a user I would like** to use my voice to control various functions within the app, **so that I** can bring up the color picker, and place new canvases in the world.

**User Story # 887 Draw Line on Hologram Canvas**

* **As a user I want** to be able to draw on the world with gestures and gaze

**User Story # 888 Create shader/texture for color picker**

* **As a developer I want** to create a shader for the color picker **so that I** can offload computations onto the GPU and increase performance.

### Alexander Karpis

**User Story #768: Draw Lines with Kinect**

* **As a User I would like to** draw lines using the Kinect device **so I can** draw on the canvas.

**User Story #769: Learn to use the Windows Kinect Device**

* **As a Team Developer I would like to** better understand how the Kinect device works **so I can** understand the user interactions with device for the Paint Program

**User Story #806: Get a response from body basics**

* **As a Team Developer I would like to** better understand the Body Basics sample **so that I can** get a response from the Kinect for painting.

**User Story #807: Get responses from other connect features**

* **As a User I would like to** use features of the Kinect device and get a response **so that we** can use it for the program or others can use it in the future.

**User Story #838: Explain the Code for future developers**

* **As a Team Developer I would like to** explain the project’s overall design and recent changes for new developers.

**User Story #839: Be able to run and understand Speech Basics**

* **As a User I would like to** speak using the Kinect device and understand the response **so that we** can paint.

**User Story # 880: Work with Jorge Nonell to get multiple users to draw on the Kinect**

## Pending User Stories

* User Story #855: Work to get Kinect speech integrated
* **As a user I would like to** speak to the application and have it respond to my commands **so that it can** make interacting with the drawing program easier or more accessible. This will also make the kinect more usable since the user usually has to stand so far away from it.
* User Story #856: Multithread device handlers
* **As a user I would like to** use many devices at the same time **so that I can** gain the benefits of accessibility and interactivity that each device has to offer.
* User Story # 880: Work to get multiple users to draw on the Kinect
* **As a User I would like to** draw lines using the Kinect device with more than one person **so that** more people can draw simultaneously

# Project Plan

This section describes the planning that went into the realization of this project. This project incorporated the agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. This section also describes the components, both software and hardware, chosen for this project.

## Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

### Hardware Resources

* Windows 10 PC
  + Desktops and laptops were used, with at least an Intel Core i5 processor and a fairly decent graphics card
  + For HoloLens emulator support, a desktop and laptop with Intel VT-x (or AMD-V if using an AMD cpu) support is required
    - Additional BIOS support is required to enable VT-x/AMD-V - see your system’s manufacturer for more information
* Acer Touch Screen Display
  + The Touch Screen was one input for our Multi Modal Interactive Paint Application. As well as one of the screens used to develop the app. Computers with built in touchscreens were also used in the development of this project.
* LeapMotion Input Device
  + One of the more developed devices used to read hand gestures through infrared technology
* Tobii EyeX
  + Infrared camera used for gaze tracking in the application
* Intel Realsense Depth Camera
  + Depth camera used for facial recognition and drawing with your hand
* Xbox One Kinect
  + Infrared camera used to detect the presence and motions of users to draw with your hand
* Microsoft HoloLens
  + Augmented/Mixed reality device with multiple infrared cameras and microphones used to capture user sight, voice, and gestures

### Software Resources

* Windows 10
  + Required for Visual Studio
  + Professional or Enterprise edition required for installing Hyper-V
* Visual Studio 2013 or 2015
  + Required IDE (VS2015 for Hololens emulator)
* Unity3D Beta 5.4.0
  + Required for HoloLens development
* Programming Languages
  + C++
    - Language used for main application and many of the SDKs
  + C#
    - Language used for HoloLens and Unity
* OpenGL Framework
  + Used for API rendering of 2D and 3D shapes
* LibCinder
  + OpenGL wrapper to assist with shape drawing and canvas control
* Hyper-V
  + Only hypervisor that is supported by HoloLens emulator
* HoloLens Emulator
  + Provides the needed virtual machine for device emulation
* LibUSB 1.0
  + Used to detect if devices are plugged into a usb port
* LeapMotion SDK
  + LeapMotion gesture detection and finger drawing
* Intel RealSense SDK
  + Facial recognition and hand drawing
* Tobii EyeX SDK
  + Gaze tracking in main application

## 

## 

## Sprints Plan

### Sprint 1

(5/23/2016 - 6/3/2016)

The team members indicated their willingness to work on the following user stories.

**Eric Aguiar**

* **User Story #770: Refactor UI in previous project code**
  + - ***Tasks****:*
      * Group UI code into one namespace
      * Redesign structure
      * Redesign classes and header files
    - ***Acceptance Criteria:***

1. Group UI code into one namespace
2. Redesign structure
3. Redesign classes and header files
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #782: Refactor Device in previous project code**
  + - ***Tasks***:
      * Group Device code into one namespace
      * Redesign structure
      * Redesign classes and header files
    - ***Acceptance Criteria:***

1. Group Device code into one namespace
2. Redesign structure
3. Redesign classes and header files
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge Nonell**

* **User Story #780: Refactor Drawing in previous project code**
  + - ***Tasks****:*
      * Group Drawing code into one namespace
      * Redesign structure
      * Redesign classes and header files
    - ***Acceptance Criteria:***

1. Group Drawing code into one namespace
2. Redesign structure
3. Redesign classes and header files
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #781: Improve Build System**
  + - ***Tasks****:*
      * Get project to build in vs2015
      * Add scripts to make the project build with one click
      * Add dependencies to github
    - ***Acceptance Criteria:***

1. Get project to build in vs2015
2. Add scripts to make the project build with one click
3. Add dependencies to github
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Chris Naranjo**

* **User Story #771: Learn AR devices**
  + - ***Tasks****:*
      * Study Epson Moverio SDK
      * Study HoloLens SDK/Documentation
      * Run demo program on each device
    - ***Acceptance Criteria:***

1. Understand how each device takes user interactions
2. Determine functionality for standalone paint application
3. Determine functionality for integration into main application
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #797: Setup development environment/SDKs**
  + - ***Tasks****:*
      * Install Android SDK
      * Install Epson Moverio SDK
      * Install HoloLens SDK
      * Install HoloLens emulator
    - ***Acceptance Criteria:***

1. Able to deploy and run a demo application on each device’s virtual machine/emulator
2. Each SDK successfully installed
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #773: Study NDK (C++) to eventually add AR** 
  + - ***Tasks****:*
      * Study NDK documentation
      * Determine which AR library to use
    - ***Acceptance Criteria:***

1. Understand NDK development
2. Choose AR library
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Alexander Karpis**

* **User Story #769: Learn to use the Windows Kinect Device**
  + - ***Tasks****:*
      * SDK for Kinect Window device downloaded and device working on laptop
      * Understand how the Kinect functions
      * Come up with ideas for Paint Program
    - ***Acceptance Criteria:***

1. SDK for Kinect Window device downloaded and device working on laptop
2. Understand how the Kinect functions
3. Come up with ideas for Paint Program
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #768: Draw Lines with Kinect**
  + - ***Tasks****:*
      * Test
      * Download and Run the current program
      * Register the Kinect Device on the Computer
      * Sample
      * Initializing the Kinect in the Paint Program
      * Deciding which method to use for connecting the Kinect
      * Find a source for the kinect API
    - ***Acceptance Criteria:***

1. User must be able to draw lines using the Kinect device
2. User interaction while drawing must be smooth and simple
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

### Sprint 2

(6/6/2016 - 6/17/2016)

The team members indicated their willingness to work on the following user stories.

