*Florida International University*

*School of Computing and Information Sciences*

CIS 4911 - Senior Capstone Project

Software Engineering Focus

Final Deliverable

Multi Modal Interactive Paint

Team # X

**Team Members**

Andrew Mitchell

Garrett Lemieux

**Product Owner**: Ph.D. Francisco R. Ortega

**Instructor**: Masoud Sadjadi

Copyright and trademark notices, restrictions on copying or distributing the documentation, information for contacting the issuing organization (reader’s comments), warranties, contractual obligations or disclaimers, and general warnings and cautions.

***Abstract***

*In this document, we explore the design of our Multi Modal Interactive Paint Application using devices such as a Multi Touch Monitor, Tobii Eyex, Intel Real Sense, and Leap Motion Controller. The sections to follow cover the system design of Application, along with a description of the architectural patterns and design patterns that were chosen. Diagrams complement the described models and designs to aid in the understanding of the choices made. The testing methodologies were recorded and the important tests have been elaborated for the benefit of future developers.*

*Furthermore, a breakdown of the user stories and tasks worked on for Multi Modal Interactive Paint are described in detail. This information is accompanied with details of the hardware and software used. Great emphasis was made on the work done during the sprints, as well as the way work was divided for each sprint. The team’s experiences are detailed through the Sprint Review, Retrospective, and Planning meetings with the product owner.*

**Table of Contents**

[Introduction](#h.30j0zll)

[Current System](#h.1fob9te)

[Purpose of New System](#h.3znysh7)

[User Stories](#h.2et92p0)

[Implemented User Stories](#h.tyjcwt)

[Pending User Stories](#h.3dy6vkm)

[Project Plan](#h.k403nnbriya8)

[Hardware and Software Resources](#h.m66z6y6lu35u)

[Sprints Plan](#h.2s8eyo1)

[Sprint 1](#h.17dp8vu)

[Sprint 2](#h.auxpf86sku2x)

[Sprint 3](#h.dkguh5raizlc)

[Sprint 4](#h.8vzvajelonqb)

[Sprint 5](#h.grsa5qk80uu5)

[Sprint 6](#h.cx6ql7brd16)

[Sprint 7](#h.p5wxd0tqlzu4)

[System Design](#h.sgsdlbnsymfj)

[Architectural Patterns](#h.26in1rg)

[System and Subsystem Decomposition](#h.lnxbz9)

[Deployment Diagram](#h.35nkun2)

[Design Patterns](#h.1ksv4uv)

[System Validation](#h.44sinio)

[Glossary](#h.2jxsxqh)

[Appendix](#h.z337ya)

[Appendix A - UML Diagrams](#h.3j2qqm3)

[Static UML Diagrams](#h.1y810tw)

[Dynamic UML Diagrams](#h.27iiyjxkdfdx)

[Appendix B - User Interface Design](#h.enjhy0pwcd4v)

Appendix C - Sprint Review Reports

[Appendix D - Sprint Retrospective Reports](#h.3whwml4)

Appendix E - Sprint Planning Meetings

[References](#h.2bn6wsx)

https://libcinder.org/docs/

http://developer.tobii.com/eyex-sdk/

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

https://developer.leapmotion.com/documentation/cpp/index.html

# Introduction

This project is a software solution to solve the problem of showcasing FIU's OpenHID lab Smart Desk. The desk can contain various devices that have different input methods. Currently implemented devices consist of a Multi Touch Monitor, Tobii EyeX, Intel Real Sense Device, as well as a Leap Motion Controller. Our goal was to create a Painting Application which showcases these various devices and certain functionality they can provide such as Hand Tracking, Facial Recognition, Eye tracking, and finger tracking on a touch pad. We also wanted an environment where we have some sort of control over what the devices do. This experience is designed to be fun and interactive, and provide a way for students using the Smart Desk to get accustomed to and learn about the various devices available.

## Current System

There is no current application that provides a software solution that the OpenHID lab is looking for. No application combines the use of various devices in a fun and interactive way. To showcase the devices a student would have to use various program from many different software developers such as Intel, Tobii, and Leap to showcase the devices separately. Even then the applications are more of hardware displays rather than a fun and interactive way to showcase the devices.

## Purpose of New System

The purpose of the new application will be to provide a fun and interactive interface where the end user can learn to use and experience various devices. The environment will be a familiar painting application, with the ability to draw using various devices as well as a way to change settings using gestures for the different devices.

The Application will provide various paint functionality such as drawing in different shapes. Some shapes will include lines, circles, rectangles, and triangles. We will be able to change the colors of our drawing, as well as modifying their transparency. In addition we will have some intricate ways to paint such as using a symmetry line and layer functionality.

We will be able to draw with our hands using the leap motion as well as the Real Sense Device. The devices will also have options to cycle through shapes, colors, and change various settings throughout the application.

We also have to develop a User Interface which is easy to use as well as simple to tell what kind of shapes and colors we are drawing. The interface should also be able to control the devices and what sort of tasks they will be able to perform.

# User Stories

This section will cover all of the user stories that we worked on. The main goal of these user stories was to implement the different devices. In addition the user stories involve developing the painting aspects of the program. These two implementations were sometimes developed in tandem, but more often than not tested on a simple device such as the Multitouch device first.

## Implemented User Stories

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

**Andrew Mitchell**

User Story # 497 Read C++ Primer

User Story # 498 Change Line Color

User Story # 508 Develop Basic Software Using Magnum

User Story # 509 Install, Use, And understand TAMGef

User Story # 510 Read C++ Standard Library

User Story # 511 Learn Tobii EyeX Device Capabilities

User Story # 512 Learn to use MultiTouch Monitor

User Story # 514 Watch Videos and Read about the Kinect SDK

User Story # 537 Implement Random Color Mode

User Story # 549 Change Background Color

User Story # 550 Add Multiple Layers

User Story # 552 Create Circle Shape

User Story # 553 Smooth Lines

User Story # 558 Design Proper Feedback for MultiTouch

User Story # 560 Read EyeX Gaze Locations

User Story # 562 Implement Triangle Shape

User Story # 563 Implement Rectangle Shape

User Story # 590 Improve Performance

User Story # 591 Design UI for Multitouch

User Story # 592 Implement basic Feedback for Multitouch

User Story # 623 Create Multitouch Read Me

User Story # 624 Detect Double Tap Gesture

User Story # 626 Enable Alpha Coloring for Shapes

User Story # 627 Create Layer Visualization Menu

User Story # 628 Design Device Modes

User Story # 630 Create Vertical Symmetry Line

User Story # 631 Refactor 'Touches' code

User Story # 662 Add Layer Transparency

User Story # 664 Implement Save Image Transparency

User Story # 665 Make Transparent Eraser

User Story # 666 Implement Leap Shape

User Story # 667 Research hasMultiTouch

User Story # 668 Implement Additional Multitouch Gestures

User Story # 678 Implement 'Override Mode'

User Story # 698 Implement Basic Text

User Story # 699 Add Chrono to Device Detection

User Story # 700 Add Realsense to Override Mode

User Story # 701 Find Icons

User Story # 702 Implement Solid Backgrounds

User Story # 730 Implement Icons

User Story # 731 Implement Draw UI Improvements

User Story # 733 Add Frames Per Second Counter

User Story # 734 Develop Multitouch and EyeX User Guide

User Story # 736 Ensure Bugless Multitouch and EyeX Interaction

**Garrett Lemieux**

User Story # 500 Read C++ Primer

User Story # 501 Read C++ Standard Library

User Story # 502 Develop Basic Software Using Magnum

User Story # 503 Learn to use Kinect Windows Device

User Story # 504 Learn to use LEAP motion Controller

User Story # 507 Install, use, and Understand TAMGeF

User Story # 538 Change Line Size

User Story # 546 Implement Functioning Leap Motion Gestures

User Story # 548 Draw Lines with Leapmotion

User Story # 554 Save a File

User Story # 559 Reset Default Mode on Device Disconnect

User Story # 564 Implement Eraser Mode

User Story # 595 Implement Additional Functionality for Leap Drawing

User Story # 596 Implement Feedback for Leap Motion

User Story # 597 Design User Interface for Leap Motion

User Story # 598 Design Feedback Mechanism for Leap Motion

User Story # 625 Provide Leap Motion Readme for User

User Story # 629 Disable Gestures while Drawing

User Story # 632 Implement Radial Menu for Interactive UI

User Story # 633 Implement Icons for Gestures

User Story # 634 Enable device's Connection Status

User Story # 635 Work with Intel Camera

User Story # 636 Develop Interactive Paint Device Modes

User Story # 649 Design Start Up Image

User Story # 669 Implement a Proximity Menu

User Story # 670 Implement 'Undo' Button

User Story # 671 Implement Functionality for Real Sense Device

User Story # 675 Implement default Modes

User Story # 703 Implement Additional Gestures for Real Sense

User Story # 706 Integrate Real Sense Device into DeviceHandler

User Story # 707 Integrate Real Sense into Default Mode

User Story # 708 Implement Drawing with Hands using Real Sense Device

User Story # 732 Compress Brush Buttons

User Story # 735 Add Start up Image to Application

User Story # 737 Develop Real Sense and Leap Motion User Guide

User Story # 738 Ensure all bugs Removed From Real Sense and Leap Motion functions

## 

## 

## Pending User Stories

User Story # 672 Integrate Orion Into Interactive Paint Program

Delayed until Orion is Updated to work with C++

The Following User stories were intended for future developers and were placed in the product backlog.

User Story # 754 Add Active Pen Support

User Story # 648 Add Additional Brushes

User Story # 704 Add Multitouch Buffering

User Story # 663 Implement a Shader Effect

User Story # 753 Implement Basic Multithreading

User Story # 594 Implement Physical Eye Location Tracking

User Story # 755 Implement Speech Recognition

User Story # 756 Implement Tutorials

User Story # 705 Improve Line Draw Performance

User Story # 555 Load a File

User Story # 556 Playback Drawings

# Project Plan

Hardware is the entire basis of our Software Solution. We were called upon to make an interactive application that can make use of various hardware devices and their unique inputs. We plan on developing the Paint application in tandem with the devices, testing things on easier devices such as the multitouch first and integrating it with more complicated devices such as the Leap Motion, Tobii EyeX, and real sense in the future. The first planned implemented devices are were the Leap Motion and MultiTouch. The EyeX came next, but was soon put on the backburner due to its highly interfering Infra red field, disrupting the Leap and Real Sense devices. The last device implemented was the Intel Real Sense Depth Camera.

## Hardware and Software Resources

**Hardware Resources**

* **Windows 10 PC**
  + Both computers used were laptops, with at least an intel core i5 processor and a fairly recent graphics card, such as the Geforce 940M
* **Acer Touch Screen Display**

The Touch Screen was one input for our Multi Modal Interactive Paint Application. As well as the screen used to view the app.

* **LeapMotion Input Device**
  + One of the more developed devices we used to read hand gestures.
* **Tobii EyeX**
  + Infrared camera used for gaze tracking in the application.
* **Intel Real Sense Depth Camera**
  + Depth Camera used for facial recognition and drawing with your hand

**Software Resources**

* **Windows 10**
  + Necessary for installing Visual Studio 2013
* **Visual Studio 2013**
  + Necessary IDE.
* **C++ Programming Language**
  + The language of choice by Dr. Francisco Ortega. It is also used for many of the SDK's such as the real sense, Eyex and Leap Motion Controller.
* **OpenGL Framework**
  + Used to power the API render features such as 2D shapes and 3D shapes
* **LibUSB 1.0**
  + Used to detect if devices are plugged into the PC.
* **LeapMotion SDK**
  + Used to interact with the LeapMotion device in order to program the LeapMotion gesture detection and finger drawing.
* **Intel Real Sense SDK**
  + Used to interact with the Intel Real Sense device in order to program the facial recognition and hand drawing.
* **Tobii EyeX SDK**
  + Used to interact with the EyeX device and program Gaze Tracking.
* **LibCinder**
  + an OpenGL wrapper that helps us draw primitive shapes and have control over our canvas where we can write FBO's.

## 

## 

## 

## Sprints Plan

**Sprints Plan**

***Sprint 1***

**(1/18/2016 – 2/12/2016)**

**Andrew Mitchell**

**User Story # 497 Read C++ Primer**

### Description:

* Learn C++ to help with programming smart desk applications, including our Virtual Environment Paint program. For this story I will be using the C++ Primer 5th Edition.
* 9~ Hours…
* As a Team Developer I would like to read about C++ so I can effectively code our VE Paint Program

### Acceptance Criteria:

1. Review C++ using Primer 5th Edition

**User Story # 498 Change Line Color**

### Description:

* As a User I would like to change the color of the line I am drawing to make my drawings more diverse.

### Acceptance Criteria:

1. Draw a line of any color using multitouch.
2. Be able to change the color of the next line we begin drawing.
3. Be able to cycle through multiple colors.

**User Story # 508 Develop Basic Software Using Magnum**

### Description:

* Learning about Magnum, including downloading it’s functions, playing with the functions, and reading up on all it’s capabilities.
* As a Team Developer I would like to learn Magnum 2d/3d Developer so I can implement GUI Implementation to our VE Paint Program
* Abandoned Magnum for LibCinder

### Acceptance Criteria:

1. Review Magnum 2d/3d engine found here <https://github.com/mosra/magnum> .
2. Develop a small program using Magnum

**User Story # 509 Install, Use, And understand TAMGef**

### Description:

* I will be looking at the TAMGeF Solution the previous semester made by using their framework. I will install and use their application to test the technologies provided to us by Professor Ortega.
* As a Team Developer I would like to learn how the Visualizer works from the previous release so I can better understand how to handle input from our technologies.

### Acceptance Criteria:

1. Download and install previous semesters program.
2. Use ASUS MultiTouch Screen to test it.
3. Understand and create any questions to ask for our Mentors.

**User Story # 510 Read C++ Standard Library**

### Description:

* Read the C++ Standard Library book (More advanced than C++ Primer) to get a better understanding of the tools that C++ provides.
* As a Team Developer I would like to learn more about the C++ Standard Library so I can be a better programmer for our Paint Program.

### Acceptance Criteria:

1. Read through C++ Standard Library to understand core concepts
2. Skim through this before our C++ Workshop Tenatively next wednesday
3. Understand C++!

**User Story # 511 Learn TobiiEyeX Device Capabilities**

### Description:

* Download SDK’s for TobiieyeX and play with online examples as well as watching youtube videos to figure out its capabilities.
* Error downloading drivers. Windows cannot find them!!
* As a Team Developer I would like to learn about the tobiieyeX so I can understand its limitations and think of features for our paint program.

### Acceptance Criteria:

1. Download SDK and use tobiieyeX
2. Understand capabilities so I can design features for the paint program

**User Story # 512 Learn to use MultiTouch Monitor**

### Description:

* Download SDK’s and get the ACER Multitouch Screen to work. Play with it and follow examples online to see it’s limitations. Brainstorm features for paint program.
* As a Team Developer I would like to learn about the ACER Multi Touch Screen so I can learn about its capabilities and limitations so I can design features for the Paint Program.

### Acceptance Criteria:

1. Download SDK and have the multitouch working.
2. Better understand capabilities
3. Brainstorm ideas for Super Paint Program.

**User Story # 514 Watch Videos and Read about the Kinect SDK**

### Description:

* Learn about the Microsoft Kinect via youtube and other services so I can better understand its capabilities and limitations. I will also brainstorm ideas on how this device can be helpful for our paint program.
* USER STORY – As a Team Developer I would like to better understand the Microsoft Kinect so I can judge its capabilities and brainstorm ideas for the paint program.

### Acceptance Criteria:

1. Understand capabilities and limitations of the Microsoft Kinect
2. Brainstorm ideas for Paint Program

**User Story # 537 Implement Random Color Mode**

### Description:

* As a User I would like to be able to Randomize the colors of the line I draw so I can use the multitouch and draw with many vibrant colors at once.

### Acceptance Criteria:

1. Randomize the color of the next line drawn.
2. The mode must be able to toggle on and off.

**Garrett Lemieux**

**User Story # 500 Read C++ Primer**

### Description:

* As a Team Developer I would like to read about C++ so I can effectively code our Paint Program
* Learn C++ syntax and concepts to help with programming smart desk applications, including our Virtual Environment Paint program. We will use books provided in the dropbox provided by our product owner.

### Acceptance Criteria:

    1. Understand how to efficiently and effectively use C++ for Paint Program

**User Story # 501 Read C++ Standard Library**

### Description:

* As a Team Developer I would like to read about C++ standard library so I can efficiently code our Paint Program.
* Learn C++ standard library in order to promote re-usability for the developer and allow for more efficient development of the Paint Program.

### Acceptance Criteria:

    1. Understand how to efficiently and effectively use the C++ standard library within the Paint Program.

**User Story # 502 Develop Basic Software Using Magnum**

### Description:

* As a Team Developer I would like to learn Magnum ,a 2D/3D graphics engine, so I can implement features in our Paint Program
* Learn Magnum graphics engine to allow programmer to implement product owners desired features for Paint Program.

### Acceptance Criteria:

1. Understand how Magnum 2D/3D developer works.
2. Functions downloaded and worked with.
3. Develop a basic program using Magnum.

**User Story # 503 Learn to use Kinect Windows Device**

### Description:

* As a Team Developer I would like to better understand how the Kinect Windows device works so I can brainstorm possible user interactions with device for the Paint Program
* Set up Kinect Device on laptop and explore the device’s capabilities which will provide knowledge of the technology that will be used in the Paint Program, along with providing experience with the device to allow for better understanding of possible user interactions for the 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.

**User Story # 504 Learn to use LEAP motion Controller**

### Description:

* As a Team Developer I would like to better understand how the LEAP Motion Controller works so I can understand the user interactions with device for the Paint Program
* Set up LEAP MOTION controller on laptop and explore the device’s capabilities which will provide knowledge of the technology that will be used in the Paint Program.
* Explore user interactions provided by LEAP Motion Controller which are related specifically to hand and finger gestures for the Paint Program.

### Acceptance Criteria:

1. SDK for Leap Motion Controller downloaded and device working on laptop.
2. Understand how the Leap Motion Controller functions and be able to work with device.
3. Come up with ideas for Paint Program.

**User Story # 507 Install, use, and Understand TAMGeF**

### Description:

* As a Team Developer I would like to explore the Visualizer created in previous release so I can get a better understanding of how multi touch functionality can be implemented for the Paint Program
* I will be examining TAMGeF and the framework created from previous semester.
* I will also use the application in order to test the LEAP Motion Controller and Kinect Device

### Acceptance Criteria:

1. Previous semester program installed on laptop.
2. Program tested with both Kinect and Leap devices.
3. Record any possible questions.

**User Story # 538 Change Line Size**

### Description:

* As a User I would like to change the width of the a line to be drawn in order to allow for varying line sizes to be drawn.

### Acceptance Criteria:

1. User can change the size of the line before line is drawn.

**User Story # 564 Implement Functioning Leap Motion Gestures**

### Description:

* As a User I would like to erase previously drawn lines so I can fix mistakes made while using the multitouch device.

### Acceptance Criteria:

1. Be able to turn eraser mode on and off at users desire.
2. When previously drawn lines are touched the portion touched by user is erased.

***Sprint 2***

**(2/1/2016 – 2/12/2016)**

**Andrew Mitchell**

**User Story # 549 Change Background Color**

### Description:

* As a User I would like to change the background color so I can work on different canvas colors

### Acceptance Criteria:

1. Be able to change the background color
2. Cycle through background colors.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [567](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/567) | [Make Background Color Cycle](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/567) | Done |

**User Story # 550 Add Multiple Layers**

### Description:

* As a User I would like to have multiple layers to draw on so that I can edit a single layer at a time.

### Acceptance Criteria:

1. Enable multiple layers
2. EraserMode only erases active layers

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [588](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/588) | [Implement Frame Buffer Objects.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/588) | Done |
| [583](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/583) | [Design how layering will work.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/583) | Done |
| [526](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/526) | [Attend C++ Workshop](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/526) | Done |

**User Story # 552 Create Circle Shape**

### Description:

* As a User I would like to draw a predefined circle so my circles are smooth.

### Acceptance Criteria:

1. Enable drawing of a circle
2. Circle can be various sizes.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [571](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/571) | [Draw filled circle](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/571) | Done |
| [570](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/570) | [Draw dynamically changing circles.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/570) | Done |
| [569](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/569) | [Draw a basic circle.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/569) | Done |

**User Story # 553 Smooth Lines**

### Description:

* As a User I would like my large lines to be smooth and not rough so my drawings look presentable.

### Acceptance Criteria:

1. Larger lines must be smooth (No more sharp indents on larger lines).

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [589](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/589) | [Implement Algorithm.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/589) | Done |
| [584](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/584) | [Learn about Bresenhams line Algorithm](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/584) | Done |
| [577](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/577) | [Attend LibCinder Workshop](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/577) | Done |
| [572](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/572) | [Learn more about openGL and cinder](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/572) | Done |

**User Story # 560 Read EyeX Gaze Locations**

### Description:

* As a User I would the eyeX device to be able to read where I am looking so I can access certain functionality based on where I am looking.

### Acceptance Criteria:

1. Be able to read which section of the screen the user is looking
2. 4 Distinct sections.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [582](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/582) | [Develop simple mode switch using eyeX](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/582) | Done |
| [581](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/581) | [Read EyeX location inside App.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/581) | Done |
| [580](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/580) | [Follow Samples for EyeX](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/580) | Done |
| [578](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/578) | [Get EyeX Drivers working on Laptop.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/578) | Done |

**User Story # 562 Implement Triangle Shape**

### Description:

* As a User I would like to draw a predefined triangle shape so my triangles are smooth and normal looking.

### Acceptance Criteria:

1. Implement Triangle Shape
2. Must be variable size (takes input from multitouch)

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [576](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/576) | [Draw a triangle dynamically](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/576) | Done |

**User Story # 563 Implement Rectangle Shape**

### Description:

* As a User I would like to implement a rectangle shape so I can draw a smooth rectangle.

### Acceptance Criteria:

1. Implement Rectangle Shape
2. Reads input (Top left corner and bottom right corner)

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [575](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/575) | [Draw rectangle dynamically](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/575) | Done |
| [574](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/574) | [Draw basic rectangle](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/574) | Done |

**Garrett Lemieux**

**User Story # 546 Implement Eraser Mode**

### Description:

* As a User I want preset Leap Motion gestures so I can perform additional functionality within the interactive paint program.

### Acceptance Criteria:

1. Create at least two gestures to be used by TouchPointsApp.
2. User must be provided basic feedback that gesture was detected.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [573](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/573) | [Implement leap motion gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/573) | Done |
| [568](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/568) | [Review Leap Motion SDK](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/568) | Done |

**User Story # 548 Draw Lines with Leap Motion**

### Description:

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

### Acceptance Criteria:

1. User must be able to draw lines using the leap motion.
2. User interaction while drawing must be smooth and simple.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [585](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/585) | [Attend LibCinder Workshop](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/585) | Done |
| [579](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/579) | [Create simple line drawing UI](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/579) | Done |

**User Story # 554 Save a File**

### Description:

* As a User I would like to save to a file so I can reload it at a later time.

### Acceptance Criteria:

1. Save a file.
2. Save multiple types of files.
3. All data must be in the file.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [587](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/587) | [Allow file to be saved in different formats](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/587) | Done |
| [586](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/586) | [Implement Save Function](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/586) | Done |
| [499](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/499) | [Attend C++ Workshop at FIU](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/499) | Done |

***Sprint 3***

**(2/15/2016 – 2/26/2016)**

**Andrew Mitchell**

**User Story # 558 Design Proper Feedback for MultiTouch**

### Description:

* As a User I would like to have logical feedback when using a multitouch device so I know when I am using the multi touch properly.

### Acceptance Criteria:

1. Design a good way to implement feedback for the multitouch.
2. Design on paper how we want to implement feedback.
3. Meet with Francisco Ortega for in depth discussion on feedback.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [606](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/606) | [Design Basic Feedback](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/606) | Done |
| [605](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/605) | [Use Current 'Touch' Sketch App's](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/605) | Done |
| [593](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/593) | [Attend Design Meeting](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/593) | Done |

**User Story # 590 Improve Performance**

### Description:

* As a User I want the line drawing to be smooth (performance wise) so it is not choppy while I draw.

### Acceptance Criteria:

1. Improve performance of line drawing
2. Improve / modify Breshams algorithm.
3. Learn more about how often libcinder draws and calls certain functionality.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [604](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/604) | [Integrate code](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/604) | Done |
| [601](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/601) | [Implement performance improvement changes](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/601) | Done |
| [600](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/600) | [Study Algorithm and Code](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/600) | Done |

**User Story # 591 Design UI for Multitouch**

### Description:

* As a User I would like a proper UI Design when using the multitouch so it is intuitive and easy to use.

### Acceptance Criteria:

1. Design a UI for our program that primarily uses the multitouch
2. Design button placement etc.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [618](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/618) | [Implement Status Icons.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/618) | Done |
| [617](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/617) | [Meet with Jonathan](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/617) | Done |

**User Story # 592 Implement basic Feedback for Multitouch**

### Description:

* As a User I would like feedback when I change from various modes using the multitouch and its UI so I can tell what functionality I can expect when I draw on the multitouch.

### Acceptance Criteria:

1. Implement feedback for multitouch
2. Follow design previously done in this sprint.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [616](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/616) | [Create Shapes button](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/616) | Done |
| [613](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/613) | [Create inInteractiveUi function.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/613) | Done |
| [612](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/612) | [Develop Simple Button](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/612) | Done |
| [607](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/607) | [Implement Draw Mode Square.](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/607) | Done |

**Garrett Lemieux**

**User Story # 595 Implement Additional Functionality for Leap Drawing**

### Description:

* As a User I would like to be able to to have the same functionality as the multi touch device when drawing lines with leap motion in order to allow for diverse images to be created with leap motion.

### Acceptance Criteria:

1. Be able to draw different size lines.
2. Be able to change color of lines.
3. Leap motion and multi touch must have same functionality.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [602](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/602) | [Implement random color and change color](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/602) | Done |
| [599](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/599) | [Implement different Size lines](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/599) | Done |

**User Story # 596 Implement Feedback for Leap Motion**

### Description:

* As a User I would like feedback when I switch draw settings using the Leap Motion device and interact with UI so I know intended actions with Leap Motion were successful.