**Eric** **Aguiar**

* **User Story #801 : Implement responsive application window**
  + - ***Tasks****:*
      * Open App in Window and detect size
      * Update devices with new height and width
      * Update menus with new height and width
      * Fix memory issue
    - ***Acceptance Criteria:***

1. Open Application into windowed view
2. Allow menus to scale to size based on the window size
3. Responsive window should not affect user drawings
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge** **Nonell**

* **User Story #812 : Implement unified drawing system for devices**
  + - ***Tasks****:*
      * Design draw event class
      * Implement Illustrator class
      * Interpretation of events draws
      * Draws all shapes
    - ***Acceptance Criteria:***

1. There exists a class to represent the event of drawing
2. There exists a class that handles all drawings and accepts draw events
3. When I pass the illustrator class draw events, it draws
4. The illustrator can draw circles, squares, triangles, and lines
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Chris** **Naranjo**

* **User Story #799 : Add spacial mapping**
  + - ***Tasks****:*
      * Generate 3D mesh
      * Add collision to mesh
      * Gaze cursor collision and rotate parallel to detected planes
    - ***Acceptance Criteria:***

1. Dynamically create a 3D spatial mesh of whatever room the HoloLens is in
2. Allow users to place holograms along this mesh
3. Mesh should only attempt to update on following cases:
   1. User Gesture/Command
   2. User grabs a hologram
      * ***Modeling:***
        + Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #800 : Move holograms with gaze and gestures**
  + - ***Tasks****:*
      * Cursor detection of interactable objects
      * Cursor collision with holograms
    - ***Acceptance Criteria:***

1. Be able to select/grab a hologram and move it around the room
2. Be able to place the hologram selected down the room
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #818 : Add a gaze cursor**
  + - ***Tasks****:*
      * Create cursor texture
      * Have cursor follow user’s gaze
      * Cursor texture changes depending on conditions
    - ***Acceptance Criteria:***

1. Created a cursor texture to represent the following
   1. In-range, Out-of-range, or interactable
2. Must follow user’s gaze
3. Cursor texture must change depending on various conditions
   1. In-range, Out-of-range, or interactable
      * ***Modeling:***
        + Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Alexander Karpis**

* **User Story #806 : Get a response from body basics**
  + - ***Tasks****:*
      * Research the body basics sample
      * Get coordinates from the Body Basics program
      * Document the changes and steps taken
    - ***Acceptance Criteria:***

1. User must be able to gesture and get a response
2. Response is acceptable data
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #807 : Get responses from other connect features**
  + - ***Tasks****:*
      * User must be able to use the Kinect feature and get a response
    - ***Acceptance Criteria:***

1. User must be able to use the Kinect feature and get a response
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

### Sprint 3

(6/20/2016 - 7/1/2016)

The team members indicated their willingness to work on the following user stories.

**Eric Aguiar**

* **User Story #824: Fix remaining bugs with app responsiveness while resizing the**
  + - ***Tasks****:*
      * Fix but with memory which causes pixels to be drawn randomly while resizing
      * Fix bug where the device menu disappears when resized, reappears when updated
      * Fix bug where the app crashes when minimized
      * Initialize the size of the app window to the size of the display
      * Fix issue where current drawings are erased when the window is resized
    - ***Acceptance Criteria:***

1. Fix bug with memory which causes pixels to be drawn randomly while resizing the application
2. Fix any other remaining bugs related to window responsiveness
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #830: Refactor UI to be object oriented**
  + - ***Tasks****:*
      * Begin making menu object oriented
      * Handle the dependencies
      * Extract Multi Touch interaction
    - ***Acceptance Criteria:***

1. Extract all menus from UI and make them object oriented
2. Extract Multi Touch interaction out of UI
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge Nonell**

* **User Story #847: Finalize New Drawing system**
  + - ***Tasks****:*
      * Triangles draw correctly
      * Lines draw correctly without performance issues
      * Eraser is fully functional
      * Undo button is fully functional
    - ***Acceptance Criteria:***

1. Triangles draw correctly
2. Lines draw correctly without performance issues
3. Eraser is fully functional
4. Undo button is fully functional
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #848: Menus and Layers Abstracted out**
  + - ***Tasks****:*
      * Menus drawn outside of device handlers
      * Framebuffer objects no longer need to be handled directly in device handlers
    - ***Acceptance Criteria:***

1. Menus drawn outside of device handlers
2. Framebuffer objects no longer need to be handled directly by programmer, they just need to throw shapes at the illustrator/canvas objects
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Chris** **Naranjo**

* **User Story 836: Create Color Picker**
  + - ***Tasks****:*
      * Display a box that displays the currently chosen color
      * Create a shader or texture that represents the color wheel
      * Create a shader or texture that represents alpha level
    - ***Acceptance Criteria:***

1. Created object to represent color picker
2. Created object that shows currently selected color
3. Must update selected color on tap
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story 837: Create holographic GUI**
  + - ***Tasks****:*
      * Add holographic buttons that can be tapped to change objects
      * Add HUD to display various debug info
    - ***Acceptance Criteria:***

1. Created basic UI buttons that the user can gaze and tap to affect the world
2. Have non-interactable HUD that displays various info on the edge of user’s vision
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Alexander** **Karpis**

* **User Story 838: Explain the Code for future developers**
  + - ***Tasks****:*
      * Examine and explain the overall design
      * Explain individual contributions
    - ***Acceptance Criteria:***

1. Documents that explain the overall design of the project
2. Documents that explain our individual changes of the project
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story 839: Be able to run and understand Speech Basics**
  + - ***Tasks****:*
      * Download correctly
      * Run and understand Speech Basics
      * Include correctly
    - ***Acceptance Criteria:***

1. User must be able to gesture and get a response
2. Response is acceptable data
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

### Sprint 4

(7/4/2016 - 7/15/2016)

The team members indicated their willingness to work on the following user stories.

**Eric** **Aguiar**

* **User Story #867: Redesign Layer Visualization Feature**
  + - ***Tasks****:*
      * Create dropdown when layer visualization is activated with new menu system
      * Draw the layer visualization drop down menu correctly
      * Draw the layer alpha menu correctly
    - ***Acceptance Criteria:***

1. Rewrite to work with new Menu/Canvas system
2. Rewrite Layer Visualization Feature into its own class
3. Works correctly in the same way it used to
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge** **Nonell** **and** **Alexander** **Karpis**

* **User Story # 880: Work together to get multiple users to draw on the Kinect**
  + - ***Tasks****:*
      * Get multiple users to draw on screen
    - ***Acceptance Criteria:***

1. Two users can make the kinect draw simultaneously
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge** **Nonell**

* **User Story #882: Add menu system eventing**
  + - ***Tasks****:*
      * Create callback system
      * Make sure color picker still works
      * Re integrate shape picker menu into new system
    - ***Acceptance Criteria:***

1. Menus now work with callbacks in response to events
2. Color picker menu still works
3. Shape picker menu still works
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Chris** **Naranjo**

* **User Story # 888: Create Shader or texture for color picker**
  + - ***Tasks****:*
      * Create custom shader for the color picker
      * Convert worldspace taps into local space to determine color
    - ***Acceptance Criteria:***

1. Created a shader that represents our color picker
   1. Must control Alpha
   2. Must control hue
2. Allow transformations of taps from world space to object space - determining which color was selected
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story # 886: Add voice commands for HoloLens control**
  + - ***Tasks****:*
      * Add “Close” command
      * Add “Add Object” command
      * Add “Color” command
    - ***Acceptance Criteria:***

1. Add voice command “Colors” to bring up color picker
2. Add voice command “Add Object” to bring up canvas picker
3. Add voice command “Close” to remove the current object that gaze cursor is on
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

### Sprint 5

(7/18/2016 - 7/29/2016)

The team members indicated their willingness to work on the following user stories.