### Acceptance Criteria:

1. Provide feedback in the form of an image.
2. Gesture must change current Ui or draw settings.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [622](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/622) | [Implement change shape with Leap Gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/622) | Done |
| [621](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/621) | [Implement change color with Leap Gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/621) | Done |
| [615](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/615) | [Fix feedback bug on save function](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/615) | Done |
| [614](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/614) | [Use Leap Gesture for save feedback](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/614) | Done |
| [611](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/611) | [Display image for specified time](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/611) | Done |
| [610](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/610) | [Load an image for feedback](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/610) | Done |

**User Story # 597 Design User Interface for Leap Motion**

### Description:

* As a User I would like a well designed User Interface when interacting with Leap Motion device so it is simple to use the program.

### Acceptance Criteria:

1. A simple user interface design.
2. Canvases and button placement must be mapped out.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [620](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/620) | [Radial Menu with UI](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/620) | Done |
| [619](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/619) | [Meeting with Jonathan](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/619) | Done |

**User Story # 598 Design Feedback Mechanism for Leap Motion**

### Description:

* As a User I would like feedback when using a Leap Motion device so I know when I am using the device properly.

### Acceptance Criteria:

1. Meet with Product Owner to discussion desired feedback.
2. Describe desired feedback implementation using visual representations.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [609](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/609) | [Design Feeback for UI](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/609) | Done |
| [608](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/608) | [Use sketch touch application](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/608) | Done |
| [603](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/603) | [Attend Design Meeting](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/603) | Done |

***Sprint 4***

**(2/29/2016 – 3/11/2016)**

**Andrew Mitchell**

**User Story # 623 Create Multitouch Read Me**

### Description:

* As a User I would like a readme for the multitouch device so I understand what the program can do using my multitouch surface.

### Acceptance Criteria:

1. Easy to read User ReadMe
2. ReadMe must be available on Github.
3. Goes over various Multitouch functionalities (Including buttons)
4. Go over how shapes are created.
5. What does each button or piece of the UI Represent?

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [639](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/639) | [Meet with Jonathan](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/639) | Done |
| [638](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/638) | [Create readMe for Multitouch](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/638) | Done |

**User Story # 624 Detect Double Tap Gesture**

### Description:

* As a User I would like to double tap the screen to perform various additional functionality such as pulling up a menu.

### Acceptance Criteria:

1. Double tap must successfully read
2. Cannot draw on the first tap
3. Must show that it is consistent and easy to use.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [651](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/651) | [Implement Double Tap Gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/651) | Done |
| [650](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/650) | [Design Multitouch Gestures](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/650) | Done |

**User Story # 626 Enable Alpha Coloring for Shapes**

### Description:

* As a User I would like the Option to use Alpha coloring to change the transparency of the colors I use.

### Acceptance Criteria:

1. Restructure the code to enable our ‘shapes’ to use alpha coloring instead of only accepting RGB colors.
2. Be able to change the Alpha Color value
3. Put on alpha color buttons.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [641](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/641) | [Implement Alpha Coloring in TouchPointsApp](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/641) | Done |
| [640](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/640) | [Redo Code for TouchShapes](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/640) | Done |

**User Story # 627 Create Layer Visualization Menu**

### Description:

* As a User I want an interface to view which layers are currently active and on top of eachother.

### Acceptance Criteria:

1. Show which layers are currently on top of one another.
2. Show which layers are being drawn

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [658](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/658) | [Implement Layer Visualization](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/658) | Done |
| [652](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/652) | [Design Layer Visualization UI](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/652) | Done |

**User Story # 628 Design Device Modes**

### Description:

* As a User I would like different modes so I can use the device for different purposes depending on which mode I am currently in.
* Example : Dual Draw Mode – Enables drawing with both touch and leap motion
* vs ‘Ideal Mode’ – Leap is used only for gestures (doesn’t draw), touch is used for draw.
* ‘No touch mode’ – Leap is used for drawing and gestures, etc.

### Acceptance Criteria:

1. Must discuss with Garrett which various modes we want to set up.
2. Have a list of the various modes we want to begin working on next sprint.
3. Must review with the product owner before end of sprint.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [659](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/659) | [Design Modes](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/659) | Done |

**User Story # 630 Create Vertical Symmetry Line**

### Description:

* As a User I would like to Enable a Symmetry so I can easily draw on both sides of the canvas at once, to make nice symmetric drawings.

### Acceptance Criteria:

1. Be able to draw on the left side of the canvas and it appears at the same height on the right side of the canvas and vice versa.
2. Mode must be toggle-able.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [646](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/646) | [Implement symmetryLines class](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/646) | Done |
| [645](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/645) | [Design symmetryLine class](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/645) | Done |
| [644](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/644) | [Libcinder workshop with Alain](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/644) | Done |

**User Story # 631 Refactor 'Touches' code**

### Description:

* As a Future Developer I would like the code for reading touches to be easy to read and follow so I can jump right into the project and understand how it reads touches more easily.

### Acceptance Criteria:

1. Separate the drawing functionality into their own functions.
2. Ensure we have functions that take an id as well as a vec2 point to draw.

**Garrett Lemieux**

**User Story # 625 Provide Leap Motion Readme for User**

### Description:

* As a User I would like to be able to be provided a detailed document explaining how to use the leap motion device with the Interactive Paint Program in order to efficiently and effectively interact with the Interactive Paint Program.

### Acceptance Criteria:

1. Provide a description of how the leap motion sensor works.
2. Provide description of all gestures and there functions.
3. Provide information on general use of program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [637](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/637) | [Provide user with instructions for leap motion](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/637) | Done |

**User Story # 629 Disable Gestures while Drawing**

### Description:

* As a User I would like to be able to draw using leap motion without gestures being observed in order to prevent accidental mode changes.

### Acceptance Criteria:

1. User must be able to draw and not change modes accidentally with a Leap Motion gesture.

### **Related Task:**

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [647](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/647) | [Implement the disabling of gestures while leap drawing](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/647) | Done |

**User Story # 632 Implement Radial Menu for Interactive UI**

### Description:

* As  user I would like to change modes in Interactive UI in order to allow for intuitive and effective interaction with Interactive Paint Program.

### Acceptance Criteria:

1. Implement Radial menu that will allow for different modes to appear and be selected.
2. Provide feedback to user on mode being selected.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [657](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/657) | [Implement dynamic use of Radial Menu](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/657) | Done |
| [656](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/656) | [Add funcitonality to each button](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/656) | Done |
| [655](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/655) | [Implement Basic Radial Menu](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/655) | Done |

**User Story # 633 Implement Icons for Gestures**

### Description:

* As a User I would like to have simple well designed icons that provide insight on the action or representation the icon represents in order to allow for effective User interaction.

### Acceptance Criteria:

1. Implement icons that represent shapes.
2. Implement icons that represent colors.
3. Implement icons that show different modes.

**User Story # 634 Enable device's Connection Status**

### Description:

* As a User I would like to be informed which devices are enabled in order to know what modes can be chosen and actions that can be made.

### Acceptance Criteria:

1. User Interaction UI must dynamically update when a device has been connected or disconnected.
2. Modes must be updated automatically and actions be enabled or disabled dynamically.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [661](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/661) | [Create function updates devices current status](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/661) | Done |
| [643](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/643) | [Find each devices vendor id](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/643) | Done |
| [642](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/642) | [Meet with Jason for device connection status](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/642) | Done |

**User Story # 635 Work with Intel Camera**

### Description:

* As a team developer I would like to learn how Intel camera SDK works in order to implement its functionality into the Interactive Paint Program.

### Acceptance Criteria:

1. Intel camera SDK downloaded and reviewed.
2. Determine functionality that can be used in Paint Program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [654](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/654) | [Work with Real Sense device](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/654) | Done |
| [653](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/653) | [Install RealSense SDK](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/653) | Done |

**User Story # 636 Develop Interactive Paint Device Modes**

### Description:

* As a User I would like to be provided different functionality modes so I can use the device in different ways to allow users desired painting environment.

### Acceptance Criteria:

1. Discuss with Andrew and create modes that will work best for all devices functionality and users interaction with smart desk.
2. Provide all modes and the devices functions within each mode.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [660](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/660) | [Design Different Modes](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/660) | Done |

***Sprint 5***

**(3/21/2016 – 4/1/2016)**

**Andrew Mitchell**

**User Story # 662 Add Layer Transparency**

### Description:

* As a User I would like to make each individual layer as transparent as I want to make it easier to see which layer I am working on.

### Acceptance Criteria:

1. Create buttons for layer transparency
2. Each layer is independent of one another.
3. Implement Layer Transparency!

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [683](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/683) | [Implement Button for Alpha Layering](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/683) | Done |
| [682](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/682) | [Implement Layer Transparency](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/682) | Done |
| [681](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/681) | [Design Layer Alpha Coloring](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/681) | Done |

**User Story # 664 Implement Save Image Transparency**

### Description:

* As a User I would like to save an Image with a Transparent Background so I can have more diversity on which surfaces I can put my images on.

### Acceptance Criteria:

1. When the user saves, the background must save transparent (unless colored over).

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [680](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/680) | [Implement Image Transparency](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/680) | Done |
| [679](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/679) | [Research LibCinder image Saving](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/679) | Done |

**User Story # 665 Make Transparent Eraser**

### Description:

* As a User I would like the eraser to erase to clear so I don’t see the erase lines when using multiple layers.

### Acceptance Criteria:

1. Make sure the eraser draws as ‘0’ alpha as opposed to background color.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [685](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/685) | [Implement Transparent Eraser](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/685) | Done |
| [684](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/684) | [OpenGL Research](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/684) | Done |

**User Story # 666 Implement Leap Shape**

### Description:

* As a User I would like to draw shapes with the leap motion so I can have another ‘complete’ drawing medium.

### Acceptance Criteria:

1. Must be able to use all draw functionality with leap motion. Including shapes.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [674](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/674) | [Implement Shapes for Leap Motion](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/674) | Done |
| [673](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/673) | [Design 'activeDrawings' for Leap](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/673) | Done |

**User Story # 667 Research hasMultiTouch**

Description

### Description:

* As a User I would like to see if my multitouch is plugged into my device reliably so I can see what functionality I have access to.

### Acceptance Criteria:

1. Check how libcinder implements its hasmultitouch call
2. How and where does it cache the data? Can we get rid of the cache?

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [691](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/691) | [Research other MultitouchPossiblities](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/691) | Done |
| [690](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/690) | [Look into Libcinder Code](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/690) | Done |

**User Story # 668 Implement Additional Multitouch Gestures**

### Description:

* As a User I would like more multitouch gestures so I can have more functionality at my finger tips.

### Acceptance Criteria:

1. Add more gestures.
2. Add ‘Extended Touch’
3. Add Two finger tap

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [687](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/687) | [Implement Extended Touch](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/687) | Done |
| [686](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/686) | [Implement Two Finger Touch](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/686) | Done |

**User Story # 678 Implement 'Override Mode'**

### Description:

* As a User I would like to enable or disable certain devices and their modes so I can experience the program using devices in different ways and find which way suits me best.

### Acceptance Criteria:

1. Enable ‘Override Mode’ button that enables you to switch certain functionality on and off for different devices.
2. Must work for multitouch
3. Must work for leap (Gestures and Drawing)
4. Must work for EyeX

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [697](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/697) | [Implement Override Buttons](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/697) | Done |
| [693](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/693) | [Discuss Override Mode](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/693) | Done |

**Garrett Lemieux**

**User Story # 669 Implement a Proximity Menu**

### Description:

* As a User I would like to be able to use the Leap motion device to select different modes in order to allow for more interaction with UI.

### Acceptance Criteria:

1. User should be able to change modes by using proximity menu with leap motion device.
2. User should be provided feedback when proximity menu has updated application.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [677](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/677) | [Implement gesture recognition in menu](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/677) | Done |
| [676](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/676) | [Implement menu display](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/676) | Done |

**User Story # 670 Implement 'Undo' Button**

### Description:

* As a User I would like to be able to restore previous drawn states in order to allow mistakes to be made and further development on canvas.

### Acceptance Criteria:

1. User must be able to undo or restore previous state.
2. User must be able to select this option using UI.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [689](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/689) | [Implement functionality for Undo button](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/689) | Done |
| [688](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/688) | [Create Button for Undo Function](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/688) | Done |

**User Story # 671 Implement Functionality for Real Sense Device**

### Description:

* As a User I would like to be able to use Real Sense Device in Paint Program in order to allow for more interactive experiences with smart desk.

### Acceptance Criteria:

1. User can perform a facial gesture such as a kiss gesture and action occurs in program.
2. User can perform a facial gesture such as a double eyebrow raise and action occurs in program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [695](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/695) | [Implement Kiss Gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/695) | Done |
| [694](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/694) | [Implement Eyebrow Raise Gesture](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/694) | Done |
| [692](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/692) | [Create expression for real sense](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/692) | Done |

**User Story # 675 Implement default Modes**

### Description:

* As a User I would like to have a default modes when certain devices are connected in order to allow for expected functionality with Interactive Paint.

### Acceptance Criteria:

1. Application should have preset modes for different combinations of devices connected to application.
2. Modes should be automatically set when application starts.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [696](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/696) | [Implement Default Modes](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/696) | Done |

***Sprint 6***

**(4/4/2016 – 4/15/2016)**

**Andrew Mitchell**

**User Story # 698 Implement Basic Text**

### Description:

* As a User I would like to Write Text into the Program so I can have nice letters available when I want them.
* Implement Basic Text

### Acceptance Criteria:

1. Create an On Screen Keyboard to type using the Multitouch
2. Must be able to designate where I want to write the text.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [729](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/729) | [Implement Key Functionality and Movement](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/729) | Done |
| [722](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/722) | [Implement Keyboard Graphics](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/722) | Done |

**User Story # 699 Add Chrono to Device Detection**

### Description:

* As a User I want the Program to run smoothly so I can draw without dealing with a laggy interface.

### Acceptance Criteria:

1. Implement and Test Chrono functionality for our device detection service.
2. Ensure that the deviceConnection isn’t called too often and slows down our program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [711](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/711) | [Test Chrono Implementation](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/711) | Done |
| [709](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/709) | [Add Chrono Delay to Device Detection](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/709) | Done |

**User Story # 700 Add Realsense to Override Mode**

### Description:

* As a User I would like to toggle on and off the RealSense Device and its functionality so I can use the device how I want.

### Acceptance Criteria:

1. Enable a realsense button into the override devices menu.
2. Add a button to enable and disable all realsense functionality (depending on what is added, including at least expressions)

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [719](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/719) | [Add Functionality to Real Sense Buttons](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/719) | Done |
| [718](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/718) | [Add Interface for Real Sense](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/718) | Done |

**User Story # 701 Find Icons**

### Description:

* As a User I want the Icons for all the buttons to look intuitive so I understand what each button does.

### Acceptance Criteria:

1. Research different Icons.
2. Try to find an icon package the openHID lab can buy so we can implement next sprint.
3. Get a list of icons we want to use for our specific buttons.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [712](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/712) | [Look for Icon Packages](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/712) | Done |

**User Story # 702 Implement Solid Backgrounds**

### Description:

* As a User I would like to save my images with a solid background so I know how the image file looks when saving it.

### Acceptance Criteria:

1. Implement backgrounds that save to a PNG file.
2. Implement a ‘Checkerboard’ background to indicate when the background is ‘transparent’

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [720](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/720) | [Implement Buttons and Solid Save Feature](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/720) | Done |
| [716](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/716) | [Switch Default Background to White](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/716) | Done |
| [715](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/715) | [Implement Checkerboard FBO](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/715) | Done |

**Garrett Lemieux**

**User Story # 559 Reset Default Mode on Device Disconnect**

### Description:

* As a User I would like to be able to disconnect any device or device breaks and have default mode updated so I can choose which devices to use in the program while its running.

### Acceptance Criteria:

1. User must be able to disconnect any device or device breaks and default mode updated automatically.
2. Correct functionality for each device must be set.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [721](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/721) | [Implement automatic reset default mode](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/721) | Done |

**User Story # 649 Design Start Up Image**

### Description:

* As a user I would like to see an image displayed that represents the program in order to improve my experience using interactive paint.

### Acceptance Criteria:

1. Image must be designed that will represent the program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [725](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/725) | [Meet with alan to discuss development of logo](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/725) | Done |
| [724](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/724) | [Design startup logo for interactive paint](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/724) | Done |

**User Story # 703 Implement Additional Gestures for Real Sense**

### Description:

* As a user I would like to be able to perform additional Gestures in order to allow more interaction with paint program using Real Sense device.

### Acceptance Criteria:

1. New gestures created.
2. Each gesture must perform a different function.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [714](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/714) | [Implement New Facial Gestures](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/714) | Done |
| [713](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/713) | [Interact with real sense to design new facial expressions](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/713) | Done |

**User Story # 706 Integrate Real Sense Device into DeviceHandler**

### Description:

* As a User I would like to be informed when real sense is enabled in order to know what modes can be chosen and actions that can be mad.

### Acceptance Criteria:

1. User Interaction UI must dynamically update when Real Sense has been connected or disconnected.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [710](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/710) | [Integrate Real Sense into Device Handler](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/710) | Done |

**User Story # 707 Integrate Real Sense into Default Mode**

### Description:

* As a User I would like to have  default modes set when using Real Sense device in order to allow for expected functionality with Interactive Paint.

### Acceptance Criteria:

1. Application should have preset modes for different combinations of devices which includes when Real Sense device is connected to application.
2. Modes should be automatically set when application starts.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [717](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/717) | [Update current default modes with real Sense](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/717) | Done |

**User Story # 708 Implement Drawing with Hands using Real Sense Device**

### Description:

* As a user I would like to be able to draw using Real Sense device in order to allow more interaction with paint program when using Real Sense Device.

### Acceptance Criteria:

1. User must be able to interact with paint program in a way other than facial gestures.
2. Hand interaction with device must be implemented.
3. User must be able to draw using hands with the real sense.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [728](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/728) | [Test Drawing zone](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/728) | Done |
| [727](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/727) | [Implement drawing zone](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/727) | Done |
| [726](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/726) | [Implement hover zone for drawing](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/726) | Done |
| [723](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/723) | [Implement Hand module into Interactive Paint](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/723) | Done |

***Sprint 7***

**(4/18/2016 – 4/29/2016)**

**Andrew Mitchell**

**User Story # 730 Implement Icons**

### Description:

* As a User I would like Nice and Obvious Icons to Quickly tell what a buttons functionality is.

### Acceptance Criteria:

1. Implement Icons from Iconfinder into the program.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [745](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/745) | [Implement Icons](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/745) | Done |

**User Story # 731 Implement Draw UI Improvements**

### Description:

* As a User I want the UI to only be drawn once, so the performance of the program is faster.

### Acceptance Criteria:

1. Move all the button and UI draw calls to individual frame buffer objects.
2. Only draw the framebuffers in ‘drawUi’

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [739](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/739) | [Move Draw Calls to UI Setup](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/739) | Done |

**User Story # 733 Add Frames Per Second Counter**

### Description:

* As a User I would like to view the Frames Per Second so I can see the impact that certain devices have on the smoothness of the program.

### Acceptance Criteria:

1. Add a FPS counter to the top right of the screen.
2. Must be able to toggle on and off.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [740](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/740) | [Implement FPS Counter](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/740) | Done |

**User Story # 734 Develop Multitouch and EyeX User Guide**

### Description:

* As a User I would like an in depth guide to see how the program runs as well as how to use the program.

### Acceptance Criteria:

1. Ensure all the documentation is up to date (Including Use Cases, Sequence, and Class Diagrams).
2. Make sure the ‘User Guide’ is implemented and has good images (and updated images) to show where the functionality resides.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [751](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/751) | [Update Feature Documents](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/751) | Done |

**User Story # 736 Ensure Bugless Multitouch and EyeX Interaction**

### Description:

* As a User I would like a smooth and bugless multitouch interaction so the program feels great and doesn’t crash.

### Acceptance Criteria:

1. Redo all the unit tests involving multitouch functionality
2. Redo all the unit tests involving EyeX Functionality.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [752](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/752) | [Redo all our old unit tests](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/752) | Done |

**Garrett Lemieux**

**User Story # 732 Compress Brush Buttons**

### Description:

* As a User I would like to have an organized and intuitive user interface with minimal buttons in order to allow easy interactions with application.

### Acceptance Criteria:

1. Buttons must be compressed to allow full functionality with minimal number of buttons.
2. Buttons must be intuitive to use.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [744](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/744) | [Create compressed buttons](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/744) | Done |

**User Story # 735 Add Start up Image to Application**

### Description:

* As a user I would like to be informed the name of the application in form of an appealing image in order to enhance my experience with the application.

### Acceptance Criteria:

1. When application begins, start image must appear and fade out after a designated time period.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [743](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/743) | [Test image](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/743) | Done |
| [742](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/742) | [Adjust size and fade](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/742) | Done |
| [741](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/741) | [Load Image on startup](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/741) | Done |

**User Story # 737 Develop Real Sense and Leap Motion User Guide**

### Description:

* As a user I would like a detailed user guide in order to work efficiently and correctly with the devices within the application.

### Acceptance Criteria:

1. User should be able to understand how the leap motion device works within the  application.
2. User should be able to understand how the real sense device works within the application.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [757](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/757) | [Update User Guide and Documentation](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/757) | Done |

**User Story # 738 Ensure all bugs Removed From Real Sense and Leap Motion functions**

### Description:

* As a user I would like to use the Real Sense and Leap motion devices without bugs affecting the interaction with the application in order to allow for a fun experience with interactive paint program.

### Acceptance Criteria:

1. Test all functions for Real Sense and redo unit tests.
2. Test all functions for Leap Motion and redo unit tests.

### Related Tasks:

|  |  |  |
| --- | --- | --- |
| **Number** | **Name** | **Status** |
| [750](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/750) | [Retest and Update Old Features](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/750) | Done |
| [749](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/749) | [Retest and Update Old Features](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/749) | Done |
| [748](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/748) | [Retest and Update Old Features](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/748) | Done |
| [747](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/747) | [Retest and Update Old Features](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/747) | Done |
| [746](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/746) | [Retest and update Old Features](https://fiu-scis-seniorproject.mingle.thoughtworks.com/projects/modern_touch/cards/746) | Done |

# System Design

System design defines the strategies taken by the developers to construct the desired project.

Here we break up our design goals into the architectural patterns used, the system and

subsystem decomposition models, and the design patterns. These make up a large part of the

planning that was necessary for putting this project on course. An in-depth

look at the choice’s for this semester’s design are looked at further in the upcoming subsections.

## Architectural Patterns

For our application we decided on the Model-View-Control architecture. We decided to use the Model View Control architecture because we felt it works with our system.

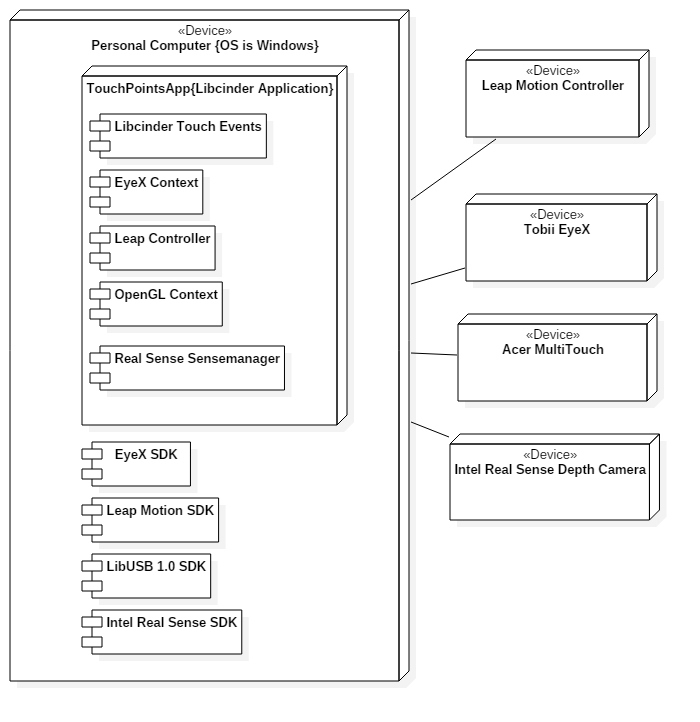
**System and Subsystem Decomposition**

For our Model we use simple Frame Buffer Objects since we do not require persistent storage. Frame Buffer Objects are simple data structures which host an individual frame to draw at a later date. This way we can store the users canvas, as well as any UI elements the user wants to interact with. The user can also use the Hard Drive to save images directly.

For our View we used the TouchPointsApp class. This class interacts with the user by drawing using an OpenGL Context constantly. In addition the TouchPointsApp class controls most of the user interactions. It contains interactions using the Leap Motion device, the Real Sense, the EyeX, as well as multitouch interactions. This way we easily have a class that represents the view by communicating with the user and then spreading this information to our control.

Our control is the rest of the implemented classes. They take the information provided by TouchPointsApp and manipulates the Framebuffers as well as any settings inside the application. These classes include the UserInterface, the Illustrator, the Brush, the DeviceHandler, and more.

## Deployment Diagram



Deployment Diagram

## 

## Design Patterns

**Structural Pattern:**

We used a structural pattern to develop our software. This pattern involves an inheritance structure which allows an overarching class to have additional functionalities. We created new objects to perform more precise objects. For example our TouchPointsApp class communicates with the Illustrator, which is our drawing class, the illustrator then has a brush, which tells him what type of settings he should be using. The illustrator also has a TouchShapes class which tells it how to draw the different shapes.

# System Validation

While developing Multi Modal Interactive Paint we used the Bottom-up approach to testing the software. We tested the different device implementations separately. When they were prepared we then integrated them into the rest of the program. We also developed our methods separately, and when they were integrated used github to merge our development and then run integration tests to assure everything worked smoothly.

Below are the unit tests we performed throughout our applications development

**Unit Tests**

Sunny Day:

Test Purpose:

Ensure that erasing with the eraser draws ‘Transparently’ on the current Layer.

Test Setup:

1. Draw a filled circle on the screen.
2. Toggle the layer visualization menu
3. Change the layer by pressing the ‘bottom’ layer
4. Draw a different color circle on top of your previous circle
5. Change shape to eraser
6. Increase line size
7. ‘Draw’ on top of your circle

Test Output:

After ‘drawing’ with the eraser we could see the color from the first circle.

Expected Output:

After drawing with the eraser, you should be able to see the layer behind your current layer, meaning the other circles color should bleed through.

Rainy Day Tests:

Test Purpose:

After changing shapes, the Eraser no longer erases.

Test Setup:

1. Draw a filled circle with any color
2. Toggle the eraser on
3. Erase a small portion
4. Change to a circle shape
5. Draw a circle

Test Output:

Circle was successfully drawn and erased from. After changing back to the circle functionality, I could continue to draw without erasing my drawings.

Expected Output:

When you change from the eraser it should still let you draw in any color you please and any shape, no longer erasing your images.

Rainy Day Tests:

Test Case: Sporadic Leap Motion Use

Test Purpose:

Make sure you can still change shapes and colors etc. After drawing with the leap and rapidly exiting the leap motion device. (Ensure activeDrawings = 0)

Test Setup

1. Make sure the leap motion is on and leap draw is available.
2. Rapidly move your hand around the leap motion device.
3. Perform long ‘stab’ motions with your hand over the leap motion device as well as sideways movements so it draws briefly and then exits the leap motions field of vision.
4. Change shape
5. Change Color
6. Draw your new shape and color

Test Output:

Successfully changed to a circle shape

Successfully changed to color teal.

Successfully drew a teal circle.

Expected Output:

You should be able to change the shape as well as the color. Drawing should continue to work afterwards.

Sunny Day:

Test Case: Extended Touch Gesture Working

Test Purpose:

Make sure the Extended Touch Gesture is working

Test Setup

1. Ensure multitouch is working
2. Press finger to screen and hold it for about 1 second.
3. Hold finger for 3 additional seconds.

Test Output:

After holding for one second shape changed to open circle

While holding the finger down for 3 additional seconds the shapes continued to cycle through.

Expected Output:

After .75 seconds of the finger touching the screen, the shape should change.

Every .75 seconds the finger is held down afterwards continues to change the shape.

Test Case: Double Tap Gesture Working

Test Purpose:

Ensure that the double tap gesture is working.

Test Setup

1. Ensure the multitouch is working
2. Push your pointer and middle finger together
3. Make sure the middle finger is not too far above the pointer finger (Should be up to about a quarter inch above the pointer finger).
4. Tap the screen briefly with bother fingers
5. Lift fingers immedietly

Test Output:

After step 4 ‘Red’ box popped up indicating we changed color to red.

Expected Output:

Color should change.

Rainy Day Tests:

Test Case: ‘Peace Sign’ drawing.

Test Purpose:

Make sure the application doesn’t accidently detect a double tap gesture.

Test Setup

1. Ensure multitouch is working.
2. Do a ‘Peace Sign’ with your hand (split the pointer and middle finger up)
3. Attempt to draw two lines on the canvas.

Test Output:

Drew two lines on the canvas.

Expected Output:

When the fingers are separated you should draw two lines (No color change should occur)

Sunny Day:

Test Case: Cycle through Background

Test Purpose:

Ensure that the user can cycle through backgrounds.

Test Setup:

1. Ensure multitouch is on.
2. Press the ‘11th’ button from the left of the buttons at the top left. (See User guide for a visual representation)
3. Continue to press the button until the background is white.

Test Output:

The user pressed the button the first time. The background became a checkerboard.

The user successfully pressed the button until the background became white again (indicating one full cycle occurred).

Expected Output:

The first button press should show a checkerboard background

Subsequent presses should change through a series of colors including Aqua, Blue, and Purple. Eventually looping back to white.

Rainy Day Tests:

Test Case: Save Transparent Background Image

Test Purpose:

Ensure the saving of images with a checkerboard background saves as an ‘alpha 0’ photo and not a checkerboard background photo.

Test Setup:

1. Change background color once.
2. Change to circle drawing
3. Enable filled shapes
4. Draw a circle
5. Press ‘n’ on the keyboard to save an image.
6. Open the image in sketchbook.
7. Attempt to change the background color.

Test Output:

After opening the file in sketchbook the user successfully could alter the background color.

Expected Output:

The background should change in sketchbook.

Test Case: Save Solid Background Image

Test Purpose:

Ensure the saving of images with a solid background saves as an ‘alpha 1’ Image which keeps the background color.

Test Setup:

1. Change background color twice.
2. Change to circle drawing
3. Enable filled shapes
4. Draw a circle
5. Press ‘n’ on the keyboard to save an image.
6. Open the image in sketchbook.
7. Attempt to change the background color.

Test Output:

After opening the file in sketchbook the user could not change the background color

Expected Output:

The background should not change in sketchbook.

Sunny Day Test:

Test Case  - Real Sense Connection

Test Purpose: To determine if user is provided feedback about Real Sense connection status.

Test Procedure: User plugs in Real Sense device and starts program. Then he or she checksUi display.

Expected Results: Yellow box should appear in UI display.

Rainy Day Test:

Test Case  - User unplugs Real Sense or device fails

Test Purpose: Test if correct feedback provided if user unplugs Real Sense or device fails.

Test Procedure: User unplugs Real Sense while application running.

Expected Results: Yellow box should disappear. If yellow box remains in Ui display test fails.

* Sunny Day Test:
  + Test Case  - Real Sense ,MultiTouch, and Leap Motion, EyeX Connected
    - Test Purpose: To determine if the correct mode is set when these four devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen the leap motion, Real Sense, and eyeX and starts the application. User performs Real Sense facial gestures. User draws with multitouch screen and presses all buttons. User pulls up UI displays with eyes.
    - Expected Results: Real Sense facial gestures should be read by Real Sense device. User should be able to draw with multitouch screen and buttons displayed in upper left hand corner and UI menu using eyeX. Radial menu should be functioning.
  + Test Case  - Real Sense ,MultiTouch, and Leap Motion Connected
    - Test Purpose: To determine if the correct mode is set when these three devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen the leap motion and Real Sense device and starts the application. User performs Real Sense facial gestures. User draws with multitouch screen and presses all buttons.
    - Expected Results: Real Sense facial gestures should be read by Real Sense device. User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Radial menu should be functioning.
  + Test Case  - Real Sense ,MultiTouch, and EyeX Connected
    - Test Purpose: To determine if the correct mode is set when these three devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen the eyeX and Real Sense device and starts the application. User performs Real Sense facial gestures. User draws with multitouch screen and presses all buttons. User pulls up UI screens using eyes.
    - Expected Results: Real Sense facial gestures should be read by Real Sense device. User should be able to draw with multitouch screen and buttons displayed in upper left hand corner if eyes look in upper left hand of canvas. Radial menu should be functioning.
  + Test Case  - Real Sense, Leap Motion, and EyeX Connected
    - Test Purpose: To determine if the correct mode is set when these three devices are connected at the start of the application.
    - Test Procedure: User plugs in the Real Sense device, the leap motion and eyeX and starts the application. User performs Real Sense facial gestures. User draws with Leap motion.
    - Expected Results: Real Sense gestures should be read by Real Sense device. User should be able to draw with Leap Motion and buttons displayed in upper left hand corner. Proximity menu and Radial menus should not be functioning. UI menu only show if user uses eyeX and looks in specified location.
  + Test Case  - Real Sense and Leap Motion connected
    - Test Purpose: To determine if the correct mode is set when leap motion and Real Sense device are connected at the start of the application.
    - Test Procedure: User plugs in the Real Sense device and the leap motion and starts the application. User performs Real Sense facial gestures.  User draws with Leap motion.
    - Expected Results Real Sense gestures should be read by Real Sense. User should be able to draw with Leap motion. Buttons in upper left hand corner should not be displayed. Both radial menu and proximity menus should not be active.
  + Test Case  - Real Sense and MultiTouch connected
    - Test Purpose: To determine if the correct mode is set when Real Sense device and multitouch devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen and Real Sense starts the application. User draws with multitouch screen and uses radial button. User performs Real Sense facial gestures.
    - Expected Results: User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Radial menu should be functioning. Real Sense device should be able to read facial gestures.
  + Test Case  - Real Sense and EyeX connected
    - Test Purpose: To determine if the correct mode is set when eyeX and Real Sense device are connected at the start of the application.
    - Test Procedure: User plugs in the Real Sense device and EyeX starts the application. User turns on UI menu with eyes. User draws with Real Sense device and performs Real Sense facial gestures.
    - Expected Results: User should be able to draw with Real Sense device and Real Sense facial gestures should be read. Buttons in upper left hand corner should not be active. EyeX should be able to turn on UI with eye gaze in specified region.
  + Test Case  - Real Sense connected
    - Test Purpose: To determine if the correct mode is set when only Real Sense is connected at the start of the application.
    - Test Procedure: User plugs in Real Sense device and starts the application. User tries to draw with Real Sense and perform real Sense Facial Gestures.
    - Expected Results: User should be able to draw with Real Sense and facial gestures should be read. The UI menu should be present and showing at all times and buttons in left hand corner should not be visible or active.
* Rainy Day Test:
  + Test Case  - Real Sense, MultiTouch, Leap Motion, and EyeX Connected User tries to Draw with leap Motion
    - Test Purpose: To determine if the user can make leap gestures when real sense device is connected.
    - Test Procedure: User plugs in the multitouch screen the leap motion and eyeX, Real Sense and starts the application. User then tries to draw with the leap motion.
    - Expected Results: User should not be able to draw with leap motion since the default mode specifies this when all four devices are connected.
  + Test Case  - Real Sense, MultiTouch and Leap Motion Connected User tries perform leap motion gestures.
    - Test Purpose: To determine if the user can perform leap motion gestures when default mode is set for specified devices
    - Test Procedure: User plugs in the multitouch screen the leap motion and Real Sense and starts the application. User then tries to perform leap motion gestures.
    - Expected Results: Leap motion should not be able to read leap gestures since the default mode specifies this when real sense and leap are connected.
* Sunny Day Test:
  + Test Case  - Real Sense Hover Zone
    - Test Purpose: Test if a user can enter the hover zone and provided feedback of finger location.
    - Test Procedure: User starts application. Then slowly moves right hand with index finger extended toward screen. Then observes results. This procedure is repeated with left hand.
    - Expected Results: As user’s right hand moves toward canvas a green circle should appear once hover zone is entered by user. Same should occur when left hand is used.
  + Test Case  - Real Sense Hover Zone Canvas Coverage
    - Test Purpose: Test the coverage of hover zone and whether all parts of canvas can be hovered over.
    - Test Procedure: User starts application. Then slowly moves right hand with index finger extended toward screen. Once enters hover zone he or she moves hand in all directions and covers entire canvas.
    - Expected Results: As user’s right hand moves toward canvas a green circle should appear once hover zone is entered. When user moves hand around canvas the green circle should follow the hand and every part of the canvas should be accessible.
  + Test Case  - Real Sense Draw Zone
    - Test Purpose: Test if a user can enter the draw zone and provided feedback.
    - Test Procedure: User starts application. Then slowly moves right hand with index finger extended toward screen. He or she enters hover zone and then continues to move toward screen until draw zone has been entered. Then observes results. This procedure is repeated with left hand.
    - Expected Results: As user’s right hand moves toward canvas a green circle should appear once hover zone is entered by user. Once user enters draw zone green circle should disappear and shape currently selected should appear on canvas at specified location. Same should occur when left hand is used.
  + Test Case  - Real Sense Draw Zone Canvas Coverage
    - Test Purpose: Test the coverage of draw zone and whether all parts of canvas can be drawn to.
    - Test Procedure: User starts application. Then slowly moves right hand with index finger extended toward screen. Once enters hover zone he or she moves further towards screen until draw zone entered. User should then attempt to draw on all parts of canvas.
    - Expected Results: As user’s right hand moves toward canvas a green circle should appear once hover zone is entered. User should then see circle disappear when draw zone entered. As user attempts to draw in all sections of canvas current shape should be drawn.
* Rainy Day Test:
  + Test Case  - User exits Real Sense boundaries while in Hover Zone
    - Test Purpose: Test if user exits boundaries while in hover zone expected result will occur.
    - Test Procedure: User enters hover zone and begins to move around canvas. He or she accidently leaves real sense boundary and observes result.
    - Expected Results: When user leaves real sense boundary the green circle should disappear. If green circle remains or program experiences any non-expected actions then test fails.
  + Test Case  - User exits Real Sense boundaries while in Draw Zone
    - Test Purpose: Test if user exits real sense boundaries while in draw zone expected result will occur.
    - Test Procedure: User enters draw zone and begins to draw on canvas.

He or she accidently leaves real sense boundary and observes result.

* + - Expected Results: When user leaves real sense boundary drawing should stop immediately. If program experiences any non-expected actions then test fails.

Sunny Day:

Test Case: EyeX UI Icons.

Test Purpose:

Ensure the program uses the icons when the EyeX enables or disables certain elements of the UI.

Test Setup

1. Start Program with EyeX
2. Gaze at the top left
3. Gaze at the bottom right
4. Gaze at the top left.

Test Output:

When I looked at the top left I could see all the icons. Looking at the bottom right I could see the mode box (and settings button).

Expected Output:

Gazing to the top left should enable the mode buttons to turn on as well as display some of the new icons.

The bottom right gaze you should be able to see the settings button inside the mode box.

Rainy Day Tests:

Test Case: Rapid Circle Drawing Leap

Test Purpose:

Make sure the buttons and icons reappear when we turn off and on the UI with the Radial Menu.

Test Setup:

1. Have multitouch plugged in.
2. Double tap to pull up radial menu.
3. Tap the bottom button
4. Tap the bottom button

Test Output:

Ui turned off at step 3.

Ui and all icons turned back on step 4.

Expected Output:

After tapping the button twice the icons should still be visible (not just empty boxes).

Sunny Day:

Test Case: Open UI Menus

Test Purpose:

Ensure the UI elements can still open up and display all the correct buttons.

Test Setup

1. Press Colors button
2. Press Shapes Button
3. Press Device modes button
4. Press Settings Button
5. Repeat steps 1-4

Test Output:

The first repetition of step 1-4 resulted in opening all appropriate menus.

Repeating the steps closed them all.

Expected Output:

The menus should open up just as they did before this user story. The colors buttons, shapes button, device modes button, and settings button should all toggle open or closed with a button pressed.

Rainy Day Tests:

Test Case: All Menus Open Performance Test

Test Purpose:

Ensure that the UI changes have improved performance, even when trying to open all of our menus!

Test Setup:

1. Press settings button
2. Press ‘Frames Per Second’
3. Press settings button
4. Record FPS
5. Press Settings button
6. Press Colors Button
7. Press Shapes Button
8. Press Device Modes Button
9. Record FPS

Test Output:

Step 4 : FPS 49~

Step 9 : FPS 45~

Expected Output:

FPS Drop should be substantially lower (Pre-User story implementation frame drop was around 30~ FPS). About a 4-5 FPS drop should be normal.

Sunny Day :

User selects first layer

Nothing happens.

User Selects second layer

First layer is ‘swapped’ with second layer

User draws

Draws successfully on new ‘first layer’

User selects third layer

First layer is ‘swapped’ with third layer.

User Draws

Draws successfully on new ‘first layer’

User Draws

Drawing is successfully reflected inside the layer buttons.

Rainy Day Tests :

* User clears the screen
  + Layers are still visible behind one another. Did not just fill the background with black.

Sunny Day:

Test Case: Save Transparent Image

Test Purpose:

Ensure alpha preservation on save.

Test Setup:

1. Draw large filled Green Square to cover half the screen.
2. Draw a Filled Red Circle.
3. Press the alpha reduce button
4. Draw a filled Red Circle in a new location
5. Repeat step 3 through 4, about 3 to 4 times.
6. Press ‘n’ to save the image.
7. Open the image in sketchbook
8. Change background color to blue.

Test Output:

Opening the image in sketchbook appeared with a white background.

Altering the ‘background’ in sketchbook changed the background of all areas that were not drawn in.

Expected Output:

Any area in the picture not drawn on should appear ‘white’ in the thumbnail and when opened with Sketchbook.

After changing the background color in sketchbook, any area that was not drawn on should be blue by the end of the test.

Sunny Day:

Test Case: Toggle multitouch

Test Purpose:

Ensure that touching the multitouch device override device button correctly turns off the multitouch functionality as well as turns on (if the device is actually plugged in).

Test Setup:

1. Have multitouch plugged in
2. Press the ‘Device Modes’ button in the mode box
3. Press the ‘Multitouch’ button (toggles it off)
4. Try to draw
5. Try double tap gesture
6. Try to select buttons on the top right (Colors, shapes)
7. Press the ‘Multitouch’ button (toggle it on)
8. Try to draw
9. Try to select buttons on top right.
10. Try double tap gesture.

Test Output:

After toggling multitouch off the user could not draw or touch any buttons, nor could they do the double tap gesture. Once they toggle the button back on, the user successfully drew. They could select the color button and change color. They could ‘double tap’ to pull up radial menu.

Expected Output:

When the multitouch is ‘off’ you cannot draw with the multitouch, cannot select any UI elements (except the Device Modes buttons), canot do a multitouch gesture.

When the multitouch is on you can draw with the multitouch, can select top right button, and can perform multitouch gestures.

Test Case: Toggle Leap Motion

Test Purpose:

Ensure that disabling leap motion turns off both leap draw and leap gesture (device no longer performs any actions). While turning it on (while device is connected) will allow you to use the enabled functions.

Test Setup:

1. Open Device Mode menu
2. Make sure leap motion is toggled on
3. Toggle on Leap Draw
4. Draw with leap motion
5. Toggle on Leap Gesture
6. Swipe gesture with leap motion
7. Toggle off Leap motion
8. Try to draw with leap motion
9. Try to do a swipe gesture with leap motion

Test Output:

The first draw with leap motion successfully drew.

The first swipe gesture successfully read and changed colors

After toggling off Leap Motion

The next draw attempt failed.

The new gesture attempt failed.

Expected Output:

When the leap motion is on, it should be able to perform any leap motion functionality that is also on.

* If Leap Motion and Leap Gesture is on it should detect leap gesture
* If leap motion and leap draw is on it should be able to draw
* If leap motion is off and leap draw is on it will NOT draw
* If leap motion is off and leap gesture is on it will NOT detect gestures

Test Case: Toggle Leap Draw

Test Purpose:

Ensure that the leap draw button turns on and off leap draw functionality

Test Setup:

1. Open Device Modes menu
2. Ensure Leap motion is on.
3. Toggle on Leap Draw
4. Attempt to draw with leap draw
5. Toggle off leap draw
6. Attempt to draw with leap
7. Toggle on leap draw
8. Attempt to draw with leap draw

Test Output:

Step 4: Draw was successful

Step 6: Could not draw with leap motion

Step 8: Could successfully draw with leap motion.

Expected Output:

When leap motion and leap draw are on the user should be able to draw with the leap device. When leap motion is on and leap draw is disabled, they should not be able to draw with the leap motion.

Test Case: Toggle Leap Gesture

Test Purpose:

Test Setup:

1. Open Device Modes menu
2. Ensure Leap motion is on.
3. Toggle on Leap Gesture
4. Attempt to do a swipe gesture
5. Toggle off leap gesture
6. Attempt to do a swipe gesture
7. Toggle on leap gesture
8. Attempt to do a swipe gesture

Test Output:

Step 4: Changed color with leap swipe gesture

Step 6: Could not perform a swipe gesture

Step 8: Could successfully perform a swipe gesture.

Expected Output:

When leap motion is on and leap gesture is on, the user can perform leap motion gestures. When leap gesture is off, the program will not detect any leap gestures.

Test Case: Toggle EyeX

Test Purpose:

Make sure the EyeX ‘Menu Grab’ functionality is turned off when eyeX is enabled and is off when EyeX is disabled.

Test Setup:

1. Pull up Device Modes menu
2. Toggle EyeX off
3. Toggle EyeX on
4. Look to the top right corner of the screen
5. Look to the bottom right corner of the screen
6. Toggle EyeX Off

Test Output:

After toggling off EyeX the menus were pulled up.

After eyeX was toggled on, the menus disappeared.

After Looking at the top right the buttons came up

After Looking at the bottom right the mode box came up

Expected Output:

When EyeX is on, the UI gets pulled up only when looking at those UI elements (Top left for mode buttons, bottom right for mode box).

When EyeX is disabled the UI elements should always be available.

Rainy Day Tests:

Test Case: Toggle Disconnected Devices.

Test Purpose:

Make sure you cannot ‘enable’ a device when it is not connected.

Test Setup:

1. Start the program with only multitouch plugged in
2. Toggle the Device Modes menu
3. Tap Leap Motion Button
4. Tap Leap Draw button
5. Tap Leap Gesture button
6. Tap EyeX button

Test Output:

* After tapping the leap motion button the box stayed black
* After tapping the leap draw button the box turned green
* After tapping the leap gesture button the box turned green
* After tapping the EyeX button the box stayed black

Expected Output:

You cannot enable a device that is not plugged in. Only leap draw and Leap Gesture should get a green box (functionality is on, but cannot be performed since leap motion is not ‘on’), while EyeX and Leap Motion will not turn on (Black Box)

Sunny Day :

Test Purpose:

To ensure layers successfully change alpha color when using the slider.

Test Setup

1. Draw a filled circle with any color
2. Open Layer Visualization menu
3. Slide your finger along the slider next to the layer which your circle in drawn

Test Output:

When the finger slides upward, the image becomes more transparent. When the finger slides down the image becomes less transparent (more visible).

Expected Output:

When the finger is placed on the slider it should change the alpha depending on where the touch occurred, higher on the scale means it is more transparent, lower on the scale makes it more visible.

Rainy Day Tests:

Test Purpose:

Ensure altering alpha color with slider only affects that one specific layer.

Test Setup:

1. Draw a filled circle with any color
2. Open Layer Visualization menu
3. Switch layers by touching the bottom or middle layer.
4. Draw a filled circle at a separate location from the first.
5. Slide your finger along the slider next to the layer which either circle in drawn on.

Test Output:

When the user slides their finger along the slider, only one of the two circles are affected.

Expected Output:

Only the circle within the layer you are doing the ‘sliding’ next to should be affected.

Sunny Day:

Test Case: Leap Single Line Draw

Test Purpose:

Ensure the program runs smoothly while drawing a single line with leap inputs, as well as testing device disconnect and connect.

Test Setup

1. Start Program
2. Open Device Modes menu
3. Plug in Leap Motion wait ‘x’ seconds
4. Unplug Leap Motion wait ‘x’ seconds
5. Plug in leap motion
6. Enable Leap Draw
7. Attempt to draw a single line with leap motion.

Test Output:

Devices statuses were properly updated.

Line was drawn smoothly

Expected Output:

Plugging in the leap motion should toggle on the ‘Leap Motion Device’

Unplugging it should turn off the device

Plugging it back in should turn off the device.

Drawing should be fairly smooth and fast  (But ‘Jumpy’ because leap motion input is not as precise).

Test Case: Multitouch Single Line Draw

Test Purpose:

Ensure the program runs smoothly while drawing a single line with multitouch inputs, as well as testing device disconnect and connect.

Test Setup

1. Start Program
2. Open Device Modes menu
3. Plug in Multitouch wait ‘x’ seconds
4. Unplug Multitouch wait ‘x’ seconds
5. Plug in Multitouch
6. Attempt to draw a single line with multitouch.

Test Output:

Devices statuses were properly updated.

Line was drawn smoothly.

Expected Output:

Plugging in the Multitouch should toggle on the ‘Multitouch’

Unplugging it should turn off the device

Plugging it back in should turn on the device (green box).

Drawing should be smooth and fast.

Rainy Day Tests:

Test Case: Rapid Circle Drawing Leap

Test Purpose:

Make sure circles are still drawn very smooth (Smoothly grow larger and smaller) and do not seem to make large leaps in radius changes using leap motion.

Test Setup:

1. Make sure leap device is connected and ‘leap draw’ is enabled.
2. Change to circle mode
3. Change to filled shapes
4. Move 5 fingers into the draw zone
5. Move around the hand
6. Leave the draw zone

Test Output:

Drew circles. No huge jumps in radius between frames. Growth and shrinking of circles were smooth.

Expected Output:

Circles should expand and retract smoothly without huge jumps in radius between each re-draw.

Test Case: Rapid Circle Drawing Multitouch

Test Purpose:

Make sure circles are still drawn very smooth (Smoothly grow larger and smaller) and do not seem to make large leaps in radius changes using multitouch.

Test Setup:

1. Make sure multitouch is connected.
2. Change to circle mode
3. Change to filled shapes
4. Place 5 fingers on the touch screen.
5. Move around your fingers on the screen.
6. Release your fingers.

Test Output:

Drew circles. No huge jumps in radius between frames. Growth and shrinking of circles were smooth.

Expected Output:

Circles should expand and retract smoothly without huge jumps in radius between each re-draw.

Sunny Day:

Test Case: Toggle Real Sense

Test Purpose:

Ensure that touching override Real Sense button correctly turns off the real sense functionality as well as turns on (if the device is actually plugged in).

Test Setup:

1. Have real sense plugged in
2. Press the ‘Device Modes’ button in the mode box
3. Press the ‘Real Sense’ button (toggles it off)
4. Try to perform expressions
5. Press the ‘Real Sense’ button (toggle it on)
6. Try to do Expressions.

Test Output:

After toggling the real sense off the user could perform any sort of expressions to change settings. In addition when toggling off the real sense the ‘Real Sense Expressions’ button also was toggled off. Once they toggle the button back on, the user successfully could perform expressions. They could change the shape with a smile, as well as change shape with a puffy cheek.

Expected Output:

When the real sense is ‘off’ you cannot use expressions to change brush settings. In addition it should ‘shut down’ the real sense expressions button.

When the Real Sense is on you can use expressions to change shape, as well as color.