**Eric Aguiar**

* **User Story #896: Redesign Layer Alpha Menu**
  + - ***Tasks****:*
      * The layer alpha indicator should redraw itself based on user's touch
      * A layer's alpha value should change based on where the user moved the alpha indicator
    - ***Acceptance Criteria:***

1. Layer Alpha menu indicator should move based on the user's touch
2. The alpha of the layer should change based on where the user moved the indicator for that layer
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #897: Fix bug with Layer Visualization Menu**
  + - ***Tasks****:*
      * Fix bug where layer visualization feature only displays two layers
      * Fix bug where user can only toggle between two layers on touch
    - ***Acceptance Criteria:***

1. The layer drop down should correctly display each layer
2. The user should be able to toggle between layers by touch
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Jorge** **Nonell and Alexander Karpis**

* **User Story #855: Work together to get Kinect speech integrated**
  + - ***Tasks****:*
      * Name of a supported shape
      * Name of a supported color
      * Fill and unfill shapes
      * Undo
      * Symmetry
      * Figure out how to handle speech input from kinect
    - ***Acceptance Criteria:***

1. When speaking the following commands to the kinect it should respond accordingly
   1. Name of a supported shape (i.e. circle, rectangle, triangle, etc) it changes the current shape I am drawing
   2. Name of supported color (i.e. red, green, etc) it changes the color I am currently painting with.
   3. “Fill shapes” it makes the subsequent shapes I draw filled if they are currently not
   4. “Unfill shapes” it makes the subsequent shapes i draw not filled if they currently are
   5. ”Undo” it undoes the last drawing done
   6. “Symmetry” will turn on the symmetry line if it is off, or turn it off if it is on.
      * ***Modeling:***
        + Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #856: Multithread device handlers**
  + - ***Tasks****:*
      * Research threading in C++
      * Create class to be able to abstract away the idea of threading
      * Each device handler should run on own thread
    - ***Acceptance Criteria:***

1. Each device handler runs on its own thread
2. Process input from multiple devices at the same time doesn’t cause lag
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

**Chris Naranjo**

* **User Story #887: Draw Line on holographic canvas**
  + - ***Tasks****:*
      * Add UV textures for dynamic painting
      * Add sprite paintbrush
      * Capture manipulation gesture into paint strokes
    - ***Acceptance Criteria:***

1. Can paint on objects in the world
2. Manipulation gestures repeatedly ‘stamp’ the sprite texture to simulate a painted line
   * + ***Modeling:***
       - Refer to UML diagrams in Appendix A that were created or modified to model the functionality that will be implemented in this sprint.

* **User Story #916: Create HoloLens development guide**
  + - ***Tasks****:*
      * Create basic HoloLens-unity development guide
    - ***Acceptance Criteria:***

1. Guide shows how to easily capture gaze, gestures, and voice commands in Unity for HoloLens

# System Design

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

## Architectural Patterns

The main Multimodal Interactive Paint Application has a Model View Controller architectural pattern. MVC consists of three kinds of objects. The Model is the application object, the View is its screen presentation, and the Controller defines the way the user interface reacts to user input.

Whereas the augmented reality version has a Component-based design. Encapsulating related functions into independent, reusable, and unique objects. This pattern was chosen so future developers can easily reuse specific specific components of the application if any redesign is needed in further iterations.

## System and Subsystem Decomposition

The Model or main objects of the Multimodal Interactive Paint Application are the event and frame buffers that are given the draw events from Cinder and the other devices.

The View is the cinder application window itself which is drawn on with openGL functionalities.

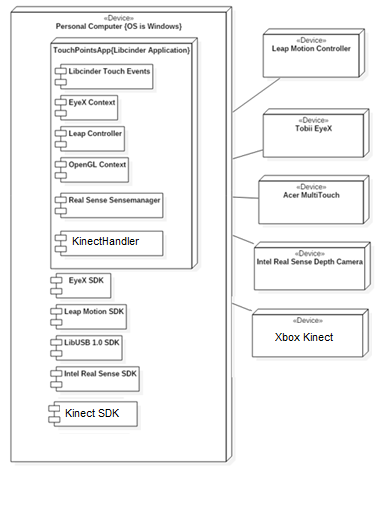
The Controller are the various header files, functions, and variables that control the information between the event buffers and the application.

Mainly, the buffers go to the Illustrator class which sends the events to the TouchShapes class, which determines what shape to draw and how to draw it using the draw method of each shape. All of the other code is used for supporting this process.

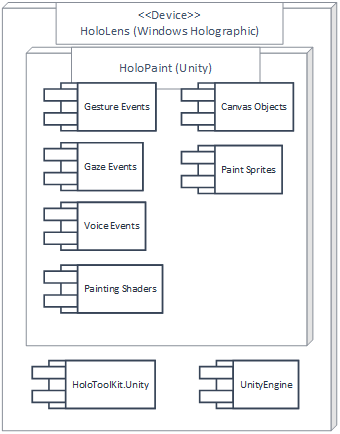
With the Component-based augmented reality application, we split up each unique function into their own components For example, both voice and gesture commands are encapsulated into separate components *VoiceCmdManager* and *NavManipManager* respectively.

## Deployment Diagram

Deployment diagrams display the hardware components of a system, what software components they use, and how the hardware interacts.



*Deployment Diagram for main application*



*Deployment diagram for HoloLens application*

## Design Patterns

With the Multimodal Interactive Paint Application being an integration of devices, features, software libraries, and components, there are many design patterns used in the application. Some are design patterns used are:

* Separations of Concern
* Object Oriented Principles
* Single Responsibility Principle
* Principle of Least Knowledge (Law of Demeter)
* Don’t Repeat Yourself (DRY)
* Layered Architecture Style Composition

Whereas, for the augmented reality application, singleton patterns were widely used to prevent unnecessary object creation. This was used since; for example, only one component that tracks gestures is needed, so this inherits the singleton interface and further objects of this type are prevented from being created at runtime. While at the same time opening its public methods and variables to any other component through the *Object.\_instance.method* call.

# System Validation

Since the project was built upon an existing framework and our teams work focused on only adding a small amount of new subsystems, we primarily used big bang testing and concerned ourselves with sunny and rainy day tests. These tests cases are shown in our videos and other documentation.

## User Story #806 Unit Test

Sunny Day Tests

Test Case: Output from Body Basics

Test Purpose: Ensure that user can get a response from Body Basics..

Test Setup: run program

Test Output: Coordinates of the hands within a range.

Expected Output: After testing, coordinates should be printed to an output file.

## User Story #839 Unit Test

Sunny Day Tests

Test Case: Output from Speech Basics

Test Purpose: Ensure that user can get a response from Speech Basics..

Test Setup: run program

Test Output: A picture moves in response to a command.

Expected Output: After testing, the picture should be moved from the starting point.

## User Story #876 Unit Test

Sunny Day Tests

Test Case: Draw with Kinect

Test Purpose: Ensure that 6 users can draw with the Kinect…

Test Setup: Setup Kinect and run program.

Test Output: Not drawn correctly with the motions of multiple people.

Expected Output: After closing one’s hands with the Kinect, the screen should draw correctly.

## User Story #768 Unit Test

Sunny Day Tests

Test Case: Draw with Kinect

Test Purpose: Ensure that user can draw with the Kinect…

Test Setup: Setup Kinect and run program.

Test Output: Drawn according to motions with the Kinect.

Expected Output: After closing one’s hands with the Kinect, the screen should draw correctly.

// Sprint 1

## User Story #780 Unit Test

Sunny Day Tests

Test Case: Drawing Still Works

Test Purpose: Ensure that user can still draw on screen after drawing is refactored..

Test Setup: Run program

Test Output: Every line drawn correctly

Expected Output: After pressing multitouch screen, the screen should draw

## User Story #781 Unit Test

Sunny Day Tests

Test Case: Build Works

Test Purpose: Ensure that user can still draw on screen after drawing is refactored..

Test Setup:

* Pull down Installers from git repo
* Run installers
* Pull down project code
* Navigate to “Code/TouchPoints/vs2015”
* Pull down dependencies from repo

Test Output: Project builds and runs successfully

Expected Output: After pressing run button screen, the build succeeds and the program runs correctly

## User Story #782 Unit Test

Sunny Day Tests

Test Case: Devices Still Work

Test Purpose: Ensure that user can still use the devices to draw on screen after devices are refactored..

Test Setup: Run program

Test Output: Every line drawn correctly

Expected Output: After testing all integrated devices, i.e. the RealSense, Leap and Eyex can still contribute to drawing, the screen should draw correctly

## User Story #771 Unit Test

Unit Test

Sunny Day Tests

Test case: Open an application.

Test Purpose: Ensure that the user can open and interact with a single application

Test Setup:

1) Bloom gesture to open the start menu if it’s not already open.

2) Air tap an application

3) Use gaze to position application on a surface then air-tap to place and run it

at that location

4) Repeat 1-3 for various applications.

Applications successfully placed on surfaces.

After placing an application on a surface, it should correctly open and remain

Test Output:

Expected Output:

anchored to that surface.

Rainy Day Tests

Test Purpose: Ensure that the user can open, place, and manipulate applications within

the same view.

Test Setup:

Test Output:

Expected Output: To correctly interact with a specific application by centering their gaze over the intended

application.

1) Bloom gesture to open the start menu if it’s not already open.

2) Air tap an application

3) Use gaze to position application on a surface, and air-tap to place and run it.

4) With first application still visible to the user, bloom and open another

application.

5) Repeat 1-4 for various applications

# 

## User Story #801 Unit Test

Sunny Day Tests

Test Case 1: Devices Still Work

Test Purpose: Ensure that user can still use the devices to draw on screen after the window is resized

Test Setup:

⦁ run program

Test Output:

Every line drawn correctly

Expected Output:

After testing all integrated devices, i.e. the RealSense, Leap and Eyex can still contribute to drawing, the screen should draw correctly

Test Case 2: Window is responsive

Test Purpose: Ensure that user can expand the window and have the menus and touch points scale to size.

Test Setup:

⦁ run program

Test Output:

Any shape can be drawn correctly anywhere on the screen and the menus resize according to the size of the window. Works but there is a memory issue with the menus, which causes pixel fragmentation.

Expected Output:

After testing all integrated devices, i.e. the RealSense, Leap and Eyex can still contribute to drawing anywhere on the screen correctly. The menus are responsive to the screen size.

## User Story #812 Unit Test

Sunny Day Tests

Test Case 1: Devices Still Work

Test Purpose: Ensure that user can still use the devices to draw on screen after the window is resized

Test Setup:

⦁ run program

Test Output:

Every line drawn correctly

Expected Output:

After testing all integrated devices, i.e. the RealSense, Leap and Eyex can still contribute to drawing, the screen should draw correctly

## User Story #837 Unit Test

Sunny Day Tests

Test case: Able to remove specific layer, and recover them

Test purpose:to recover them.

Test Setup

User can only remove layers that are active, and tap on the button again

Program Running, with a canvas up.

Test output:

Layer disappears when corresponding button is tapped

Any invisible layer comes back when button is tapped again

Expected output: Layers disappear and reappear correctly

Rainy Day Tests

Test case 1:

Tapping on a button with no corresponding layer or color does nothing

Test purpose

User should not be able to cause the app to arrive at some undefined

state if it tries to hide a non-existent layer or color.

Test Setup

Program Running

Canvas up

Color red not present in the painting

Test output: Nothing happens when user taps the remove red button

Expected output: Nothing happens when user taps the remove red button

Test case 2:

Canvas should not be able to be moved on top of the GUI, or catch the GUI as

the user walks around the room.

Test purpose

User should not be able to get the GUI or canvas holograms stuck on

each other. Since the GUI follows the user around, it is possible for it to ‘catch’ on the

spatial map and get stuck.

Test Setup

Program Running

Canvas up

Test output

hidden behind the canvas.

Expected output

hidden behind the canvas.

User moves around the room with the GUI not catching or becoming

User moves around the room with the GUI not catching or becoming

# Glossary

**Alpha Color -** Fourth color of the 'Red Green Blue' spectrum which determines a colors transparency (Alpha being the inverse of transparency).

**Leap Motion Controller -** Infrared device which detects hand movements. Contains gesture recognition as well as joint detection in 3d space.

**Intel Real Sense Depth Camera -** Infrared device which detects depth in a 3d environment. Contains facial gesture support, currently implemented is a way to draw using your hand.

**Acer Multitouch Monitor -** Currently used multitouch monitor that supports 10 finger touch.

**Lib Cinder -**  Open GL wrapper that makes it easier to draw primitive shapes, as well as handles many useful functions such as image loading, drawing to canvases, writing to frame buffers etc.

**EyeX -** Infrared device developed by Tobii which tracks your eye position as well as eye gaze.

**Shapes** - Shapes implemented by the Interactive Paint Application which can be drawn by the Real Sense, Leap Motion, and Multitouch Screen. Shapes include

* TouchPoints (lines)
* TouchCircles
* TouchRectangles
* TouchTriangles
* TouchEraserPoints
* TouchImage
* TouchTextLayout
* TouchTexture
* TouchVerticalTriangle

**Filled Shapes -** An option that allows any drawing of a TouchCircle, Rectangle, and Triangle to be filled instead of hollow.

**Line Size -** Interactive Paint is capable of manipulating the line size you are drawing with for TouchPoints, as well as unfilled shapes.

**Color -** Various colors we can select to draw using our Application.

**Mode Buttons -** Buttons located at the top left of the User Interface. It contains various buttons for interacting with the application.

**Mode Box -** Box located at the bottom right of our application. It contains an interface to see which shape you are drawing, as well as a way to see which devices are plugged in. It also contains menus to alter device states and a settings menu.

**Layers** - The Canvas is split into three layers. You can draw on specific layers and manipulate their alpha values to create more interesting drawings. You can access the layers by using the layer visualization menu found on the Mode Buttons menu.

**Symmetry Line -** A line of symmetry which reflects any drawing over the axis. Can be toggled in various ways including the mode buttons.