Sunny Day:

Test Case: Toggle Real Sense Expressions

Test Purpose:

Ensure that touching the Real Sense Expressions override button correctly turns off the real sense functionality as well as turns on (if the device is actually plugged in).

Test Setup:

1. Have real sense plugged in
2. Press the ‘Device Modes’ button in the mode box
3. Draw 3 simple shapes (Lines, circles, or rectangles)
4. Make sure the ‘Real Sense’ button is on and green.
5. Press the ‘Real Sense Expressions’ button (toggles it off)
6. Try to perform expressions
7. Press the ‘Real Sense Expressions’ button (toggle it on)
8. Try to do Expressions.

Test Output:

After toggling the real sense off the user could perform any sort of expressions to change settings. Once they toggle the button back on, the user successfully could perform expressions. They changed the shape with a smile, as well as change shape with a puffy cheek. When doing a raised eyebrow the device called the undo function on our shapes.

Expected Output:

When the real sense expressions is ‘off’ you cannot use expressions to change brush settings.

When the Real Sense Expression was on is on you can use expressions to change shape, as well as color.

Rainy Day Tests:

Test Case: Toggle Disconnected Real Sense

Test Purpose:

Make sure you cannot ‘enable’ a device when it is not connected.

Test Setup:

1. Start the program with only multitouch plugged in
2. Toggle the Device Modes menu
3. Tap ‘Real Sense’ button
4. Tap the ‘Real Sense Expressions’ button.

Test Output:

* After tapping the Real Sense button the box stayed black
* After tapping the Real Sense Expressions button the box stayed black.

Expected Output:

No boxes should turn black when tapping on them, since the device is not plugged in.

Sunny Day Tests

Test Case: Draw a Rectangle

Test Purpose:

Ensure we can draw a rectangle shape

Test Setup:

1. Press the shapes button at the top left
2. Press the rectangle button
3. Drag finger across multitouch

Test Output:

After dragging finger across multitouch we drew a rectangle

Expected Output:

Dragging your finger should create a rectangle.

Rainy Day Tests

Test Case: Return to line draw

Test Purpose:

Ensure we can change back to drawing lines

Test Setup:

1. Press Shapes buttons at the top left
2. Press rectangle button
3. Press line button
4. Drag finger across multitouch

Test Output:

Drew a line

Expected Output:

Should draw a line, not a rectangle.

Sunny Day Tests

Test Case: Draw a Triangle

Test Purpose:

Ensure we can draw a Triangle shape

Test Setup:

1. Press the shapes button at the top left
2. Press the Triangle button
3. Drag finger across multitouch

Test Output:

After dragging finger across multitouch we drew a Triangle

Expected Output:

Dragging your finger should create a Triangle.

Rainy Day Tests

Test Case: Return to line draw

Test Purpose:

Ensure we can change back to drawing lines

Test Setup:

1. Press Shapes buttons at the top left
2. Press Triangle button
3. Press line button
4. Drag finger across multitouch

Test Output:

Drew a line

Expected Output:

Should draw a line, not a Triangle.

Sunny Day:

Test Case: Draw Circle

Test Purpose:

Ensure that drawing a circle is possible.

Test Setup:

1. Press the ‘Shapes Button’
2. Press the Circle Shape
3. Press a finger to the multitouch
4. Drag Finger

Test Output:

Drew a circle shape.

Expected Output:

A black circle should be drawn based on how far you dragged your finger.

Rainy Day Tests:

Test Case: Draw lines after circles

Test Purpose:

Ensure we can change from one shape to another without any hiccups.

Test Setup:

1. Drag finger across multitouch
2. Press ‘Shapes Button’
3. Press ‘Circle’ button
4. Drag finger across multitouch
5. Press shapes button
6. Press Line button
7. Drag finger across multitouch.

Test Output:

Expected Output:

The user should have drawn one line first, then a circle, then another line.

Sunny Day:

Test Case: Toggle FPS

Test Purpose:

Ensure we can toggle FPS on.

Test Setup

1. Start Program
2. Open settings menu.
3. Press ‘Frames Per Second’
4. View FPS at top left corner

Test Output:

The top right corner displayed ‘Framerate: 49’ and changed slightly every once in a while.

Expected Output:

It should display the words ‘Framerate’ in the top right corner, followed by a number that will chance frequently.

Rainy Day Tests:

Test Case: Toggle Off FPS

Test Purpose:

Make sure the user can toggle off FPS if he doesn’t want it anymore.

Test Setup:

1. Start application
2. Press Settings button
3. Press ‘Frames per second’
4. Press ‘Frames Per second’ again.

Test Output:

After step 3 the FPS showed up. After step 4 it disappeared.

Expected Output:

Every touch of the ‘Frames Per Second’ button will toggle the appearance of the FPS counter. 3 Should bring it up, while 4 will make it go away.

Sunny Day Tests

Test Case: EyeX Shape Change

Test Purpose:

Ensure we can change shapes using the eyex and the spacebar.

Test Setup:

1. Begin the program with the EyeX Plugged in.
2. Move your gaze to the top right
3. Press space
4. draw
5. Move your gaze to the bottom right
6. Press space
7. Draw
8. Move your gaze to the bottom left
9. Press space
10. Draw
11. Move your gaze to the top right
12. Press space
13. draw

Test Output:

Drew 4 different shapes. A shape change occurred every space press.

Expected Output:

You should be able to draw 4 different shapes. Line, circle triangle, and rectangle

Rainy Day Tests

Test Case:

Test Purpose:

Ensure space doesn’t change shape when EyeX isn’t active

Test Setup:

1. Start program without EyeX
2. Look at the top right
3. Press Space
4. Draw

Test Output:

Drew a line

Expected Output:

You should not change shape, you should draw a line only.

Sunny Day Tests

Test Case: Change Colors

Test Purpose: Ensure that we can change colors with the multitouch color change buttons.

Test Setup:

1. Press the Color wheel button at the top left corner.
2. Press the first color
3. Draw a Line
4. Repeat step 1 through 3 until you try every color

Test Output:

Every line drawn correctly corresponded to the proper color

Expected Output:

After pressing a multitouch color button the following shapes drawn should be in that corresponding color.

Rainy Day Tests

Test Case: Shapes in Different Colors

Test Purpose:

Ensure that the different colors work with different shapes

Test Setup:

1. Press the color wheel button at the top left corner
2. Press the 3rd color
3. Draw a line.
4. Press the shapes button and change to circle
5. Draw a circle

Test Output:

Both the line and circle were yellow.

Expected Output:

The line and circle should be the same color.

## 

Sunny Day Tests

Test Case: Change Background Color

Test Purpose:

Ensure that we can change background color.

Test Setup:

1. Press the ‘cog’ in the mode box (bottom right).
2. Press the ‘Cycle Background button’
3. Continue pressing until background is white.

Test Output:

Every press of ‘Cycle Background’ changed the background color.

Expected Output:

The background should cycle through various colors, and eventually return to white.

Rainy Day Tests

Test Case: Save Images with background.

Test Purpose:

Make sure the background saves with the drawing.

Test Setup:

1. Draw a few lines
2. Change background
3. Press ‘n’ to save image.
4. Repeat step 2 – 3 until you return to the white background.

Test Output:

Each saved images had a different background (except transparent appeared white)

Expected Output:

Every background should have saved a different background color.

Sunny Day Tests

Test Case: Change Layers

Test Purpose:

Ensure each layer is separated.

Test Setup:

1. Change to filled circles
2. Draw a circle
3. Change color to red
4. Open layer menu
5. Press second layer
6. Draw a red circle on top of your black circle
7. Press the original layer

Test Output:

After drawing the red circle it was on top of the black circle. When we pressed the layer with the ‘black’ circle it then was on top.

Expected Output:

After drawing a red circle it should be on top. When you swap layers, the black circle should be on top.

Rainy Day Tests

Test Case: Swapping layers

Test Purpose:

Make sure we switch to the proper layer.

Test Setup:

1. Draw a filled Red circle
2. Change to an empty layer
3. Draw a blue circle that overlaps with the red circle
4. Change to an empty layer
5. Draw a Green circle that overlaps with both the red and blue circle.
6. Change to layer with Red circle
7. Change to layer with blue circle
8. Change to layer with green circle.

Test Output:

When I changed to red circle layer, the red moved to the top.

When I changed to blue circle layer, the blue circle moved to the top.

When I changed to the green circle layer, the green circle moved to the top.

Expected Output:

Whichever layer you select, the layer with that specific color should be on top.

Sunny Day :

* Started application with all 3 devices plugged in
  + Successfully drew all 3 color coordinated boxes.

Rainy Day Tests :

* Started Application with leap and multitouch but without Tobii EyeX
  + Drew the Leap Motion and multitouch squares successfully. Did not draw Red EyeX Box.
* Start App. with Multitouch, no leap or eyeX
  + Drew blue box. Did not draw leap or EyeX box.
* Start App without any devices
  + Drew no boxes inside the mode box.
* Start App with Leap only
  + Only drew green box.
* Start App with EyeX Only
  + Only drew red box.
* Start app without Multitouch, but with Leap and EyeX
  + Drew green and red box.

Sunny Day :

Tap the screen twice in quick repetition (same finger, same location)

Successfully called up radial menu, no leftover drawing residue.

Rainy Day Tests :

Tap the screen very fast with two fingers, far away from each other

Did not read a ‘double tap’ because they were too far.

No residue

Double tap, then drag finger on second tap

Successfully read the double tap (pulled up radial menu)

Did not ‘draw’ when dragging the second finger.

Sunny Day Tests

Test Case: Pull up radial menu

Test Purpose:

Make sure the double tap works

Test Setup:

1. Tap twice in the same location

Test Output:

A radial menu popped up at the location I tapped.

Expected Output:

A radial menu should pop up where you tapped the multitouch device.

Rainy Day Tests

Test Case: Two far away taps.

Test Purpose:

Make sure the double tap doesn’t happen when you tap far apart.

Test Setup:

1. Tap on the left side of the screen and right side at the same time

Test Output:

Nothing happened.

Expected Output:

Nothing should occur.

Test Case : Double tap and drag

Test Purpose:

Make sure that when you perform a double tap you can’t draw as well.

Test Setup:

1. Perform a double tap, but keep finger down on the second tap.
2. Drag finger

Test Output:

No lines were drawn.

Expected output:

No drawings should occur when you drag your finger after the radial menu popped up.

Sunny Day Tests

Test Case:  Symmetry Line – Draw Line

Test Purpose:

Check to see if the user can successfully draw a line symmetroic to our ‘Symmetry Line’

Test Setup:

1. Tap the symmetry line button
2. Draw a few lines

Test Output:

Drew a line opposite our line of symmetry whenever we drew lines.

Expected Output:

Two lines should be drawn for every line you draw. It should be opposite our line of symmetry.

Rainy Day Tests

Test Case: Symmetry Line – Turn off Symmetry

Test Purpose:

Ensure Symmetric lines don’t draw when it is turned off.

Test Setup:

1. Toggle on symmetry
2. Toggle off symmetry
3. Draw lines

Test Output:

Lines were only drawn where I dragged my finger.

Expected Output:

Only a single line should be drawn per finger input.

Sunny Day Tests

Test Case: Draw in a circle pattern

Test Purpose:

Ensure the lines aren’t broken and are ‘smooth’

Test Setup:

1. Increase line size by pressing ‘x’ 5 times.
2. Drag your finger on the multitouch in a circular pattern

Test Output:

Lines were smooth and not ‘broken’ up (No breaks in the lines).

Expected Output:

The lines you create should not be broken up, they should be smooth and connected to itself.

Sunny Day :

Updated :

After failure to erase previous shapes we now have a working test.

* ‘Gaze’ at the bottom right of the screen
  + Mode box successfully appeared.
* Change to line draw mode from circle mode
  + Successfully drew all 3 lines. Removed circle.
* Change to Circle draw mode.
  + Drew the circle
* Enable fill mode
  + Circle became full
* Change color
  + Correctly changed through colors
* Draw with red circle in mode box
  + Successfully drew red circle in the canvas.
* Enable eraser mode
  + Correctly made the proper sized eraser in the mode box.
* Switch to triangle mode
  + Correctly drew the triangle
* Switch to rectangle mode
  + Correctly drew rectangle in mode box.
* Switch to non-filled shapes
  + Rectangle correctly unfilled
* Change line size
  + Line size Dynamically changed.
* Look away from bottom right
  + Mode box successfully shut off.

Rainy Day Tests :

* Hold down color change key
  + Correctly runs through all colors correctly. Draws with the same color as mode box at the end of the key down.
* Hold down line size key
  + Lines grew dynamically and correctly.
* Press button while EyeX gaze is not looking at them
  + Buttons did not activate changing shapes or colors.

Sunny Day :

User increases Alpha

Successfully made color less ‘transparent’ based on current color

User decreases Alpha

Successfully made color more transparent based on background color and current color

Rainy Day Tests :

* Decrease Alpha, then change shape and color
  + Alpha successfully changed even after changing shape and color
* Increase Alpha, then change shape and color
  + Alpha successfully changed even after swapping shape and color

Sunny Day:

Test Case: Keyboard Key Test

Test Purpose:

Ensure that all the keys function properly.

Test Setup

1. Press the ‘Text’ Button
2. Press a location on the screen to write text.
3. Press every button on the keyboard except Shift, Font, Enter,  and ‘Done’. (Starting from the top left, continue all the way right. Then the next line starting from the left).
4. Press ‘Font’ four times to change through the fonts.
5. Press Shift.
6. Press Enter
7. Repeat step 3.
8. Press outside the keyboard to shut it off.

Test Output:

Pressing Font changed fonts

Pressing Shift changed some keys

Pressing Enter made the new typing happen on a new line.

The ending string was

“`123456789 qwertyuiop\asdfghjkl;zxcvbnm,./

~!@#$%^&\*( QWERTYUIOP|ASDFGHJKL:ZXCVBN<>?”

After tapping outside the keyboard it disappeared and I could draw again.

Expected Output:

Pressing font should cycle through the font.

Pressing shift should change the key types

Pressing Enter should do ‘New Line’

The resulting string should be

“`123456789 qwertyuiop\asdfghjkl;zxcvbnm,./

~!@#$%^&\*( QWERTYUIOP|ASDFGHJKL:ZXCVBN<>?”

Rainy Day Tests:

Test Case: Backspace on Enter

Test Purpose:

Ensure that backspace is working properly. It is properly deleting strings, as well as ‘Enter’ keys, and not breaking when ‘backspacing’ nothing.

Test Setup:

1. Press the Text Button
2. Press where you want the text
3. Type ‘asdf’
4. Press ‘Enter’
5. Type ‘qwer’
6. Backspace 4 times
7. Type ‘qwer’
8. Backspace 5 times
9. Type ‘qwer’
10. Backspace 12 times.

Test Output:

Step 3

“asdf”

Step 4

“asdf”

Step 5

“asdf

qwer”

Step 6

“asdf”

Step 7

“asdf”

qwer”

Step 8

“asdf”

Step 9

“asdfqwer”

Step 10

“”

Expected Output:

Step 3

“asdf”

Step 4

“asdf”

Step 5

“asdf

qwer”

Step 6

“asdf”

Step 7

“asdf”

qwer”

Step 8

“asdf”

Step 9

“asdfqwer”

Step 10

“”

Pressing backspace with an empty string should not crash the program.

Sunny Day Tests

Test Case: Enable Random Color Mode

Test Purpose: Ensure that random color mode is randomizing colors

Test Setup:

1. Press ‘R’
2. Draw 4 lines with multitouch
3. Change shape
4. Draw 4 shapes

Test Output:

The 4 Lines had varied colors. The 4 shapes also had varied colors.

Expected Output:

The drawings, for the most part, should have varied colors. Though they can repeat once or twice, it is unlikely.

Rainy Day Tests

Test Case: Disable Random Color Mode

Test Purpose: Ensure that random color mode can be toggled off.

Test Setup:

1. Press ‘R’
2. Draw 4 lines
3. Press ‘R’
4. Draw 4 lines

Test Output:

The first 4 lines were of varied colors, while the next 4 were all black

Expected Output:

The first 4 lines should be different colors.

The following four lines should be all a singular color.

* Sunny Day Test:
  + Test Case  - Startup Image is Displayed at Start
    - Test Purpose: To determine if the startup image automatically begins when application is started by user.
    - Test Procedure: User starts application and observes feedback from screen.
    - Expected Results: Application should start up with image being shown in the middle of the canvas. Image should display name of application.
  + Test Case  - Startup Image Fades Out
    - Test Purpose: To determine if the startup image fades out completely after specified amount of time.
    - Test Procedure: User starts application and observes canvas feedback.
    - Expected Results: Application should start up with image being shown in the middle of the canvas. Image should slowly fade out and be completely gone after 5 seconds.
* Rainy Day Test:
  + Test Case  - User accidently Starts Application
    - Test Purpose: To determine if User will be informed application has been started if he or she accidently starts application.
    - Test Procedure: User starts application without knowing application has been start.
    - Expected Results: Application should start up with image being shown in the middle of the canvas and slowly fades out. If user doesn’t see image appear test fails.
* Sunny Day Test:
  + Test Case  - User Increase Line or Shape Size
    - Test Purpose: To determine if the brush button works properly when a person wants to increase the line or shape size.
    - Test Procedure: User draws one line. He or she then presses the brush button. User presses plus button next to line size and draws a second line. He or she then presses plus button again and draws a third line.
    - Expected Results: User should see a line drawn that is the default line size. After brush button is pressed three more buttons should appear underneath the brush button. After second line is drawn user should see a line bigger than the first. After third line is drawn user should see a line bigger than the first and second.
  + Test Case  - User Decrease Line or Shape Size
    - Test Purpose: To determine if the brush button works properly when a person wants to decrease the line or shape size.
    - Test Procedure: User presses the brush button. User presses plus button three times in a row and draws a line. User then presses minus and draws a second line. He or she then presses minus button again and draws a third line.
    - Expected Results: After brush button is pressed three more buttons should appear underneath the brush button. After first line drawn user should see a line drawn that is three sizes bigger than the default line size. After second line is drawn user should see a line smaller than the first. After third line is drawn user should see a line smaller than the first and second.
  + Test Case  - User Increase Alpha of Line or Shape Size
    - Test Purpose: To determine if the brush button works properly when a person wants to increase alpha of line or shape size.
    - Test Procedure: User presses the brush button. User presses minus button next to alpha button three times in a row and draws a line. User then presses plus button and draws a second line. He or she then presses plus button again and draws a third line.
    - Expected Results: After brush button is pressed three more buttons should appear underneath the brush button. After first line drawn user should see a line that is less opaque than the default line or shape. After second line is drawn user should see more opaque line or shape than the first. After third line is drawn user should see a more opaque than the first and second.
  + Test Case  - User Decrease Alpha of Line or Shape Size
    - Test Purpose: To determine if the brush button works properly when a person wants to decrease the alpha of line or shape size.
    - Test Procedure: User draws one line. He or she then presses the brush button. User presses minus button next to alpha line size and draws a second line. He or she then presses minus button again and draws a third line.
    - Expected Results: User should see a line drawn that is the default alpha setting for line or shape. After brush button is pressed three more buttons should appear underneath the brush button. After second line is drawn user should see a less opaque line or shape than the first. After third line is drawn user should see a less opaque line or shape than the first and second.
  + Test Case  - User Turns on Filled Shapes
    - Test Purpose: To determine if the brush button works properly when a person wants to use filled shapes.
    - Test Procedure: User chooses a circle shape. He or she then presses the brush button. User presses filled shape button and draws a circle.
    - Expected Results: User should see a drawn circle that is not filled in. After brush button is pressed three more buttons should appear underneath brush button. Filled shape buttons should be green indicating filled shapes is turned on. Second circle should be on canvas that is filled in.
  + Test Case  - User Turns off Filled Shapes
    - Test Purpose: To determine if the brush button works properly when a person wants to turn filled shapes off.
    - Test Procedure: User chooses a circle shape with filled shapes turned on. He or she draws a circle and then presses the brush button. User presses filled shape button, which is currently green, and draws a circle.
    - Expected Results: User should see a drawn circle that is filled in. After brush button is pressed three more buttons should appear underneath brush button. Filled shape buttons should be black indicating filled shapes is turned off. Second circle should be on canvas that is not filled in.
* Rainy Day Test:
  + Test Case  - User Accidently Presses Box Labeled Line Size
    - Test Purpose: To determine if expected result will occur if user accidently presses Line Size box instead of plus or minus buttons.
    - Test Procedure: User draws line. User then presses brush button and then presses box that has line size label inside of it. He or she then draws a second line.
    - Expected Results: First line drawn should be a line with default size. The second line should be the same size as the first. If the second line is a different size then the first test failed.
  + Test Case  - User Accidently Presses Box Labeled Alpha
    - Test Purpose: To determine if expected result will occur if user accidently presses Alpha labeled box instead of plus or minus buttons.
    - Test Procedure: User draws line. User then presses brush button and then presses box that has Alpha label inside of it. He or she then draws a second line.
    - Expected Results: First line drawn should be alpha default value and completely opaque. The second line should also be completely opaque and default value for alpha. If the second line is less opaque then first, test failed.
  + Test Case  - User Accidently Presses Filled Shape Labeled Box
    - Test Purpose: To determine if expected result will occur if user accidently presses Filled Shape labeled box instead of button.
    - Test Procedure: User draws a none filled circle. User then presses brush button and then presses box that has filled shape label inside of it. He or she then draws a second circle.
    - Expected Results: First circle drawn should be a none filled circle. The circle drawn should also be a none filled circle. If circle filled then it failed.
  + Test Case  - User Accidently Presses Outside Boundary of Alpha, Line Size, and Filled Shapes Buttons
    - Test Purpose: To determine if expected result will occur if user accidently presses just outside of alpha, line size, or filled shapes buttons.
    - Test Procedure: User draws a  none filled circle. User then presses brush button. He or she then proceeds to press all along the boundaries of the buttons specified above.
    - Expected Results: First circle drawn should be default size, fully opaque and none filled. The second circle should be the same default size, same opaqueness, and none filled. If any of these three settings have changed test failed.
* Sunny Day Test:
  + Test Case  - Tongue Gesture Read by Real Sense
    - Test Purpose: To determine if Real Sense correctly reads tongue gesture.
    - Test Procedure: User starts program. He or she then performs a tongue gesture and holds gesture for at least one second. Observes results the feedback and then draws a line on the left side of the canvas.
    - Expected Results: After tongue gesture performed the symmetry line should appear on the canvas.  When the user begins to draw on the left side of the screen an identical line should appear on the right side of canvas.
  + Test Case  - Cheek Gesture Read by Real Sense
    - Test Purpose: To determine if Real Sense correctly reads cheek gesture.
    - Test Procedure: User starts program and draws a line with the current color. He or she then performs a cheek gesture and holds gesture for at least one second. Observes results and then draws a second line.
    - Expected Results: The user should first see a drawn line with the current color setting. After cheek gesture performed feedback should be given to the user in the form of a square with the new color setting. Once the user draws a second line the new color should be seen.
  + Test Case  - Smile Gesture Read by Real Sense
    - Test Purpose: To determine if Real Sense correctly reads smile gesture.
    - Test Procedure: User starts program and draws a line with the current shape setting. He or she then performs a smile gesture and holds gesture for at least one second. Observes results and then draws a second line.
    - Expected Results: The user should first see a drawn line with the current shape setting. After smile gesture performed feedback should be given to the user in the form of a square with the new shape setting. Once the user draws a second line the new shape should be drawn.
* Rainy Day Test:
  + Test Case  - User Accidently performs tongue gesture
    - Test Purpose: Test if accidental tongue gesture is read by Real Sense.
    - Test Procedure: User starts program and draws one line. He or she then performs a quick tongue gesture that lasts less than one second. Then User attempts to draw another line.
    - Expected Results: The first line drawn should appear in specified location. After tongue gesture nothing should happen and canvas should not be altered. When user draws a second line an identical line should not be drawn anywhere else on the canvas.
  + Test Case  - User Accidently performs cheek gesture
    - Test Purpose: Test if accidental cheek gesture is read by Real Sense.
    - Test Procedure: User starts program and draws one line. He or she then performs a quick cheek gesture that lasts less than one second. Then User attempts to draw another line.
    - Expected Results: The first line drawn should appear in specified location with current color setting. After cheek gesture nothing should happen and canvas should not be altered. When user draws a second line drawn line should be same color as previously drawn line.
  + Test Case  - User Accidently performs smile gesture
    - Test Purpose: Test if accidental smile gesture is read by Real Sense.
    - Test Procedure: User starts program and draws a shape. He or she then performs a quick smile gesture that lasts less than one second. Then User attempts to draw another shape.
    - Expected Results: The first shape drawn should appear in specified location with current shape setting. After smile gesture nothing should happen and canvas should not be altered. When user draws a second shape, the shape drawn should be same as previously drawn shape.
* Sunny Day Test:
  + Test Case  - Disconnect Multitouch Device
    - Test Purpose: To determine if the default mode is reset when multitouch device is removed while application is running.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects multitouch device. Observes feedback and interacts with application.
    - Expected Results: After multitouch is disconnected buttons in left hand corner should disappear and UI menu should remain on canvas. User should be able to perform real sense facial gestures and draw with leap motion.
  + Test Case  - Disconnect Leap Device
    - Test Purpose: To determine if the default mode is reset when leap device is removed while application is running.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects leap device. Observes feedback and interacts with application.
    - Expected Results: After leap is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform real sense facial gestures and draw with multitouch.
  + Test Case  - Disconnect Real Sense Device
    - Test Purpose: To determine if the default mode is reset when real sense device is removed while application is running.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects real sense device. Observes feedback and interacts with application.
    - Expected Results: After real sense is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform leap gestures and draw with multitouch. User should not be able to draw with leap motion.
  + Test Case  - Disconnect EyeX Device
    - Test Purpose: To determine if the default mode is reset when eyeX device is removed while application is running.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense, and eyeX connected. He or she then disconnects eyeX device. Observes feedback and interacts with application.
    - Expected Results: After eyeX is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform real sense gestures and draw with multitouch. User should not be able to draw with leap motion.
* Rainy Day Test:
  + Test Case  - Multitouch quits working user removes device
    - Test Purpose: To determine if user removes a non-functioning multitouch device the default mode is reset.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects non-functioning multitouch device. Observes feedback and interacts with application.
    - Expected Results: After multitouch is disconnected buttons in left hand corner should disappear and UI menu should remain on canvas. User should be able to perform real sense facial gestures and draw with leap motion.
  + Test Case  - Leap quits working user removes device
    - Test Purpose: To determine if user removes a non-functioning leap device the default mode is reset.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects non-functioning leap device. Observes feedback and interacts with application.
    - Expected Results: After leap is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform real sense facial gestures and draw with multitouch.
  + Test Case  - Real Sense quits working user removes device
    - Test Purpose: To determine if user removes a non-functioning real sense device the default mode is reset.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects non-functioning real sense device. Observes feedback and interacts with application.
    - Expected Results: After real sense is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform leap gestures and draw with multitouch. User should not be able to draw with leap motion.
  + Test Case  - EyeX quits working user removes device
    - Test Purpose: To determine if user removes a non-functioning eyeX device the default mode is reset.
    - Test Procedure: User starts program with multitouch, leap, and Real Sense Connected. He or she then disconnects non-functioning eyeX device. Observes feedback and interacts with application.
    - Expected Results: After eyeX is disconnected buttons in left hand corner should remain and UI menu should be on canvas if selected by using radial button. User should be able to perform real sense gestures and draw with multitouch. User should not be able to draw with leap motion.
* Sunny Day Test:
  + Test Case  - MultiTouch, Leap Motion, and EyeX Connected
    - Test Purpose: To determine if the correct mode is set when these three devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen the leap motion and eyeX and starts the application. User performs leap gestures and tries to turn on UI menu with eyes. User draws with multitouch screen and presses all buttons.
    - Expected Results: Leap gestures should be read by leap motion. User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Both radial menu and proximity menu should be functioning. UI menu only show if user uses eyeX and looks in specified location.
  + Test Case  - MultiTouch and Leap Motion connected
    - Test Purpose: To determine if the correct mode is set when leap motion and multitouch devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen the leap motion and starts the application. User performs leap gestures.  User draws with multitouch screen and presses all buttons.
    - Expected Results Leap gestures should be read by leap motion. User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Both radial menu and proximity menu should be functioning.
  + Test Case  - MultiTouch and eyeX connected
    - Test Purpose: To determine if the correct mode is set when eyeX and multitouch devices are connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen and eyeX starts the application. User turns on UI menu with eyes. User draws with multitouch screen and presses all buttons.
    - Expected Results: User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Radial menu should be functioning. EyeX should be able to turn on UI with eye gaze in specified region.
  + Test Case  - Leap Motion and eyeX connected
    - Test Purpose: To determine if the correct mode is set when eyeX and leap motion are connected at the start of the application.
    - Test Procedure: User plugs in the leap motion and eyeX starts the application. User turns on UI menu with eyes. User draws with leap motion and performs leap motion gestures.
    - Expected Results: User should be able to draw with leap motion and leap gesture should be read. Buttons in upper left hand corner should not be active. Proximity menu should be active and eyeX should be able to turn on UI with eye gaze in specified region.
  + Test Case  - MultiTouch connected
    - Test Purpose: To determine if the correct mode is set when only multitouch screen is connected at the start of the application.
    - Test Procedure: User plugs in the multitouch screen and starts the application. User draws with multitouch screen and presses all buttons.
    - Expected Results: User should be able to draw with multitouch screen and buttons displayed in upper left hand corner. Radial menu should be functioning.
  + Test Case  - EyeX connected
    - Test Purpose: To determine if the correct mode is set when only eyeX is connected at the start of the application.
    - Test Procedure: User plugs in eyeX and starts the application. User tries to turn UI menu by gazing in specified region.
    - Expected Results: User should see UI when looking in specified region and buttons in left hand corner should not be active.
  + Test Case  - Leap Motion connected
    - Test Purpose: To determine if the correct mode is set when only leap motion is connected at the start of the application.
    - Test Procedure: User plugs in leap motion and starts the application. User tries to draw with leap motion and perform leap gestures. User also tries to use proximity menu.
    - Expected Results: User should be able to draw with leap motion and gesture should be read. Proximity should function as expected. The UI menu should be present and showing at all times and buttons in left hand corner should not be visible or active.
* Rainy Day Test:
  + Test Case  - MultiTouch, Leap Motion, and EyeX Connected User tries to Draw with leap Motion
    - Test Purpose: To determine if the user can draw with leap motion when default mode is set for specified devices.
    - Test Procedure: User plugs in the multitouch screen the leap motion and eyeX and starts the application. User then tries to draw with the leap motion.
    - Expected Results: User should not be able to draw with leap motion since the default mode specifies this when all three devices are connected.
  + Test Case  - MultiTouch and Leap Motion User tries Draw with leap Motion
    - Test Purpose: To determine if the user can draw with leap motion when default mode is set for specified devices.
    - Test Procedure: User plugs in the multitouch screen the leap motion and eyeX and starts the application. User then tries to draw with the leap motion.
    - Expected Results: User should not be able to draw with leap motion since the default mode specifies this when only leap motion and multitouch devices are connected.
* Sunny Day Test:
  + Test Case  - Leap Motion Connection
    - Test Purpose: To determine if user is provided feedback about leap motion connection status.
    - Test Procedure: User plugs in leap motion device and starts program.