**Undo -** Undoes the previous drawing you did in the currently active layer.

**Device Modes** - A menu which contains the ability to alter which devices are currently active, and what their functionality entails

**Frame Buffer Object -** Data Structure which stores an entire frame of pixels. Used to reduce drawing to only one call on the GPU.

*HoloLens Specific Terms*

**Augmented/Mixed Reality (AR/MR)** - Displaying computer generated images overlaid onto the real world. Can be both interactable and non-interactable.

**Spatial Map** - A 3-dimensional mesh generated by the HoloLens that allows it to determine where it is in 3D space. Allows holograms to interact with the real world as if they were physical objects.

**Gaze/Sight Point** - A small point of ‘light’ about 1m away from the user, between both eyes. Functions as the cursor in augmented world, for more accurate interactions.

**Gesture** - A motion that the hololens is watching for, in order for it to initiate an action.

**Navigation Gesture -** Handles X,Y, Z translation, and ‘tap’ gestures

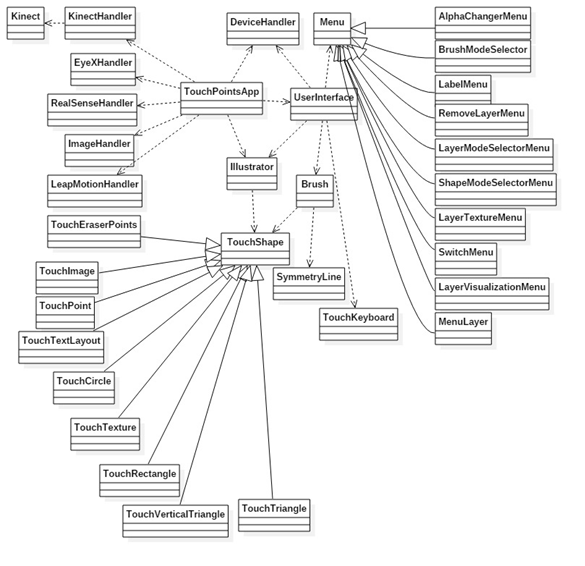
**Manipulation Gesture -** Handles X, Y, Z rotations, and ‘pinch and hold’ gestures

**Voice** - Voice commands that the hololens is always listening for, once heard, will execute an action tied to that phrase

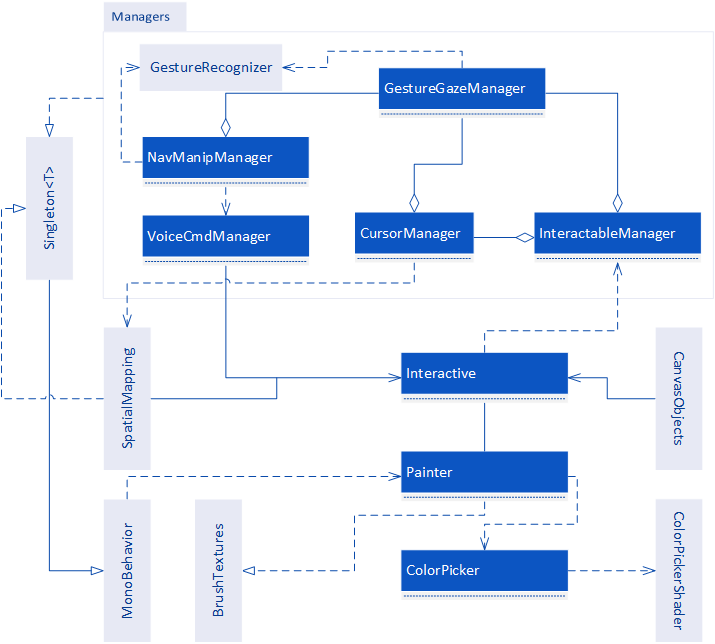
**Shader -** Small compiled program with an algorithm that solely handles the production of colors in an image. In this case, rendering our color picker.