Then he or she checks Ui display.

* + - Expected Results: Green box should appear in Ui display.
  + Test Case  - Multi Touch Connection
    - Test Purpose: To determine if user is provided feedback about multi touch connection status.
    - Test Procedure: User plugs in Multitouch device and starts program.

Then he or she checks Ui display.

* + - Expected Results: Blue box should appear in Ui display.
  + Test Case  - EyeX Connection
    - Test Purpose: To determine if user is provided feedback about eyeX connection status.
    - Test Procedure: User plugs in eyeX device and starts program.

Then he or she checks Ui display.

* + - Expected Results: Red box should appear in Ui display.
* Rainy Day Test:
  + Test Case  - User unplugs Leap Motion or device fails
    - Test Purpose: Test if correct feedback provided if user unplugs leap motion or device fails.
    - Test Procedure: User unplugs leap motion while application running.
    - Expected Results: Green box should disappear. If green box remains in Ui display test fails.
  + Test Case  - User unplugs Multi Touch or device fails
    - Test Purpose: Test if correct feedback provided if user unplugs multi Touch or device fails.
    - Test Procedure: User unplugs multi Touch while application running.
    - Expected Results: Blue box should disappear. If blue box remains in Ui display test fails.
  + Test Case  - User unplugs eyeX or device fails
    - Test Purpose: Test if correct feedback provided if user unplugs eyeX or device fails.
    - Test Procedure: User unplugs eyeX while application running.
    - Expected Results: Red box should disappear. If red box remains in Ui display test fails.
* Sunny Day Test:
  + Test Case  - User wants to undo multiple drawn lines or shapes
    - Test Purpose: To determine if user can undo a previously drawn line or shape.
    - Test Procedure: User starts program and draws three lines or three shapes starting from the left side of canvas. He or she then presses the undo button once. Observes results then presses undo button a second time. Observes results and then presses undo button a final time and observes results.
    - Expected Results: After undo button pressed for the first time the line or shape farthest to the right should disappear. After second time undo button pressed line or shape farthest to the right should disappear. After the button is pressed a third times all lines and shapes should have been removed.
* Rainy Day Test:
  + Test Case  - User presses undo button when no lines or shapes drawn
    - Test Purpose: Test undo button works correctly if no lines or shapes drawn and button pressed.
    - Test Procedure: User starts program and immediately without drawing anything presses undo button.
    - Expected Results: Nothing should happen since no lines or shapes drawn. If anything changes in the program test fails.
  + Test Case  - User trys to undo more line or shapes than undo limit
    - Test Purpose: Test undo button works correctly if 11 lines or shapes drawn and button pressed 11 times.
    - Test Procedure: User starts program and draws 11 lines or 11 shapes starting from the left side of canvas. He or she then presses the undo button 11 times in a row and observes result.
    - Expected Results: One line or shape should remain since the undo limit is set to 10 items. If more or less than one line or shape occurs than test fails.
* Sunny Day Test:
  + Test Case  - Kiss Gesture Read by Real Sense
    - Test Purpose: To determine if Real Sense correctly reads kiss gesture.
    - Test Procedure: User starts program and draws three lines or three shapes starting from the left side of canvas. He or she then performs a kiss gesture and holds gesture for at least one second. Observes results then performs another kiss gesture for one second. Observes results and then performs a kiss gesture a final time and observes results.
    - Expected Results: After kiss gesture performed for the first time the line or shape farthest to the right should disappear. After second time kiss gesture performed line or shape farthest to the right should disappear. After kiss gesture performed for a third time all lines and shapes should have been removed.
  + Test Case  - Double Eyebrow Raise Gesture Read by Real Sense
    - Test Purpose: To determine if Real Sense correctly reads double eyebrow raise gesture.
    - Test Procedure: User starts program and draws three lines or three shapes starting from the left side of canvas. He or she then performs a double eyebrow raise gesture and holds gesture for at least one second. Observes results then performs another double eyebrow raise gesture for one second. Observes results and then performs a double eyebrow raise gesture a final time and observes results.
    - Expected Results: After double eyebrow raise gesture performed for the first time the line or shape farthest to the right should disappear. After second time double eyebrow raise gesture performed line or shape farthest to the right should disappear. After double eyebrow raise gesture performed for a third time all lines and shapes should have been removed.
* Rainy Day Test:
  + Test Case  - User Accidently performs kiss gesture
    - Test Purpose: Test if accidently kiss gesture is read by Real Sense.
    - Test Procedure: User starts program and begins to draw lines. He or she then performs a quick kiss gesture that lasts less than one second.
    - Expected Results: Nothing should happen and canvas should not be altered. If item drawn disappears then kiss gesture was read and test fails.
  + Test Case  - User Accidently performs double eyebrow raise gesture
    - Test Purpose: Test if accidently double eyebrow raise gesture is read by Real Sense.
    - Test Procedure: User starts program and begins to draw lines. He or she then performs a quick double eyebrow raise gesture that lasts less than one second.
    - Expected Results: Nothing should happen and canvas should not be altered. If item drawn disappears then double eyebrow raise gesture was read and test fails.
* Sunny Day Test:
  + Test Case  - Activate Proximity Menu
    - Test Purpose: To determine if user can turn on proximity menu.
    - Test Procedure: User starts program and then while in hover zone performs a counterclockwise circle gesture using leap motion.
    - Expected Results: Four quadrants appear on canvas.
  + Test Case  - Test if Upper Left-hand Quandrant is Activated
    - Test Purpose: To determine if user can select the Upper Left-hand quadrant in order to change current drawing color.
    - Test Procedure: After user has activated Proximity menu he or she moves finger while in hover zone into Upper Left-hand corner and performs a clockwise circle gesture using leap motion device.
    - Expected Results: Feedback should be given to user that color setting has changed.
  + Test Case  - Test if Upper Right-hand Quandrant is Activated
    - Test Purpose: To determine if user can select the Upper Right-hand quadrant in order to save the current canvas.
    - Test Procedure: After user has activated Proximity menu he or she moves finger while in hover zone into Upper Right-hand corner and performs a clockwise circle gesture using leap motion device.
    - Expected Results: Feedback should be given to user that canvas was saved.
  + Test Case  - Test if Lower Left-hand Quandrant is Activated
    - Test Purpose: To determine if user can select the Lower Left-hand quadrant in order to change current shape setting.
    - Test Procedure: After user has activated Proximity menu he or she moves finger while in hover zone into Lower-Left-hand corner and performs a clockwise circle gesture using leap motion device.
    - Expected Results: Feedback should be given to user that the current shape setting has changed.
  + Test Case  - Test if Lower Right-hand Quandrant is Activated
    - Test Purpose: To determine if user can select the Lower Right-hand quadrant in order to turn off proximity menu.
    - Test Procedure: After user has activated Proximity menu he or she moves finger while in hover zone into Lower-Right-hand corner and performs a clockwise circle gesture using leap motion device.
    - Expected Results: User should see the proximity menu disappear.
* Rainy Day Test:
  + Test Case  - User trys to turn on proximity menu using clockwise circular gesture
    - Test Purpose: Test if proximity menu can be turned on by accidently gesture.
    - Test Procedure: User starts program and attempts to turn on proximity menu by performing a clockwise circular gesture anywhere on the canvas. This procedure is repeated for all leap motion gestures.
    - Expected Results: Proximity menu should not appear. If proximity menu appears test fails.
  + Test Case  - User trys to select two quadrants at stame time
    - Test Purpose: Test if correct quadrant is selected.
    - Test Procedure: After user turns proximity menu on he or she enters hover zone and places finger location on line dividing any two quadrants. He or she then performs a clockwise circular gesture. The procedure is repeated for every combination of the four quadrants.
    - Expected Results: No feedback of any kind should be provided. If feedback provided then one of the quadrants was selected and test fails.
* Sunny Day Test:
  + **Test Case  - Circle gesture not detected while drawing**
    - Test Purpose: To determine if the circle gesture is disabled while a user is drawing using the leap motion device.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then draws circles in different parts of the canvas.
    - Expected Results: The User should expect to see circles drawn in correct locations.  User should also see no change in the current drawing settings throughout test.
  + **Test Case  - Swipe gesture not detected while drawing**
    - Test Purpose: To determine if the swipe gesture is disabled while a user is drawing using the leap motion device.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then draws vertical and horizontal straight lines in different parts of the canvas.
    - Expected Results: The User should expect to see vertical and horizontal lines drawn to correct locations.  User should also see no change in the current drawing settings throughout test.
  + **Test Case  - Keyboard tap gesture not detected while drawing**
    - Test Purpose: To determine if the keyboard tap gesture is disabled while a user is drawing using the leap motion device.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then performs keyboard tap gestures.
    - Expected Results: The User should observe no change in the applications settings.
  + **Test Case  - Screen tap gesture not detected while drawing**
    - Test Purpose: To determine if the screen tap gesture is disabled while a user is drawing using the leap motion device.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then performs screen tap gestures.
    - Expected Results: The User should observe no change in the applications settings.
* Rainy Day Test:
  + **Test Case  - User begins to draw attempts to perform  circle gesture while still in drawing zone**
    - Test Purpose: To determine user can perform circle gesture while in drawing zone.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then decided to perform a circle gesture while still in drawing zone.
    - Expected Results: The User would expect to see a change to the current setting of the application, but no change in the system would occur.
  + **Test Case  - User begins to draw attempts to perform swipe gesture while still in drawing zone**
    - Test Purpose: To determine user can perform swipe gesture while in drawing zone.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then decided to perform a swipe gesture while still in drawing zone.
    - Expected Results: The User would expect to see a change to the current setting of the application, but no change in the system would occur.
  + **Test Case  - User begins to draw attempts to perform keyboard tap gesture while still in drawing zone**
    - Test Purpose: To determine user can perform keyboard tap gesture while in drawing zone.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then decided to perform a keyboard tap gesture while still in drawing zone.
    - Expected Results: The User would expect to see a change to the current setting of the application, but no change in the system would occur.
  + **Test Case  - User begins to draw attempts to perform screen tap gesture while still in drawing zone**
    - Test Purpose: To determine user can perform screen tap gesture while in drawing zone.
    - Test Procedure: User enters drawing zone with index finger and begins to draw using leap motion. He or she then decided to perform a screen tap gesture while still in drawing zone.
    - Expected Results: The User would expect to see a change to the current setting of the application, but no change in the system would occur.
* Sunny Day Test:
  + Test Case  - Enable Radial Menu
    - Test Purpose: To determine if user can pull up radial menu dynamically.
    - Test Procedure: User double taps location he or she wants radial  menu to appear.
    - Expected Results: Radial menu appears in desired location.
  + Test Case  - Change Color
    - Test Purpose: To change the current color being used to draw.
    - Test Procedure: User presses left button within radial menu.
    - Expected Results: The color updated in the Ui display and feedback provided to user.
  + Test Case  - Change Shape
    - Test Purpose: To change the current shape being used to draw.
    - Test Procedure: User presses right button within radial menu.
    - Expected Results: The shape updated in the Ui display and feedback provided to user.
  + Test Case  - Toggle Symmetry Line
    - Test Purpose: To determine if user can toggle symmetry mode on and off.
    - Test Procedure: User presses top button within radial menu.
    - Expected Results: If symmetry mode is on then it should turn off and if symmetry mode is off it should be turned on.
  + Test Case  - Toggle Ui Display
    - Test Purpose: To determine if user can toggle ui display on and off.
    - Test Procedure: User presses bottom button within radial menu.
    - Expected Results: If Ui display is on then it should turn off and if Ui display is off it should be turned on.
  + Test Case  - Disable Radial Menu
    - Test Purpose: To determine if user can turn radial menu on and off.
    - Test Procedure: User presses middle button within radial menu.
    - Expected Results: Radial menu will be removed from screen.
* Rainy Day Test:
  + Test Case  - User Presses Area between buttons
    - Test Purpose: Test  if radial menu functions correctly if user presses area between buttons within radial menu.
    - Test Procedure: User tries to press left button and misses button
    - Expected Results: No change in current states is expected. If change observed test failed.
  + Test Case  - User Presses Area Outside of Radial menu
    - Test Purpose: Test if radial menu buttons only work within radial menu
    - Test Procedure: User presses any area just outside of radial menu.
    - Expected Results: User should observe that area being drawn. If any state changed then test failed.
* Sunny Day Test:
  + User wants to increase line size by one degree.
    - He or she presses ‘x’ key once so line increases by one degree
  + User wants to increase line size by three degrees
    - He or she presses ‘x’ key the three consecutive times before drawing to get increase.
  + User wants to decrease line size by one degree.
    - He or she presses ‘z’ key once so line increases by one degree
  + User wants to decrease line size by three degrees
    - He or she presses ‘z’ key the three consecutive times before drawing to get increase.
  + User wants to change color to next possible color.
    - He or she presses ‘w’ key once so color changes to next specified color by one position.
  + User wants to change color of line to color that is three positions away.
    - He or she presses ‘w’ key the number of times the desired color is away from current color
  + User wants to change color to last color drawn.
    - He or she presses ‘q’ key once so color changes to next specified color by one position.
  + User wants to change color of line to previous color that is three positions away.
    - He or she presses ‘q’ key the number of times the desired color is away from current color
  + User wants to change random color on.
    - He or she presses ‘r’ key when random color mode is off.
  + User wants to change random color off.
    - He or she presses ‘r’ key when random color mode is on.
* Rainy Day Test:
  + User presses ‘x’ key repetitively to increase line more than system allows .
    - He or she presses ‘x’ 20 times.
  + User presses ‘z’ key repetitively to decrease line more than system allows .
    - He or she presses ‘z’ 20 times.
  + User presses ‘w’ key repetitively to change color line in forward direction more than system allows .
    - He or she presses ‘w’ 20 times.
  + User presses ‘g’ key repetitively to change color line in backward direction more than system allows .
    - He or she presses ‘g’ 20 times.
  + User press ‘r’ key repetitively to change random on and off
    - He or she presses ‘r’ 20 times
  + User presses two different keys at the same time
* Sunny Day Test:
  + User wants to change shape being drawn by both devices.
    - He or she performs circle gesture in clockwise direction.
    - He or she performs circle gesture in counter clockwise direction.
  + User wants to change color being drawn by both devices.
    - He or she performs swipe gesture in a left to right horizontal direction.
    - He or she performs swipe gesture in a right to left horizontal direction.
    - He or she performs swipe gesture in a top to bottom vertical direction.
    - He or she performs swipe gesture in a bottom to top vertical direction.
  + User wants to save current canvas
    - He or she performs keyboard tap gesture.
* Rainy Day Test:
  + User tries to perform gesture before feedback image has faded away
    - Systems only allows one gesture to be recognized at a time
  + User performs incorrect gesture
    - System will not read gesture and no action will take place
  + User performs same gesture twice in a row before drawing
    - System will read both calls and make necessary changes depending on gesture.

# 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

**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 -** Undo's 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

**Double Tap Gesture -** Multitouch gesture which involves tapping in the same location twice.

**Two Finger Tap Gesture -** Multitouch Gesture which involves tapping with two fingers next to eachother.

**Extended Touch Gesture -** Multitouch Gesture involving placing a finger and holding it in place.

**Radial Menu -**Menu, currently activated by the Double Tap Gesture, which contains various functions such as cycle colors, toggle UI, Undo, as well as changing shapes.

**Kiss Gesture -** Real Sense Gesture which detects the facial gesture of kissing

**Smile Gesture -** Real Sense Gesture which detects the facial gesture of smiling

**EyebrownRaise Gesture -** Real Sense Gesture which detects raising of the eyebrows

**Puffy Cheeks Gesture -** Real Sense gesture which detects puffing out your cheeks

**Tongue Out Gesture -** Real Sense Gesture which detects sticking your tongue out.

**Swipe Gesture -** Leap Motion gesture which detects swiping your hand over the device.

**Circle Gesture -** Real Sense Gesture which detects moving your finger in a circular pattern

**Keyboard Tap Gesture -** Real Sense gesture which detects moving your finger down, like

pressing a key on the keyboard.

**Proximity Menu -** Leap motion menu currently activated by the counter clockwise circle gesture. You can then use the clockwise circle gesture in different locations on the screen to perform various functionality.

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

# 

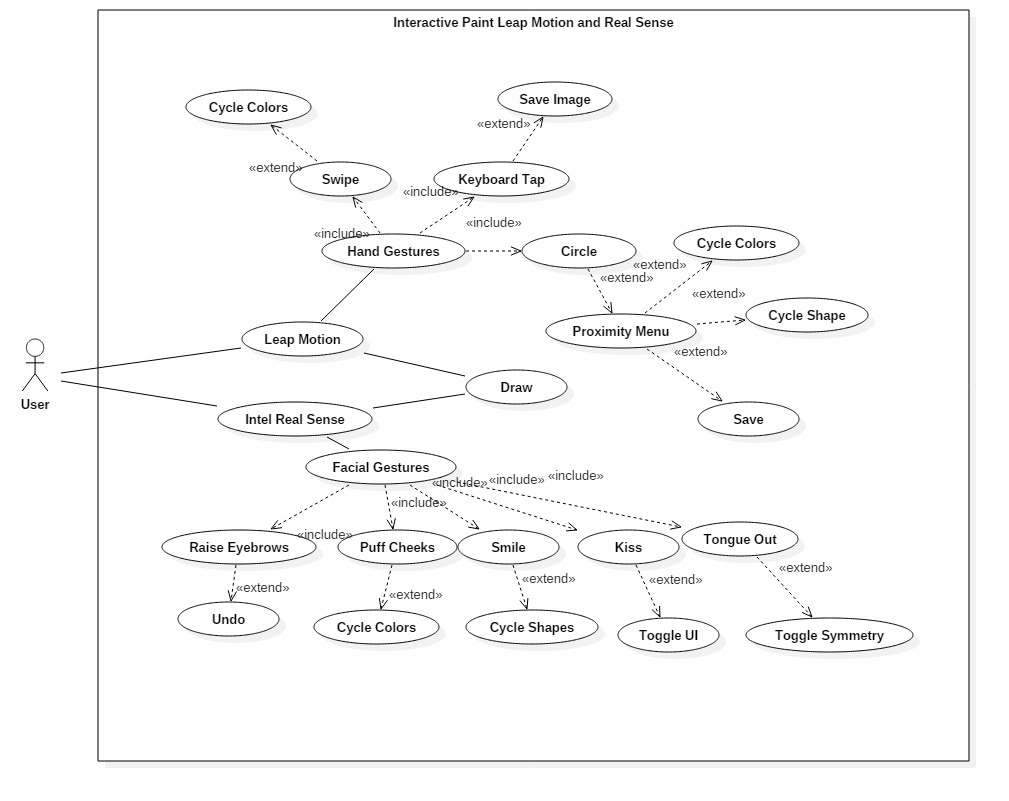
# 

# 

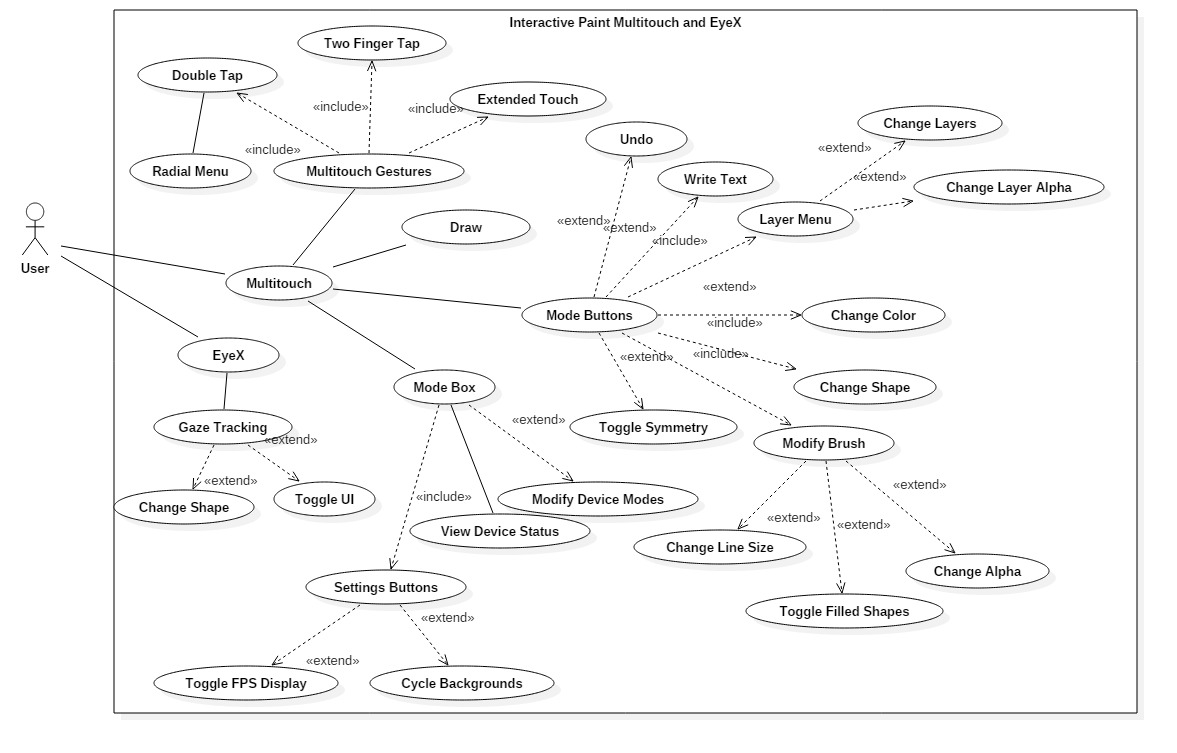
# Appendix

## Appendix A - UML Diagrams

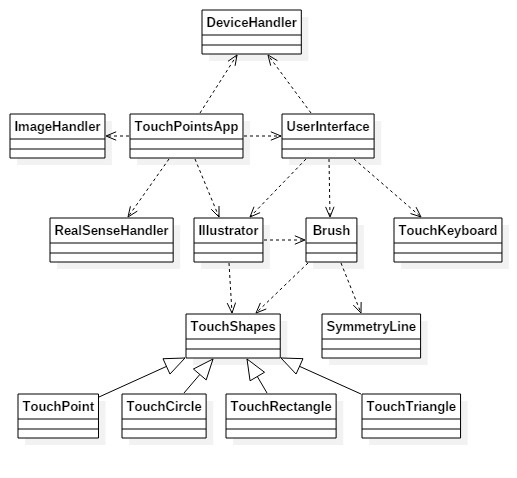
### Static UML Diagrams



Use Case Diagram Leap Motion and Real Sense



Use Case Diagram Multitouch and EyeX



Minimal Class Diagram

### 

### 

### 

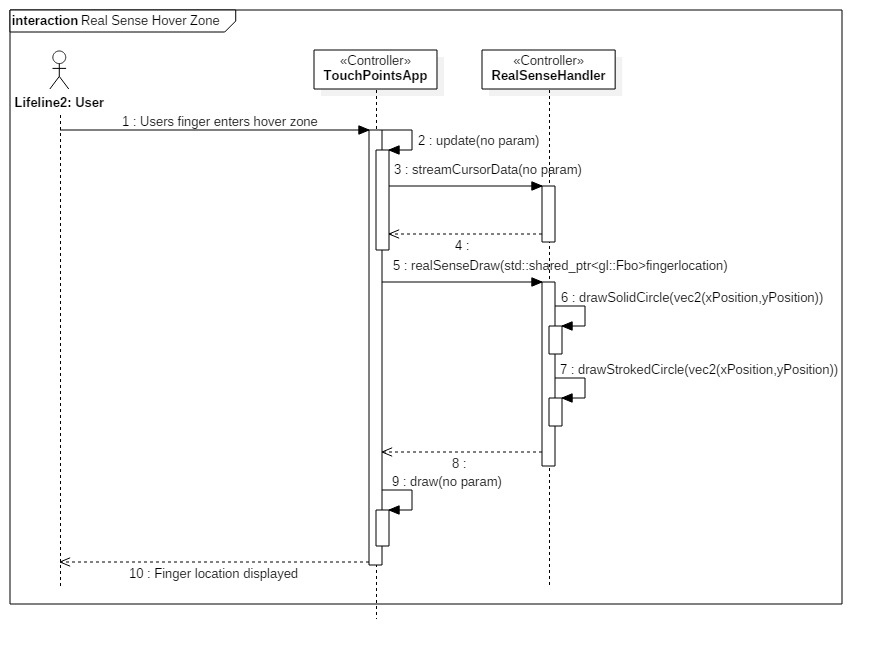
### 

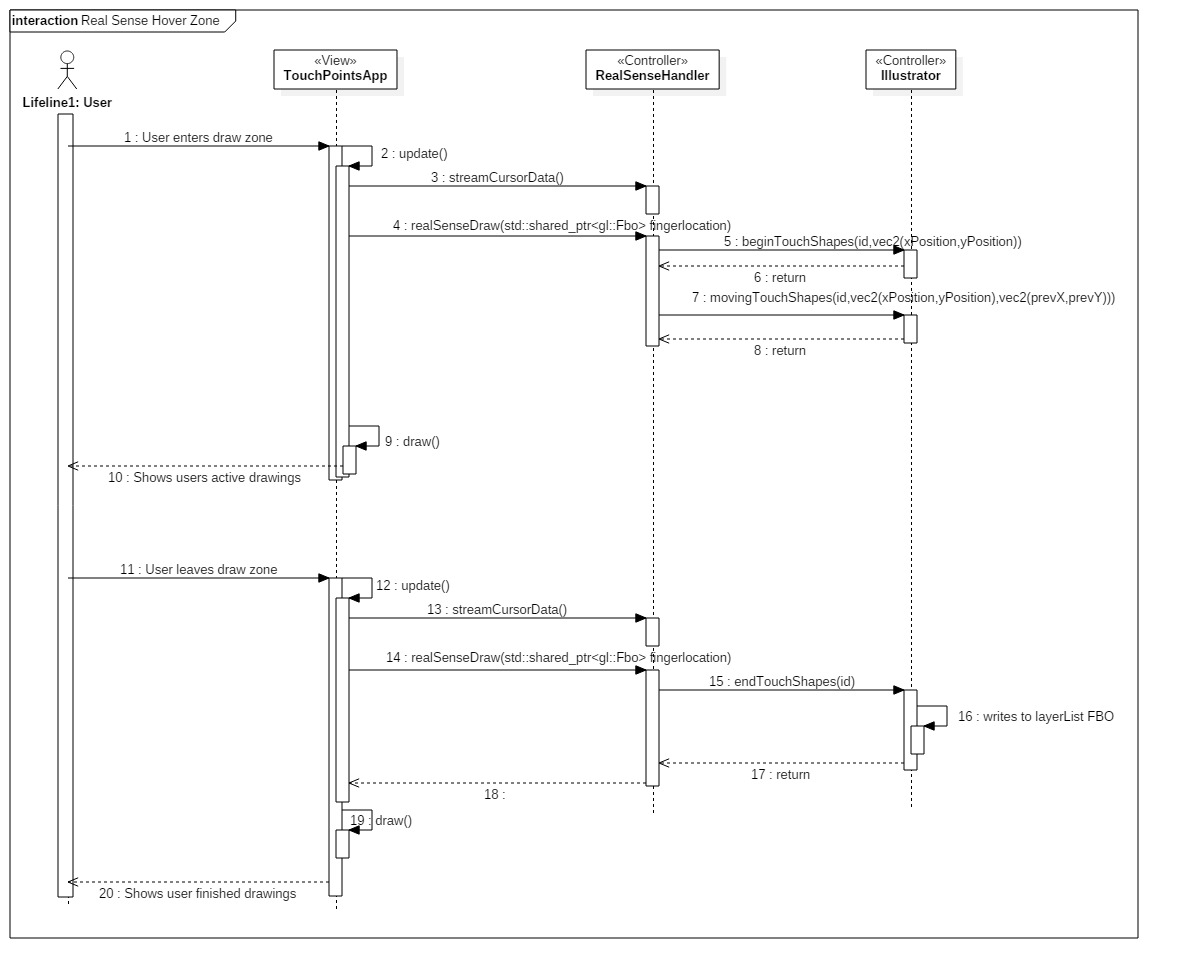
### 

### Dynamic UML Diagrams

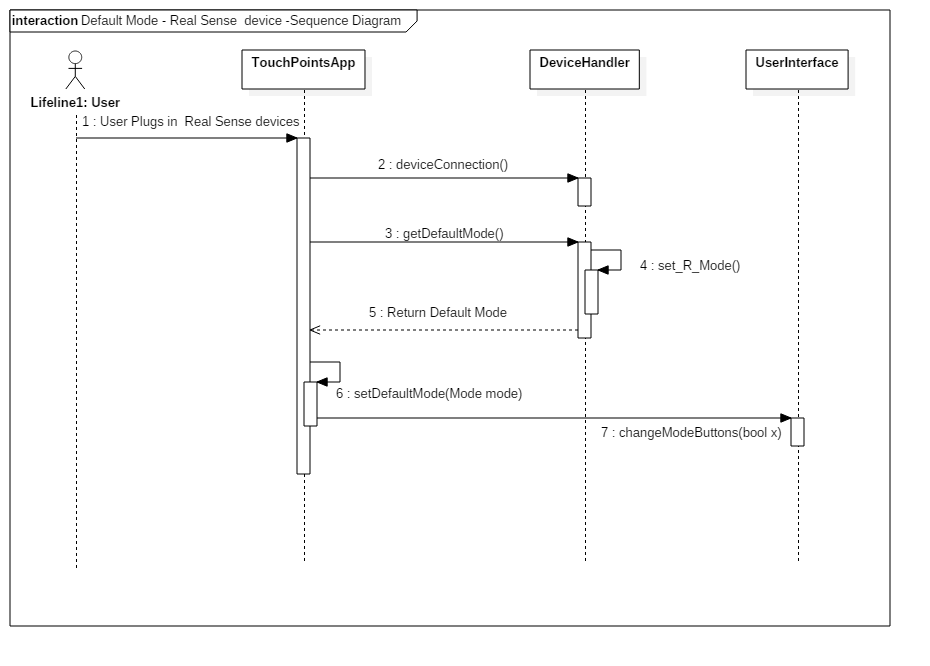
**API Dynamic UML Diagrams**

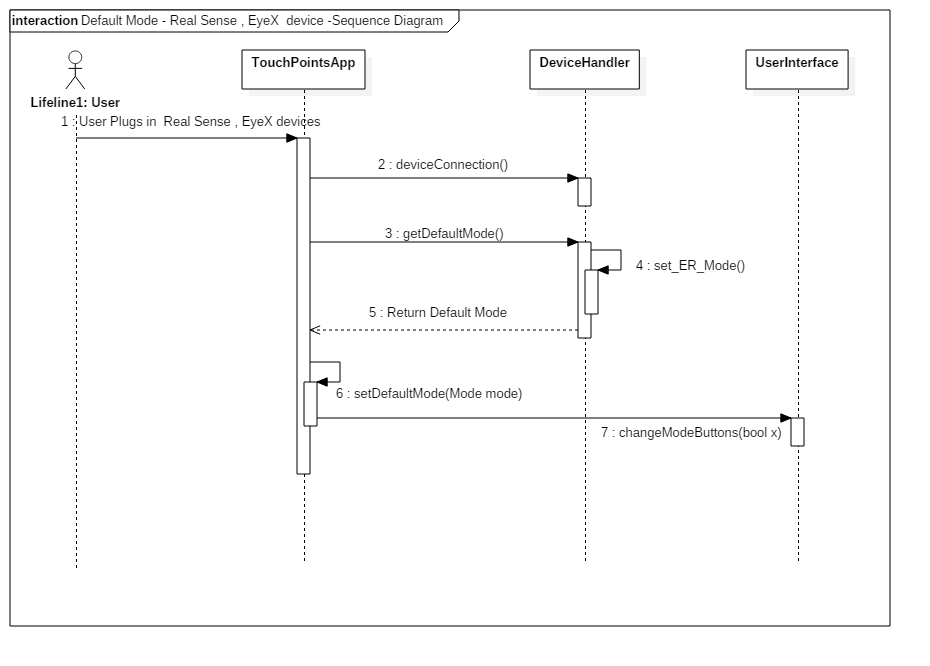
#708

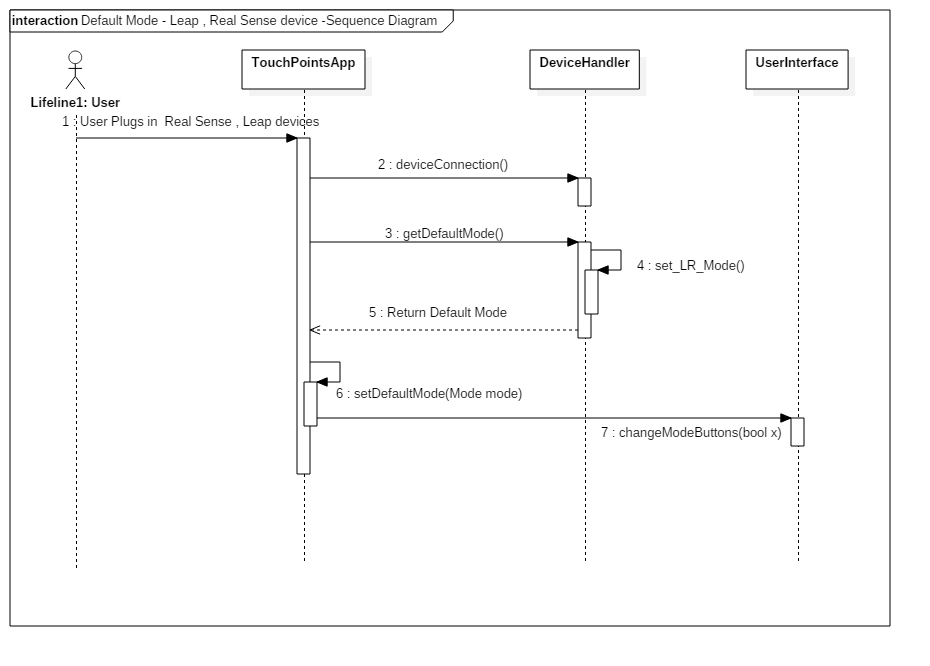
****

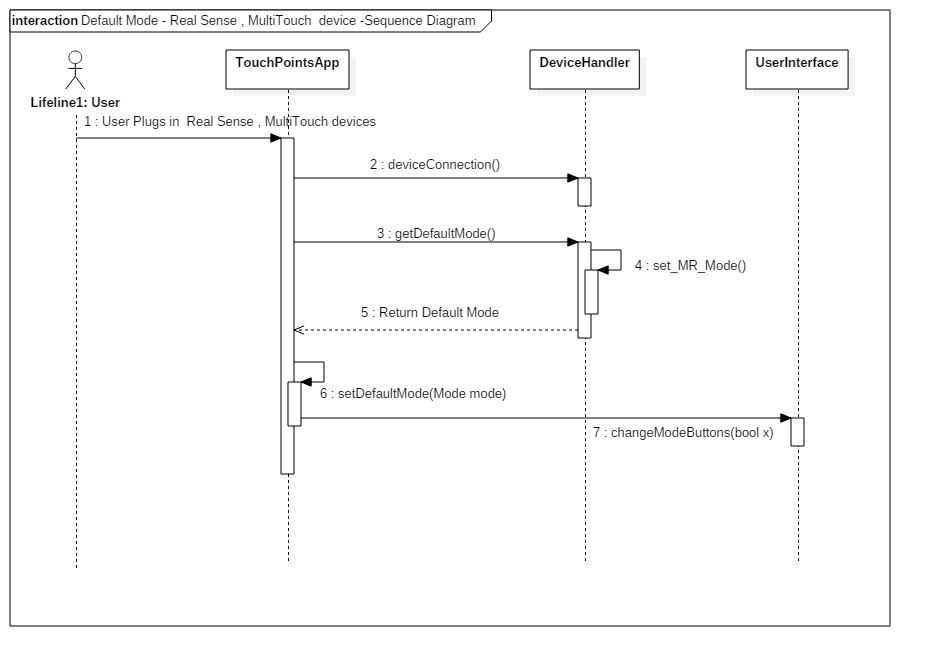
****

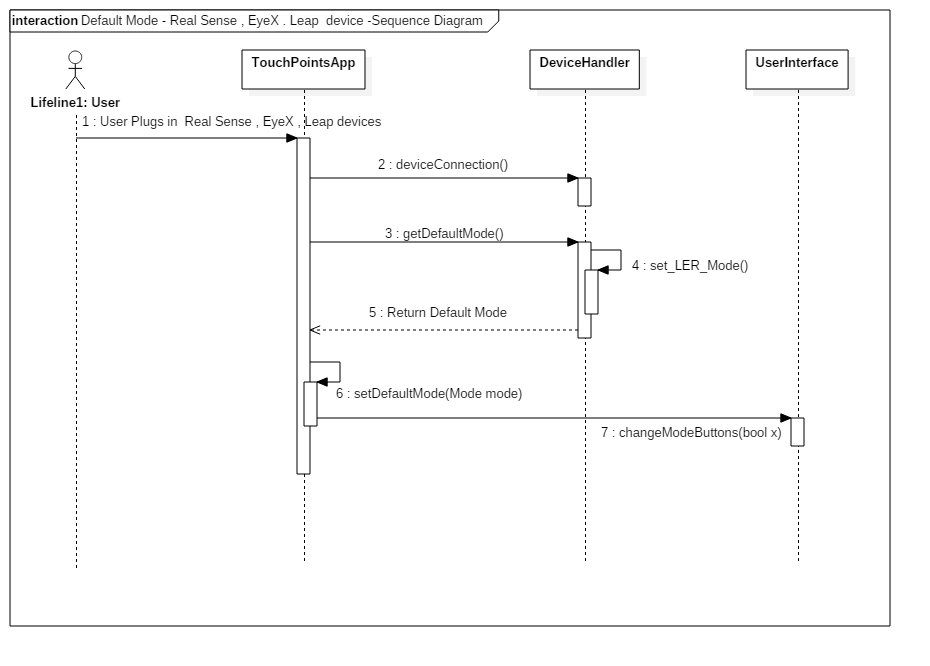
#707

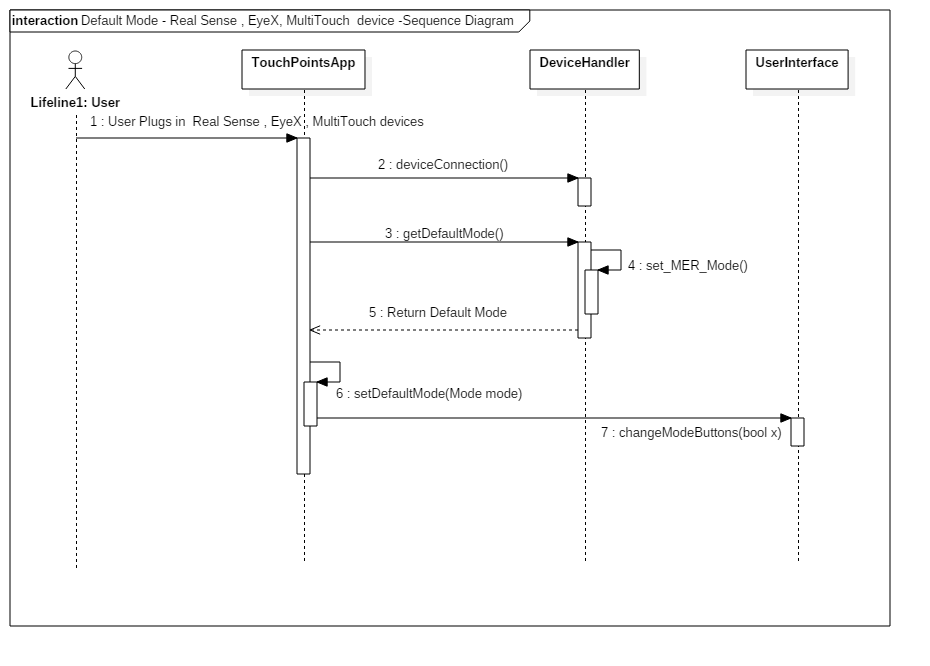