# Appendix

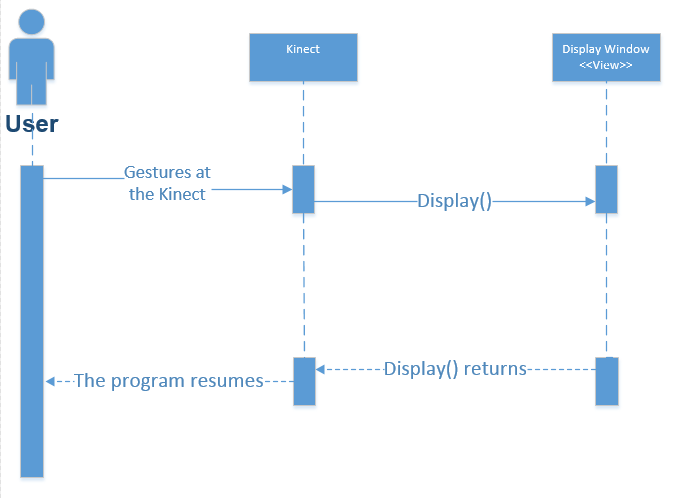
## Appendix A - UML Diagrams

*Minimal Class - Main application*

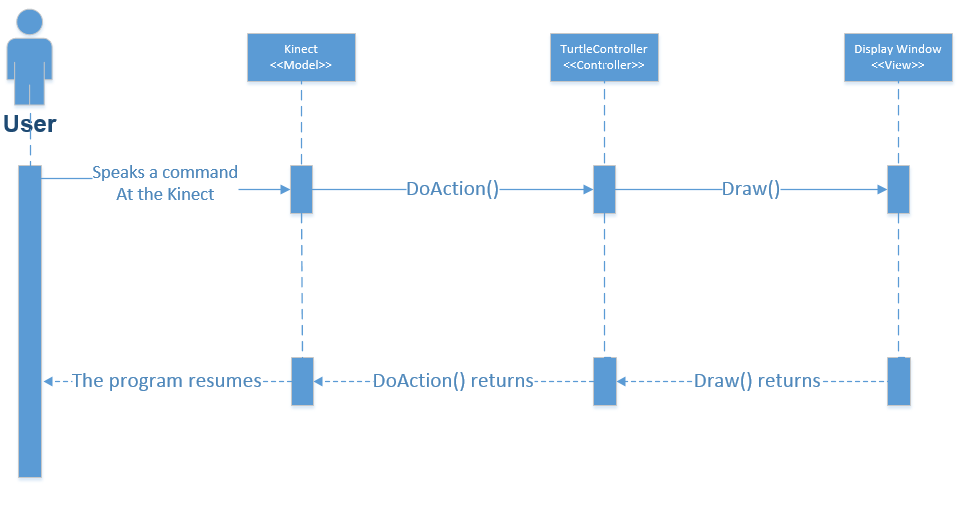
*Minimal Class - HoloLens application*



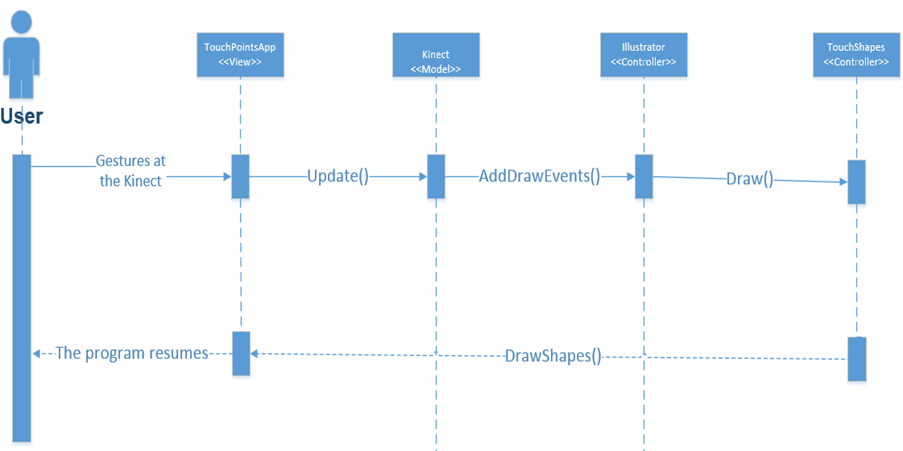
User Story #806



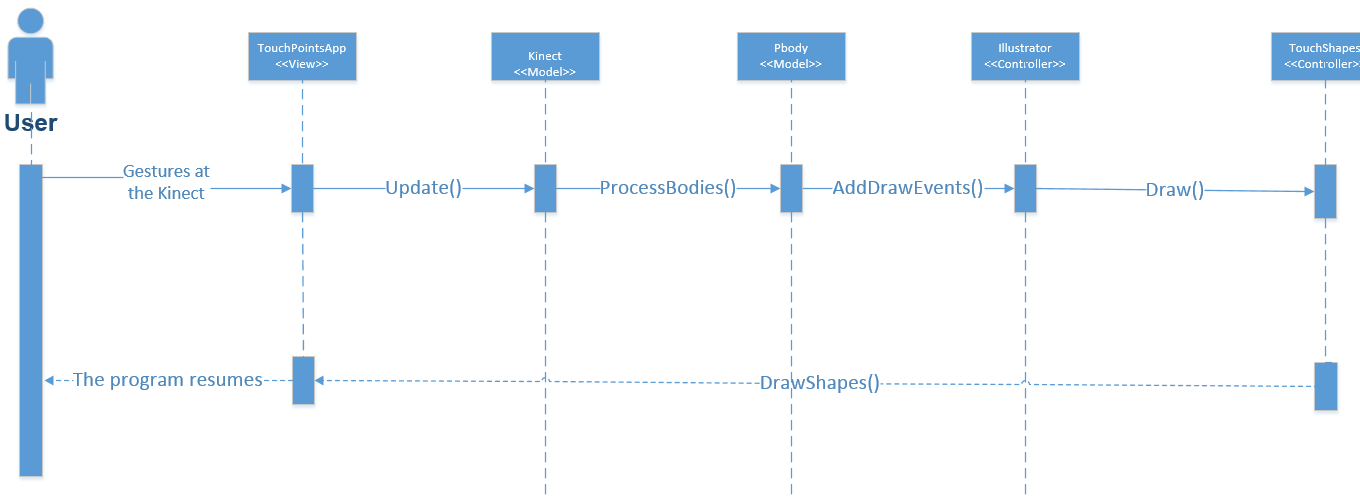
User Story #839



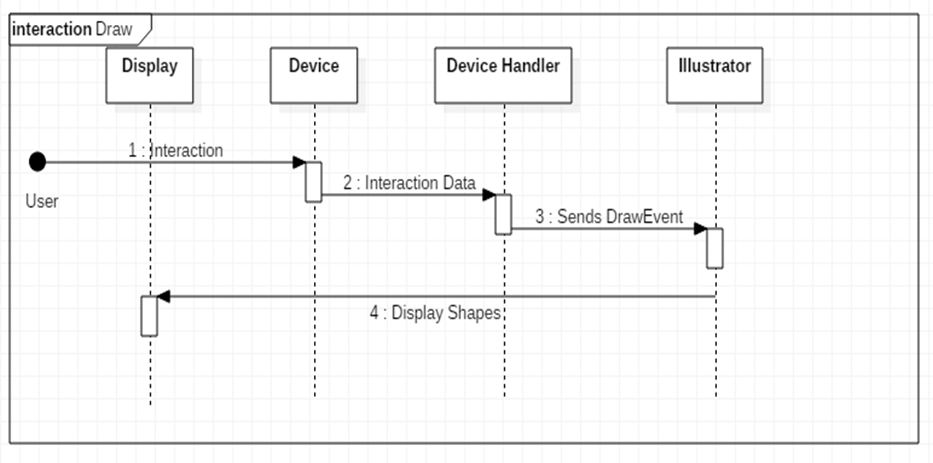
User Story #768



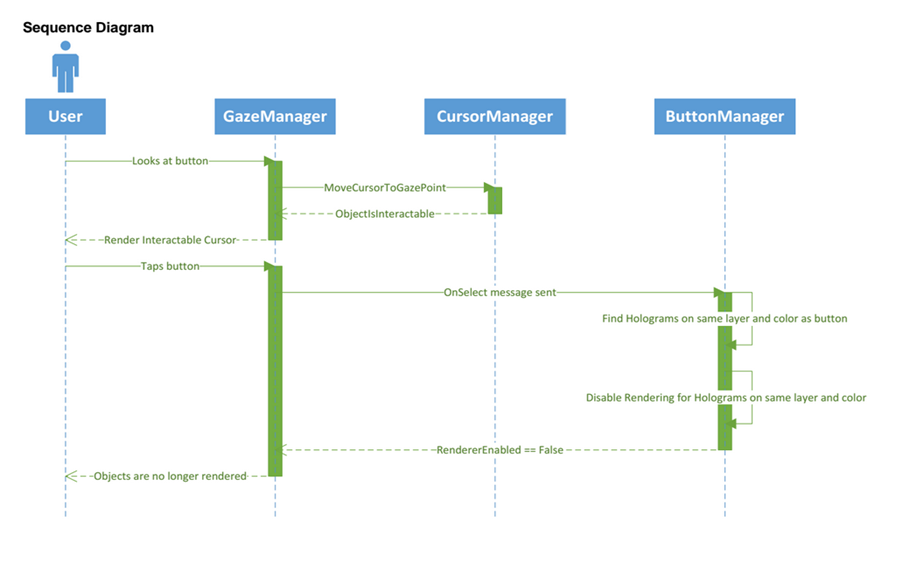
User Story #880



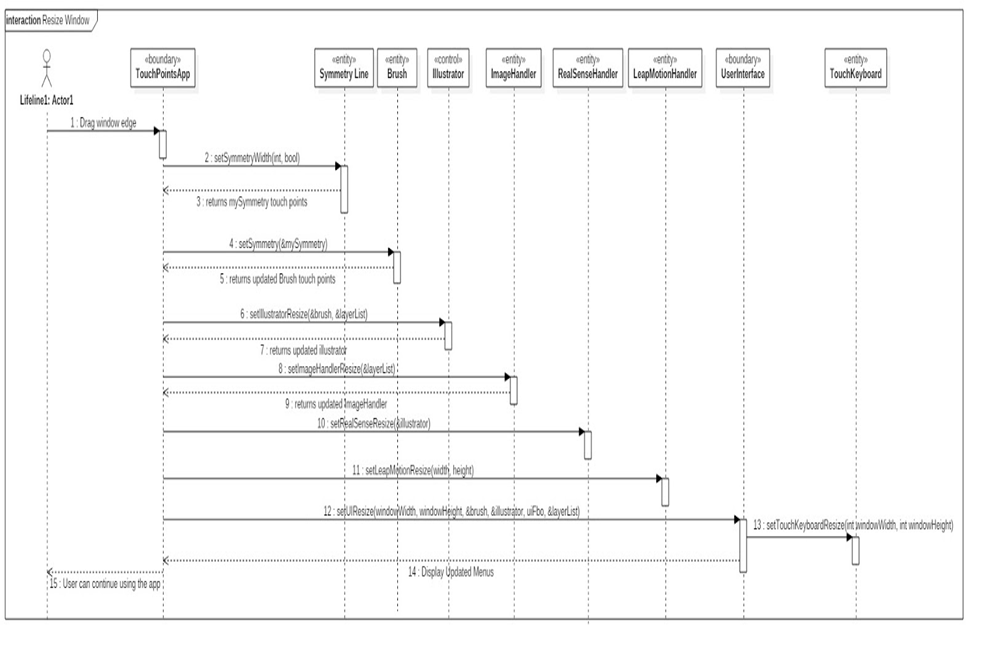
User Story #812



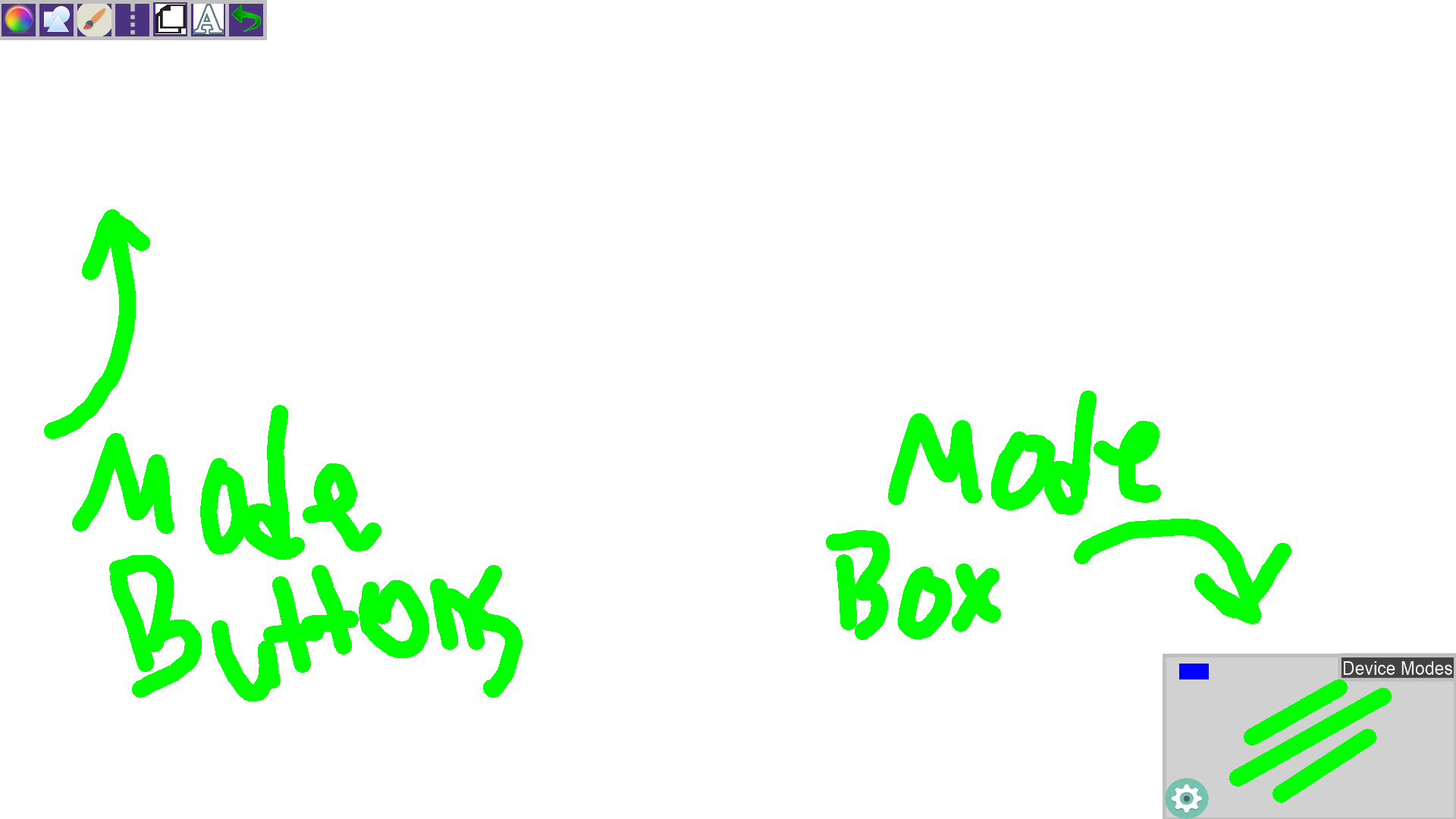
User Story #837



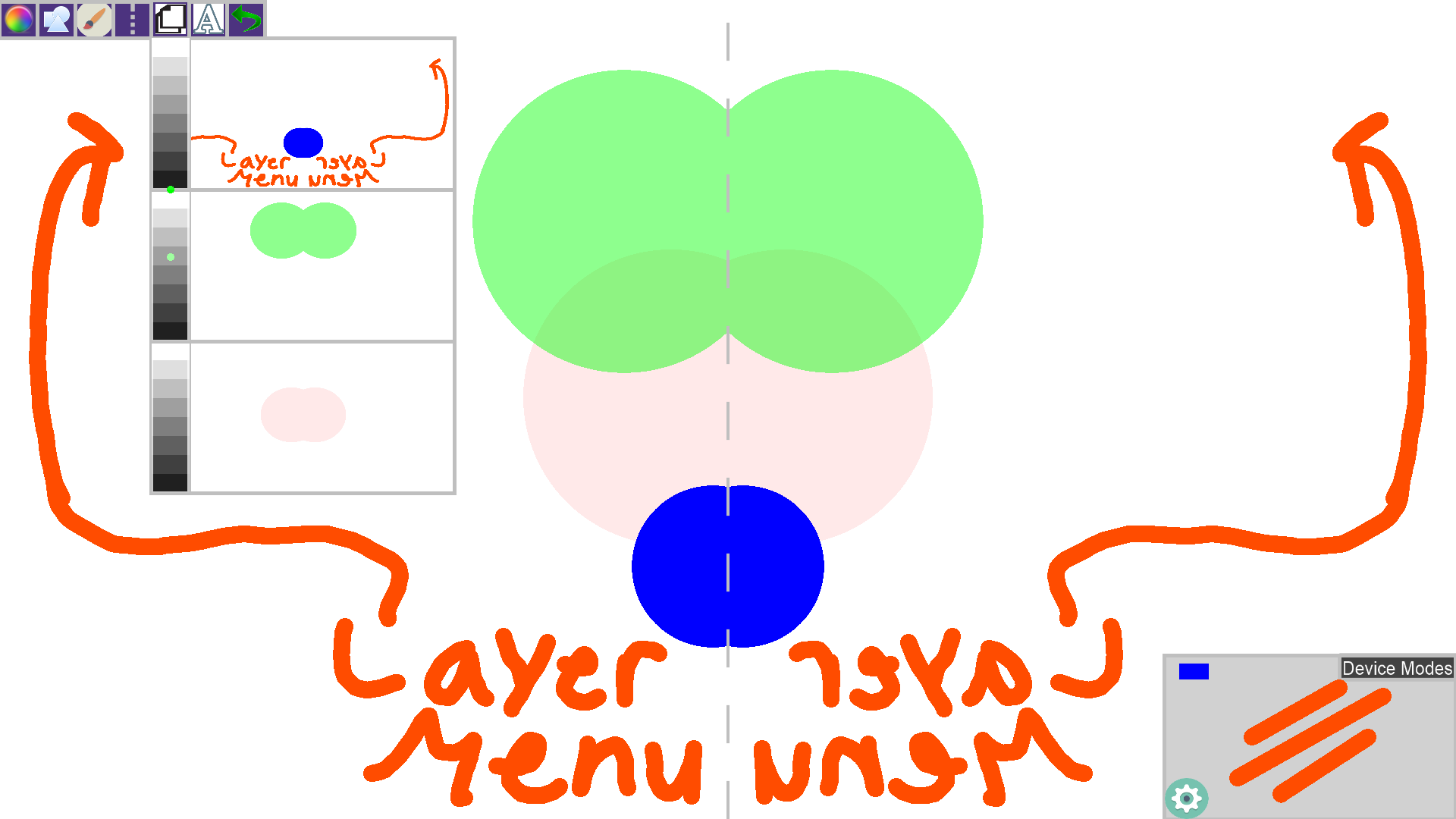
User Story #801



## Appendix B - User Interface Design



Here is our basic user interface for the Multitouch Device. You can see the mode buttons in the top left, as well as the mode box in the bottom right. The mode box shows a blue rectangle indicating that the multitouch device is plugged in and on. You can see there are three green lines, showing us what shape we are drawing and what size.



Here we can see the layer visualization menu pulled up. We can actually see what is drawn on each individual layer. In addition we have an 'Alpha Slider' located on the left side of the layer visualization menu. The green dot indicates how faded the image will appear. As you can see the green is somewhere in the center, indicating that it is semi transparent, the blue is in the dark, indicating it is very visible.

## Appendix C - Sprint Review Reports

...

### Sprint 1

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start time: 6:30

End time: 7:00

Date: 6/3/16

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* *User Story #770 : UI Refactoring*
* *User Story #771 : Learn to use AR devices*
* *User Story #773 : Study NDK to integrate AR*
* *User Story #780 : Drawing Refactoring*
* *User Story #781: Improve Build System*
* *User Story #782 : Device Refactoring*
* *User Story #768, #769 - Learn the Kinect API, Draw Lines with the Kinect.*
* *User Story #797 : Setup AR SDKs and dev environment*

### Sprint 2

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start time: 10:30

End time: 11:30

Date: 7/2/16

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* *User Story #799 : Add Spatial Mapping*
* *User Story #800 : Move holograms with gaze and gestures*
* *User Story #801 : Window resizing and responsiveness*
* *User Story #806 : Get a response from body basics sample on kinect*
* *User Story #811 : Implement and design a unified interface for devices to be able to hand off drawing to the application*
* *User Story #818 : Add cursor that follows user’s gaze*

### Sprint 3

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start time: 10:30