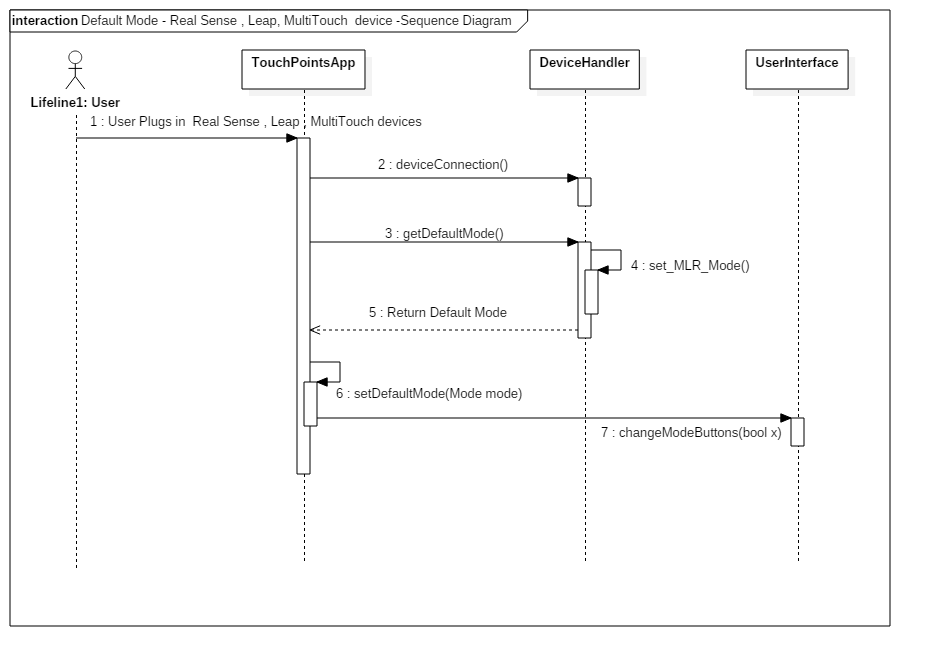


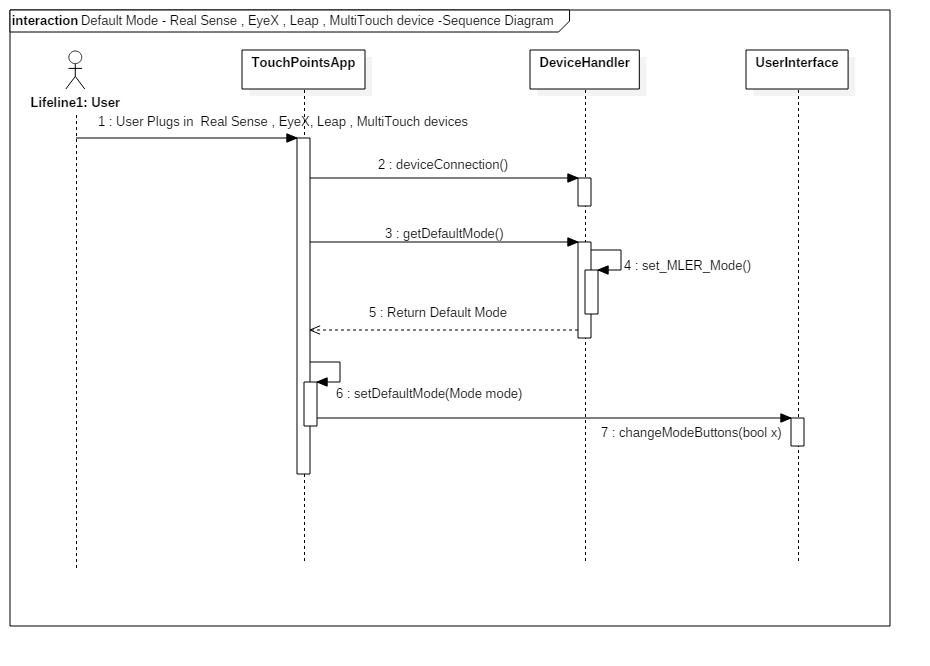




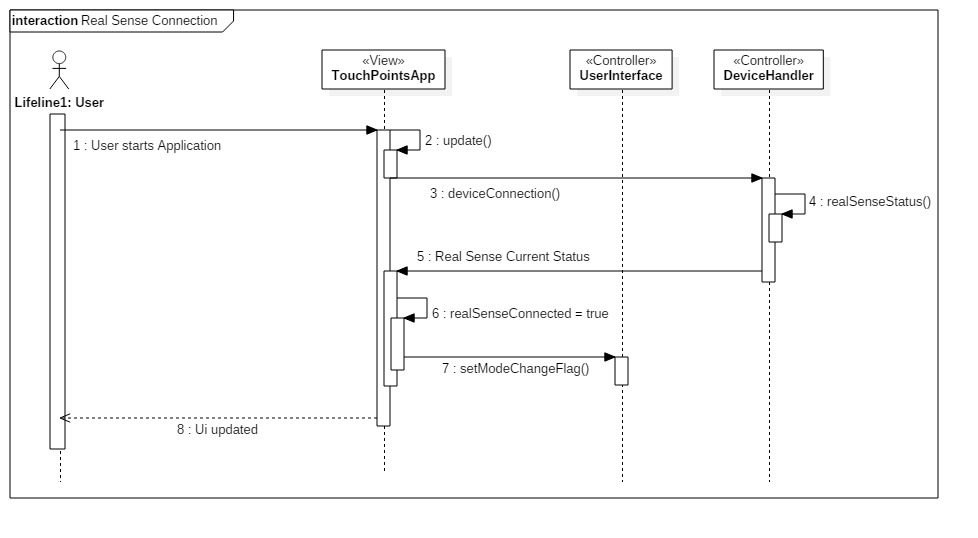




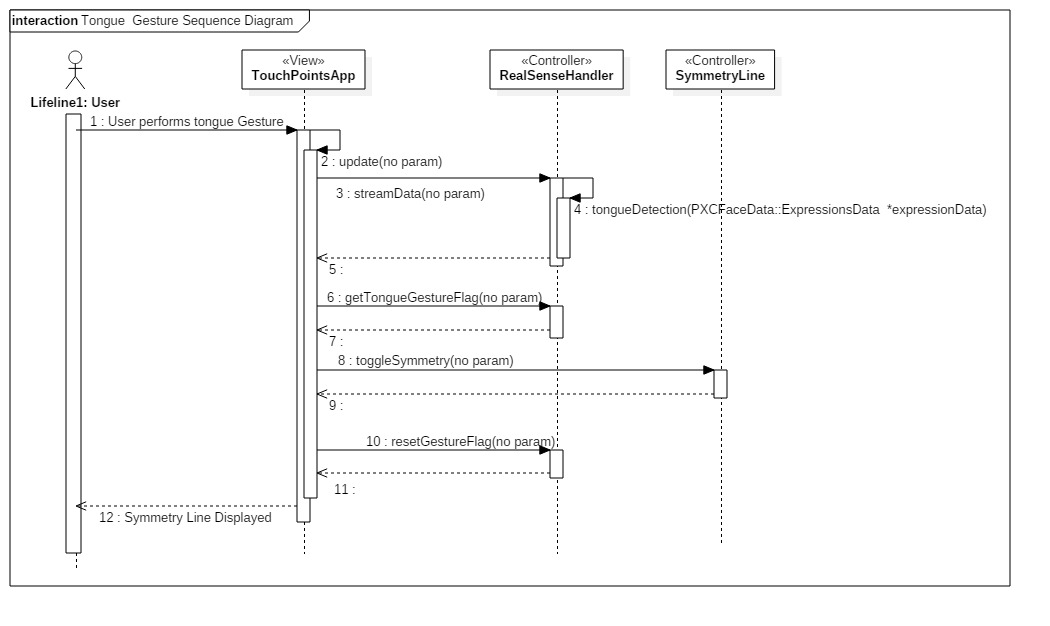


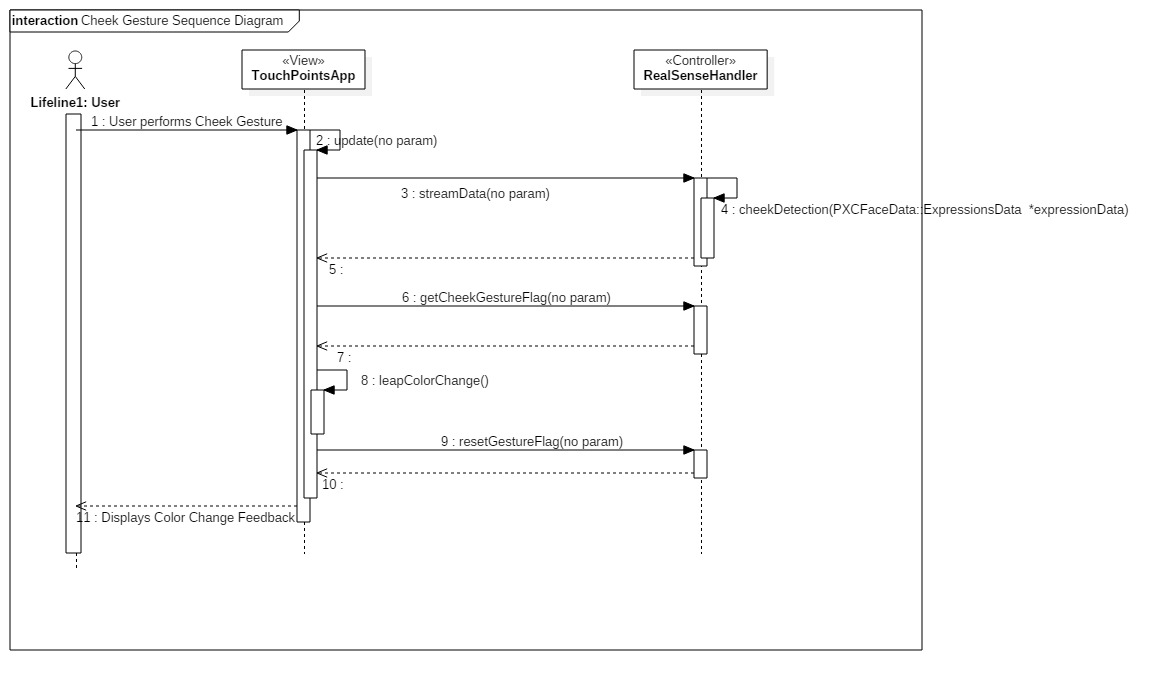


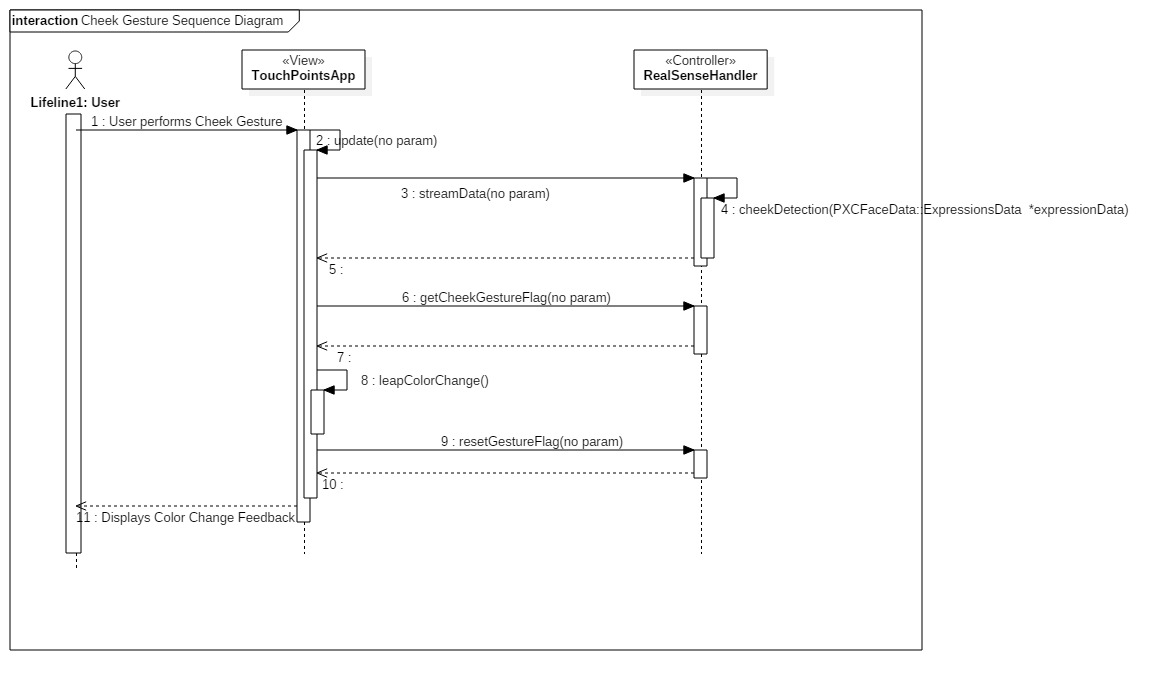
#706



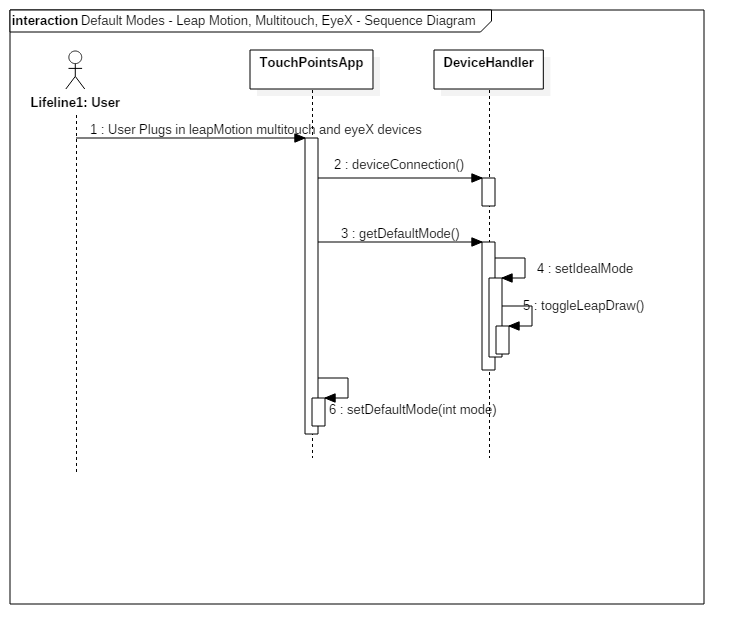
#703

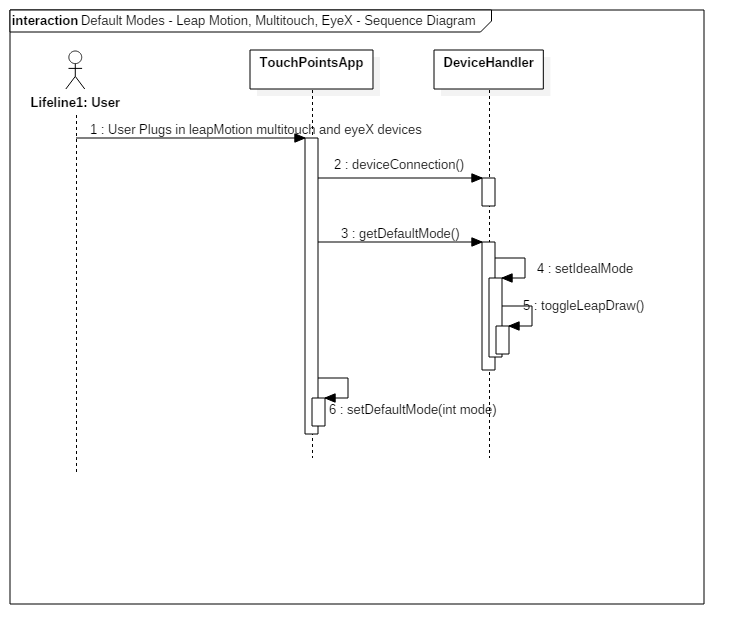


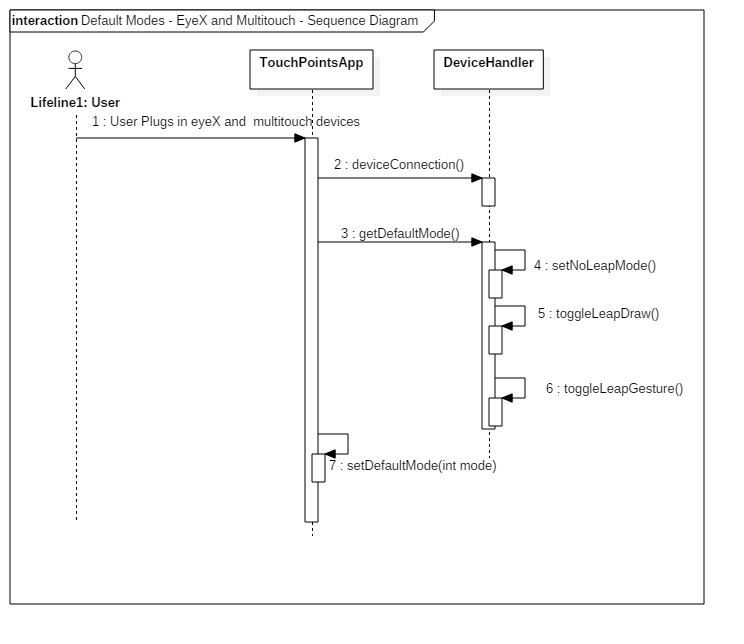


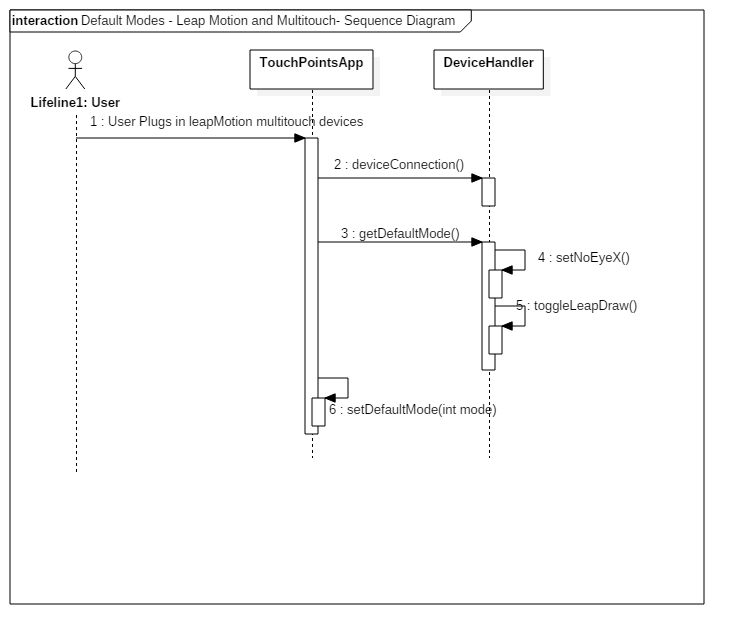


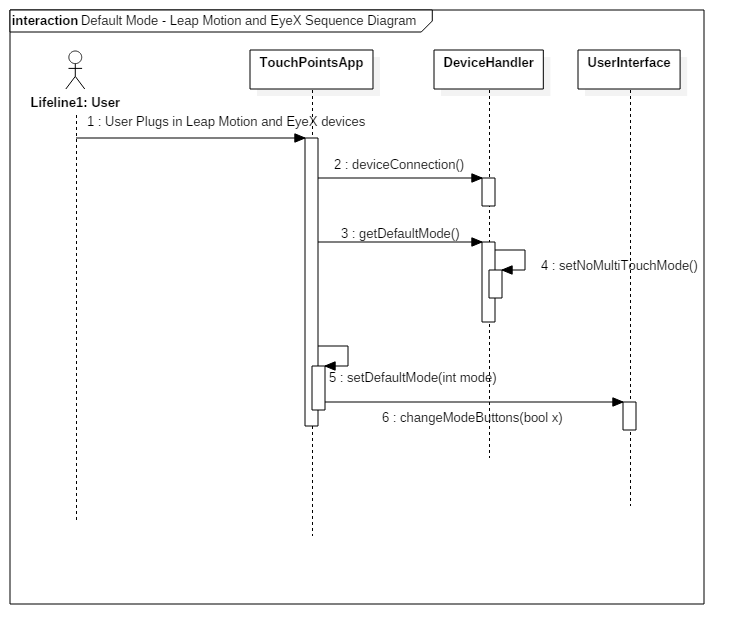
#675

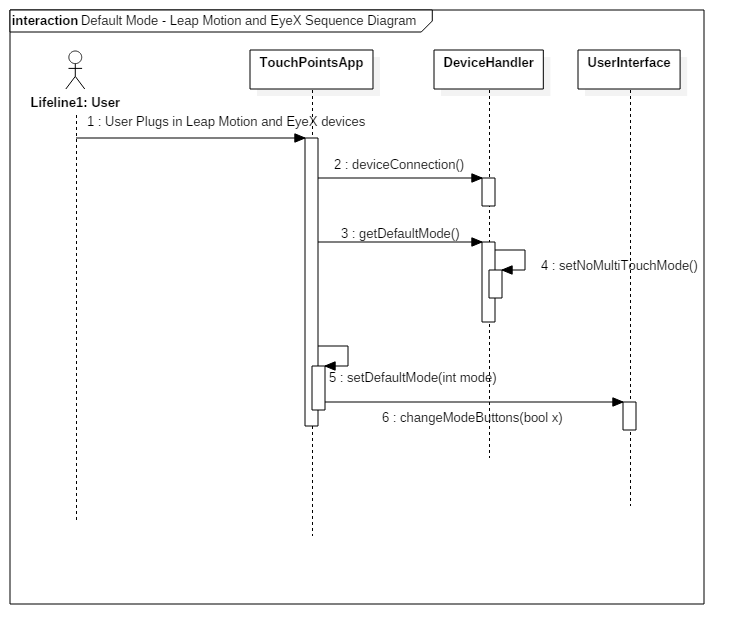
****

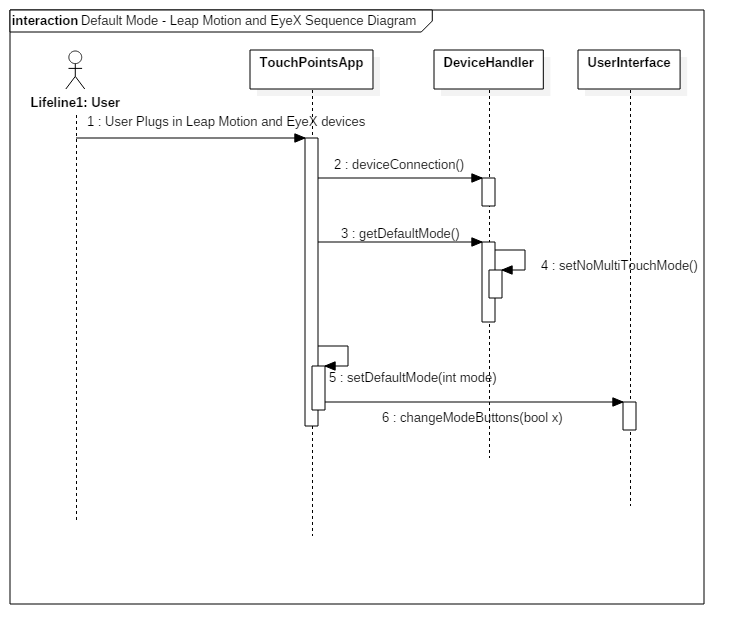
****

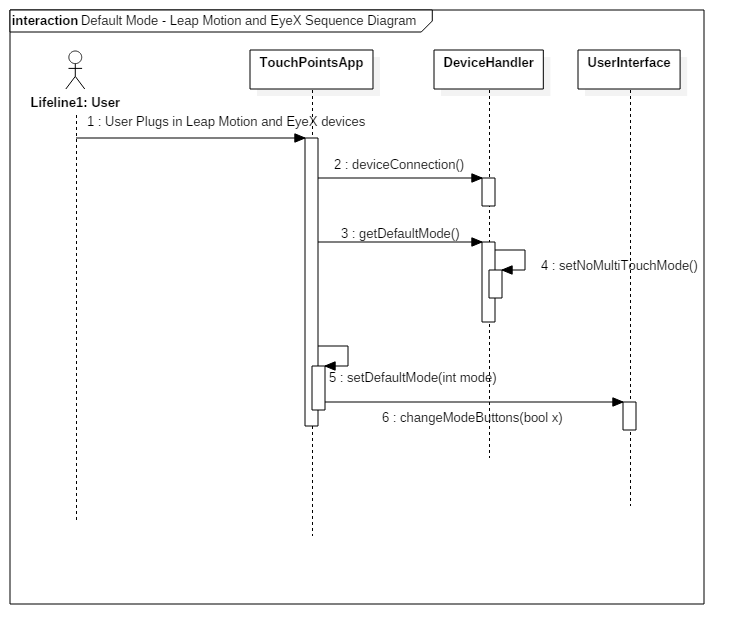
****

****

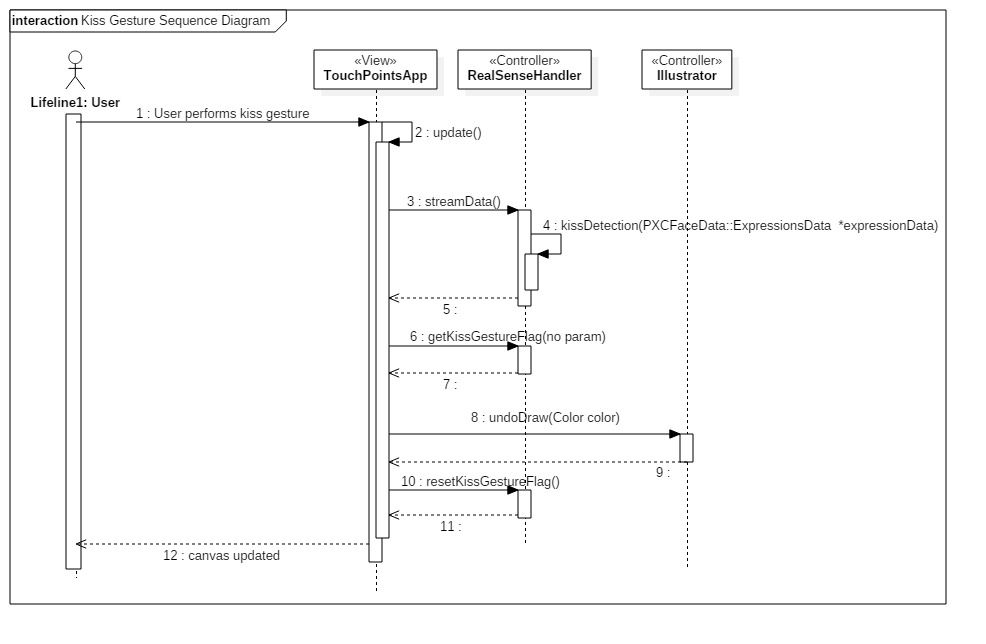
****

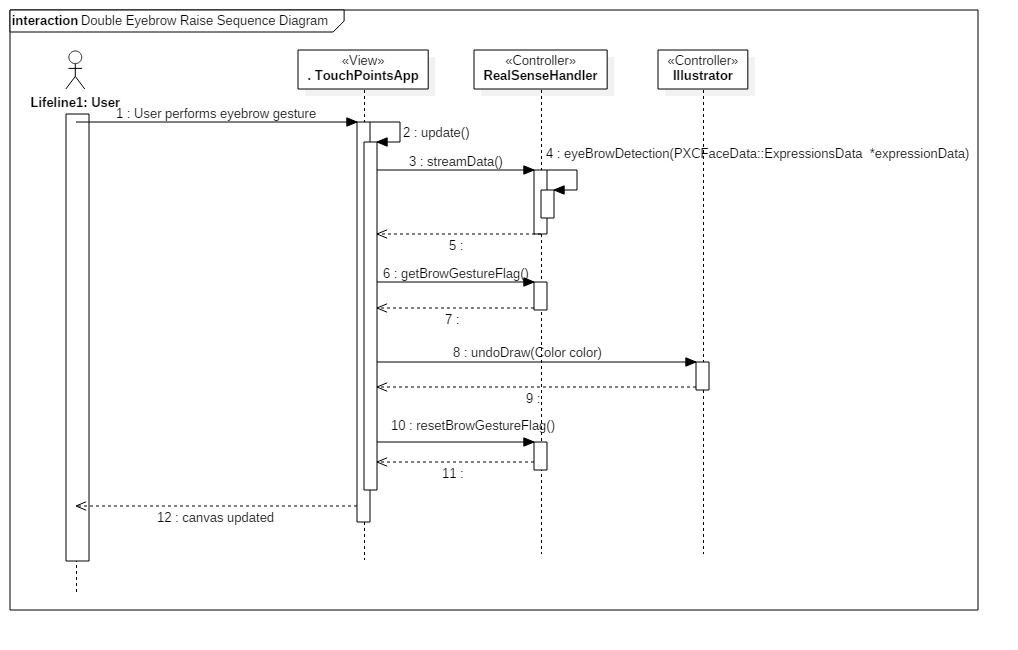
****

****

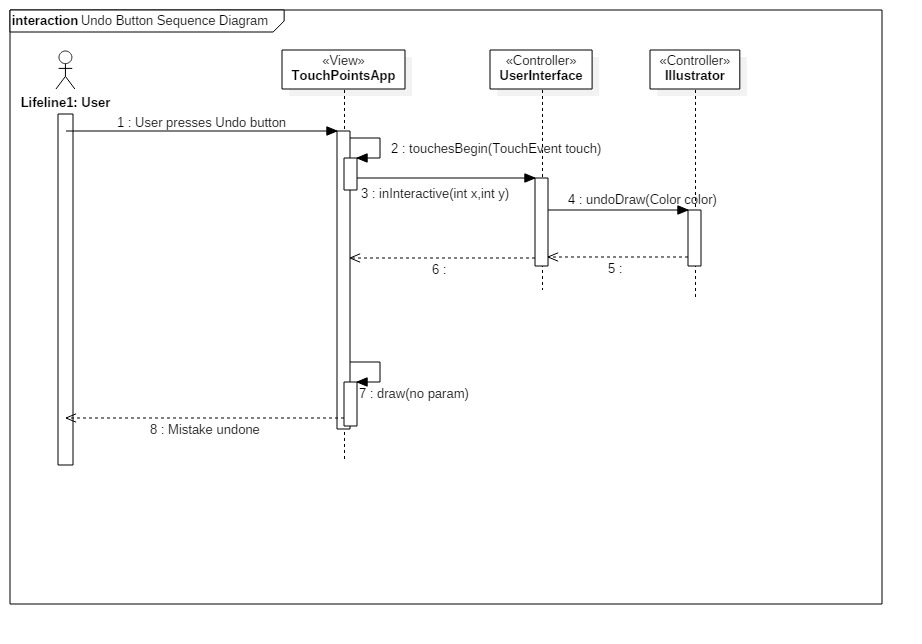