End time: 11:30

Date: 7/2/16

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* *User Story #824 : Fix remaining bugs with application window resizing*
* *User Story #830 : Refactor UI to be object oriented*
* *User Story #836 : Create ColorPicker*
* *User Story #837 : Add basic holographic to GUI*
* *User Story #838 : Explain the code for future developers*
* *User Story #839 : Be able to run and understand speech basics*
* *User Story #847 : Finalize new drawing system*
* *User Story #848 : Menus and layers abstracted out*

### Sprint 4

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start time: 7:30 pm

End time: 8:30 pm

Date: 7/16/16

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* *User Story #867: Work on refactoring the Layer Visualization Feature.*
* *User Story #876: Have the Kinect register multiple people.*
* *User Story #880: Work with Alex to get Kinect speech integrated*
* *User Story #886: Add voice commands for HoloLens control*
* *User Story #888: Create shader/texture for color picker*
* *User Story # : Add menu system eventing*

### Sprint 5

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start time: 7:30 pm

End time: 8:30 pm

Date: 7/30/16

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

* *User Story #867: Work on refactoring the Layer Visualization Feature.*
* *User Story #876: Have the Kinect register multiple people.*
* *User Story #880: Work with Alex to get Kinect speech integrated*
* *User Story #887: Draw line on holographic canvas*
* *User Story #916: Create HoloLens dev guide*
* *User Story # : Add menu system eventing*

## Appendix D - Sprint Retrospective Reports

### Sprint 1

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start Time: 7:00pm

End Time: 7:30pm

Date: 6/3/16

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes, but getting the project upgraded to VS2015 took more time than expected due to dependency issues
* Did each team member work as scheduled?
  + Yes

What went right?

* Communication through skype often to make sure everyone was updated on project changes and current tasks.

How to address the issues in the next sprint?

* How to improve the process?
  + Improve the build system to make it easier to run project
* How to improve the product?
  + Cleaned up the code and organized it to make it easier to add new features and debug in the future

### Sprint 2

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start Time: 6:30pm

End Time: 7:15pm

Date: 6/20/16

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes, but debugging the memory issue took much more time than estimated.
* Did we do a good job estimating the points (time required) for each user story?
  + Yes, but issues with memory were encountered during resizing the application
* Did each team member work as scheduled?
  + Yes

What went right?

* Communication through skype often to make sure everyone was updated on project changes and current tasks.

How to address the issues in the next sprint?

* How to improve the process?
  + Improve the build system to make it easier to run project
* How to improve the product?
  + Fix frame buffers in order to fix memory issue

### Sprint 3

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start Time: 10:30am

End Time: 11:30am

Date: 6/3/16

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Communication through skype often to make sure everyone was updated on project changes and current tasks.

How to address the issues in the next sprint?

* How to improve the process?
  + Improve the code readability
* How to improve the product?
  + Fix bugs

### Sprint 4

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start Time: 7:30am

End Time: 8:30am

Date: 6/3/16

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Communication through skype often to make sure everyone was updated on project changes and current tasks.

How to address the issues in the next sprint?

* How to improve the process?
  + Improve the code readability
* How to improve the product?
  + Fix bugs

### Sprint 5

Attendees: Jorge Nonell, Eric Aguiar, Chris Naranjo, Alex Karpis

Start Time: 7:30pm

End Time: 8:30pm

Date: 8/1/16

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes
* Did we do a good job estimating the points (time required) for each user story?
  + Yes
* Did each team member work as scheduled?
  + Yes

What went right?

* Communication through skype often to make sure everyone was updated on project changes and current tasks.

How to address the issues in the next sprint?

* How to improve the process?
  + Improve the code readability
* How to improve the product?
  + Fix bugs

# References

Here are some of the tutorials that helped us getting started with our technologies

LibCinder: *https://libcinder.org/docs/*

EyeX SDK: *http://developer.tobii.com/eyex-sdk/*

RealSense SDK: *https://software.intel.com/en-us/intel-realsense-sdk/documentation*

LeapMotion SDK: *https://developer.leapmotion.com/documentation/cpp/index.html*

Helpful references and tutorials for hololens development in unity:

Unity3D manual: *https://docs.unity3d.com/Manual/index.html*

HoloAcademy: *https://developer.microsoft.com/en-us/windows/holographic/academy*

Barebones MS HoloLens Documentation: *https://developer.microsoft.com/en-us/windows/holographic/documentation*