****

#671

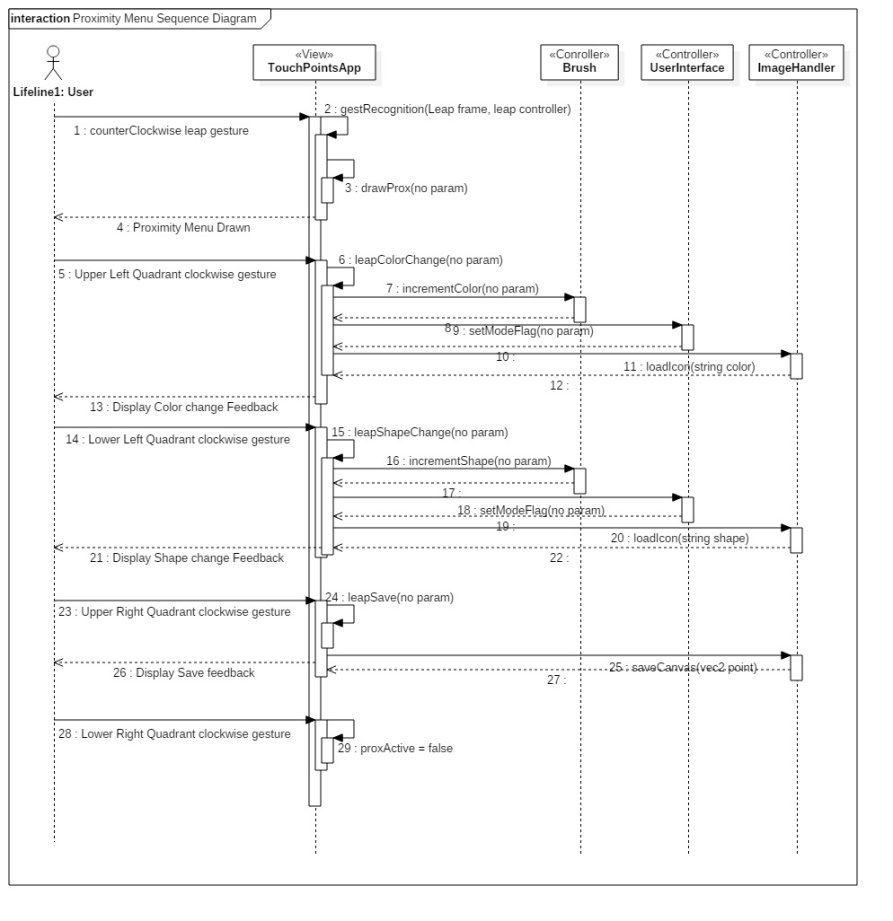
****

****

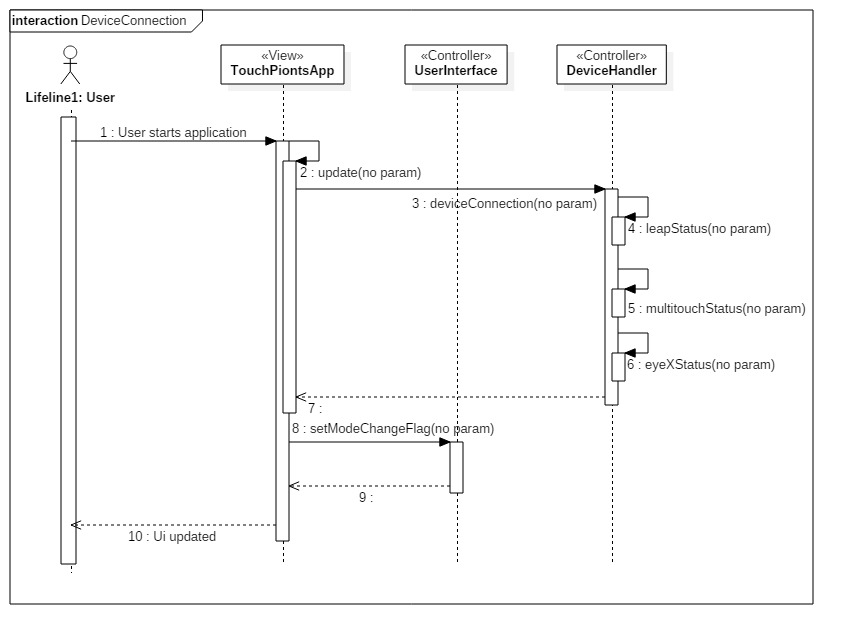
#670



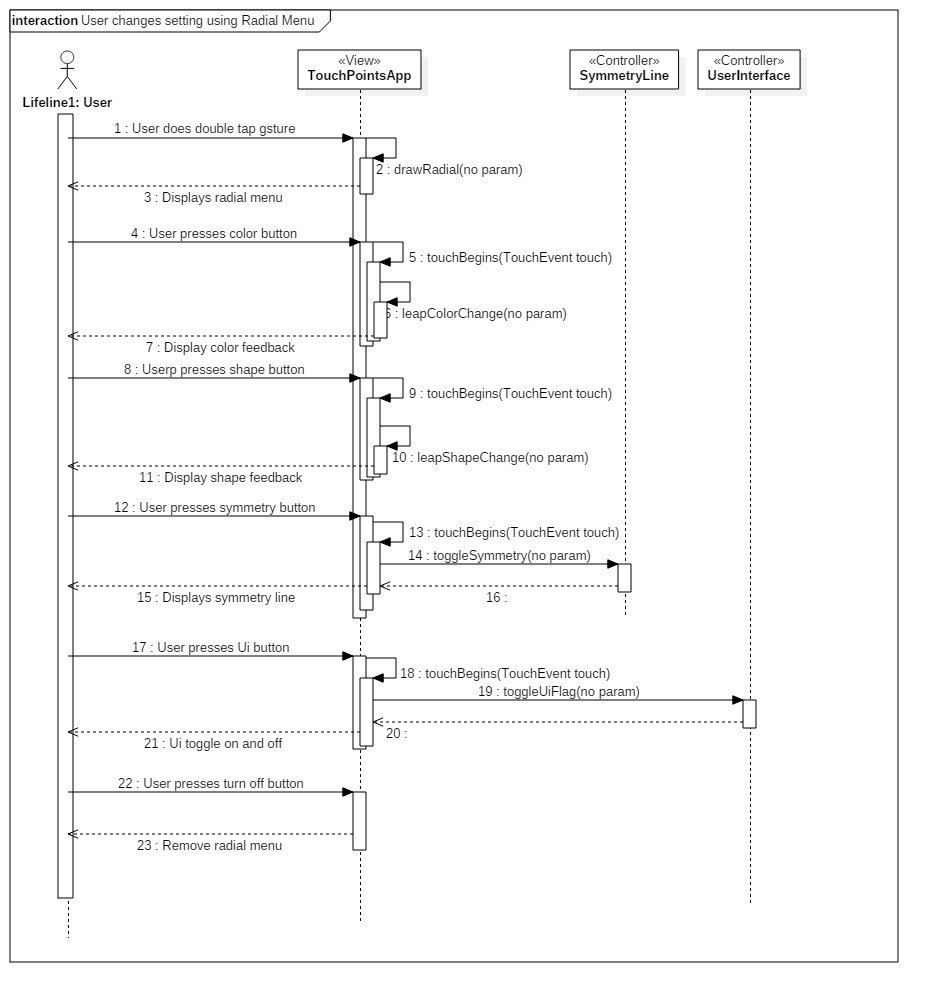
#669



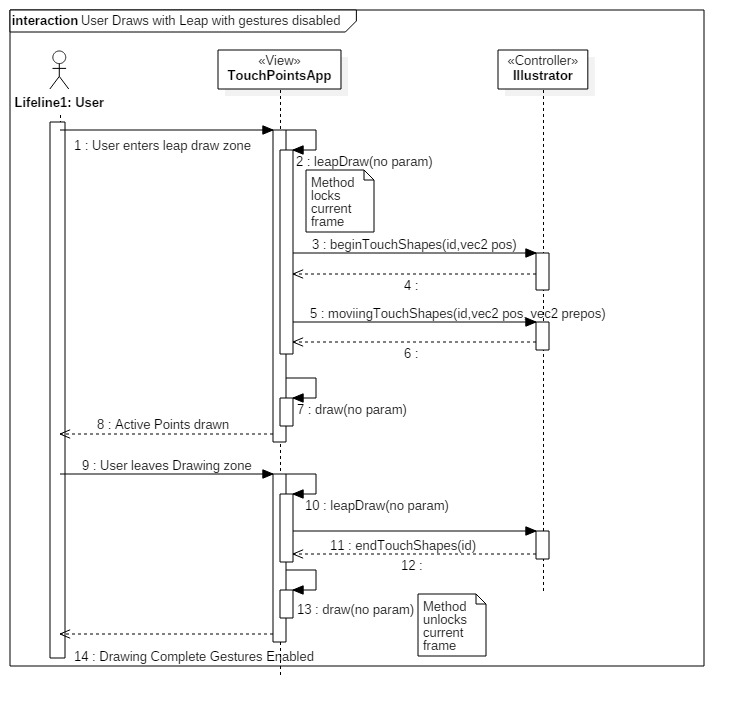
#634

****

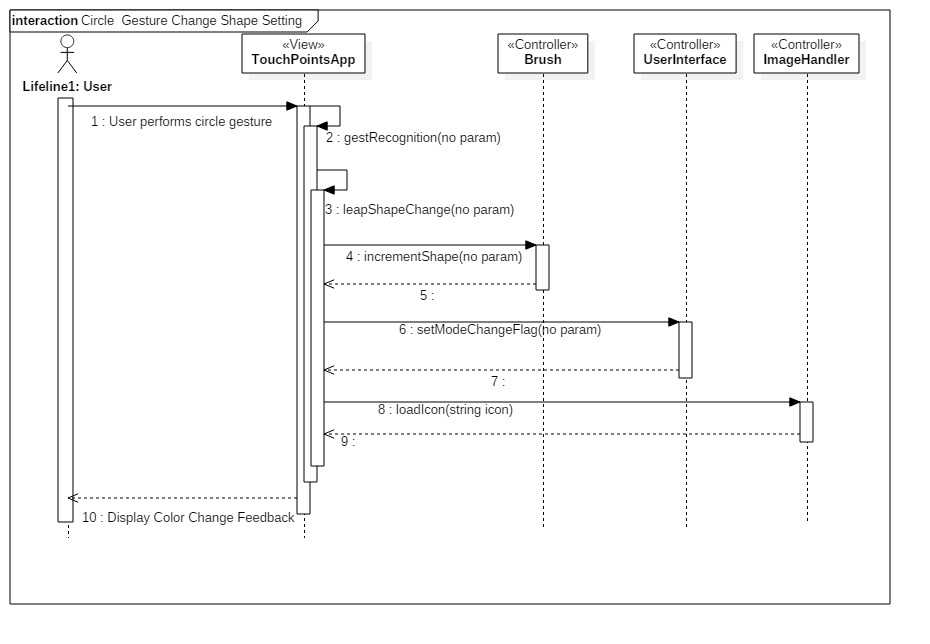
#632

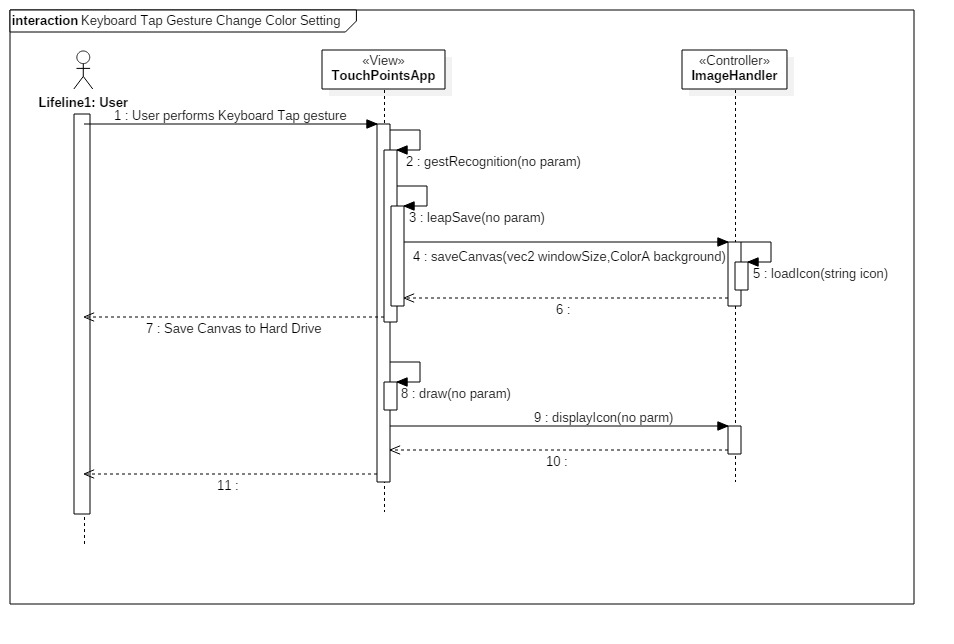
****

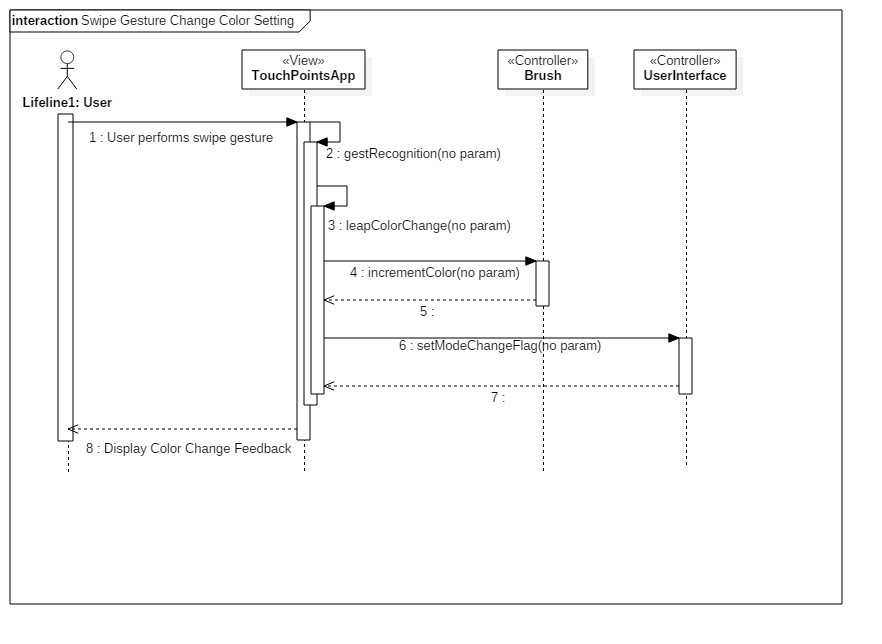
#629

****

#596

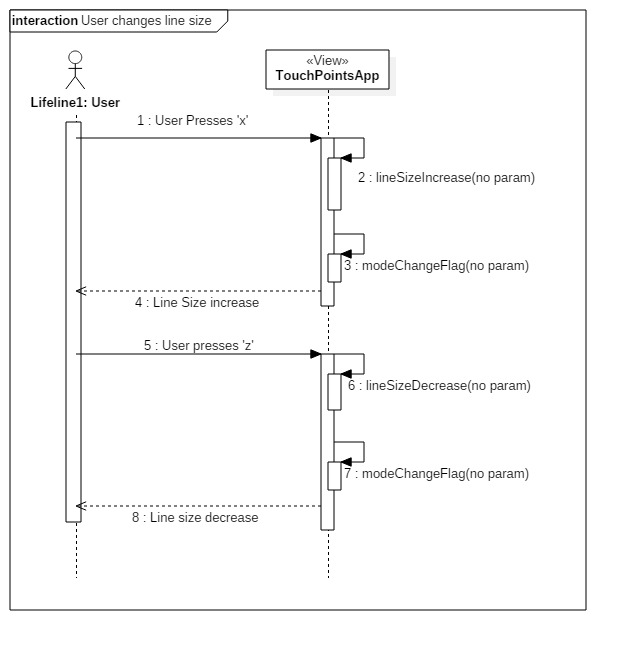




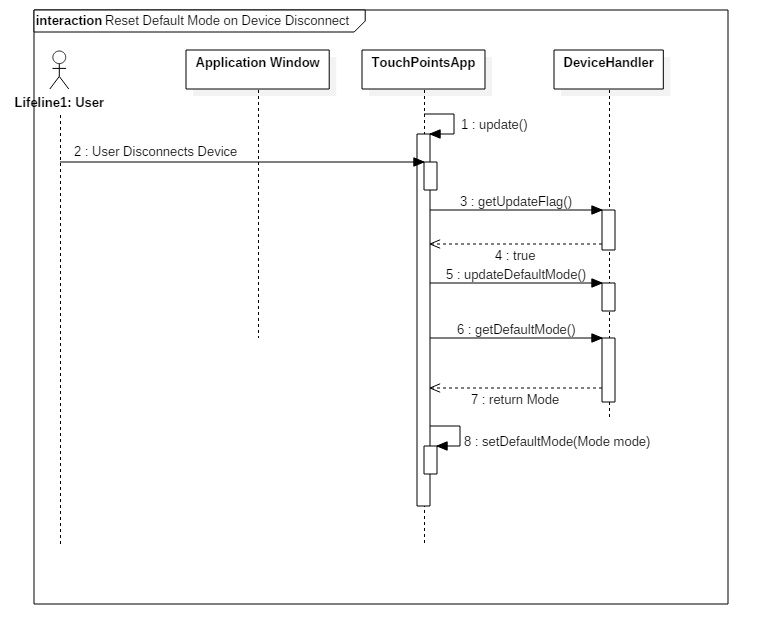


#595

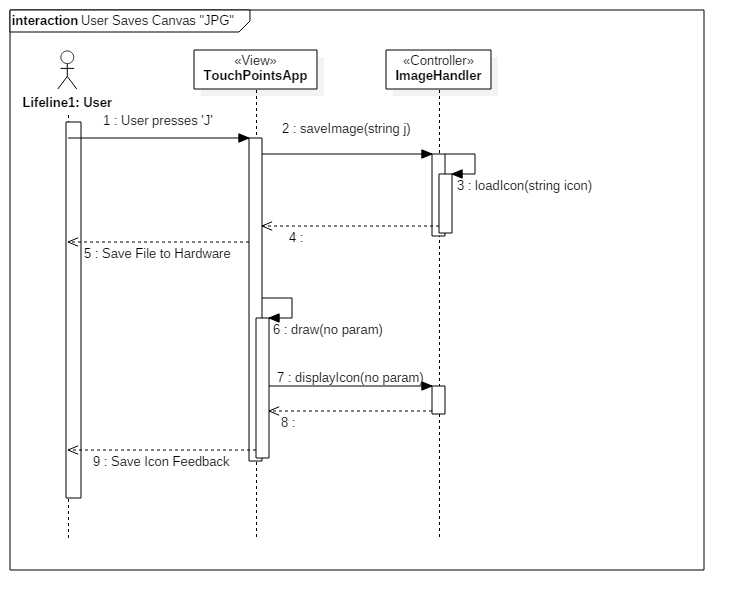


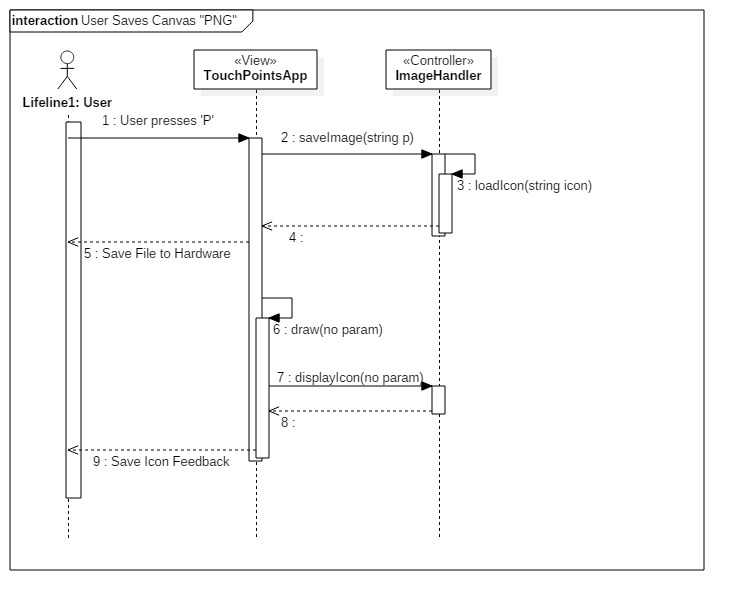


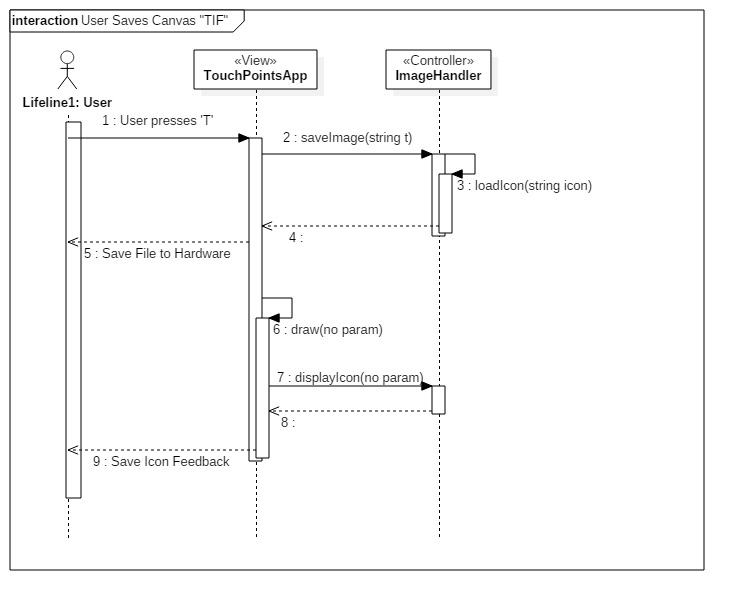
#559



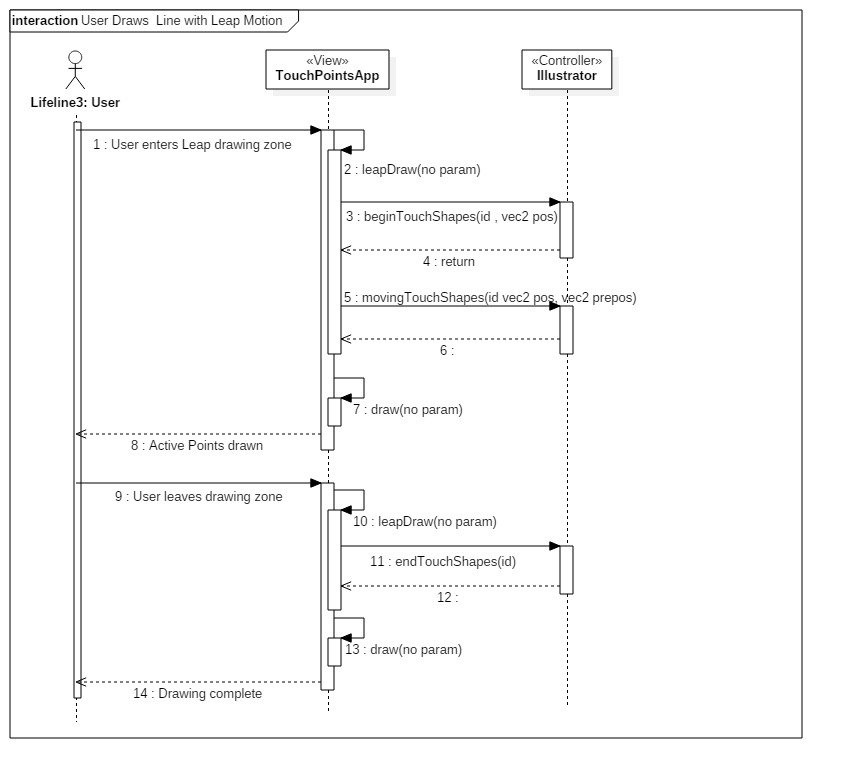
#554

****

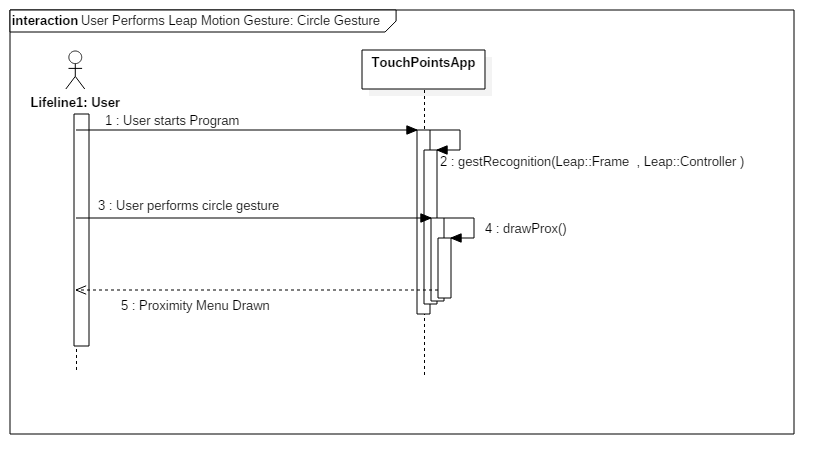
****

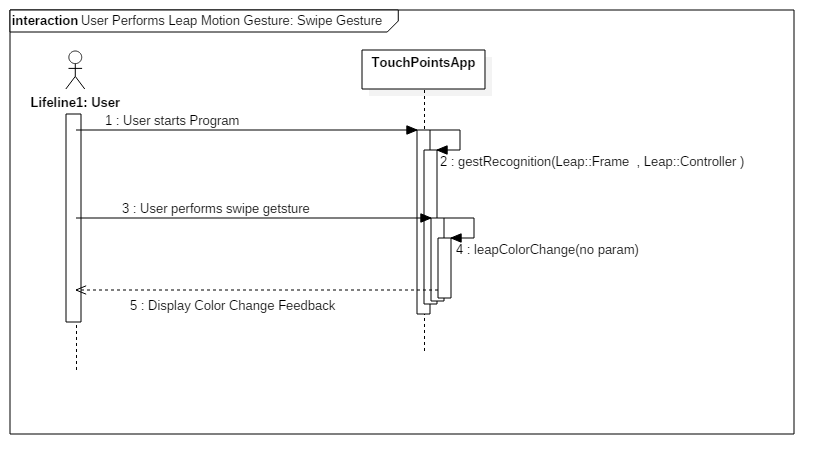
****

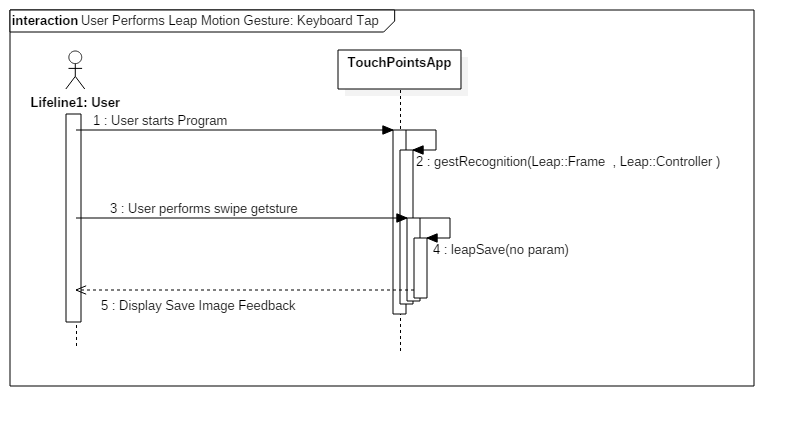
#548

****

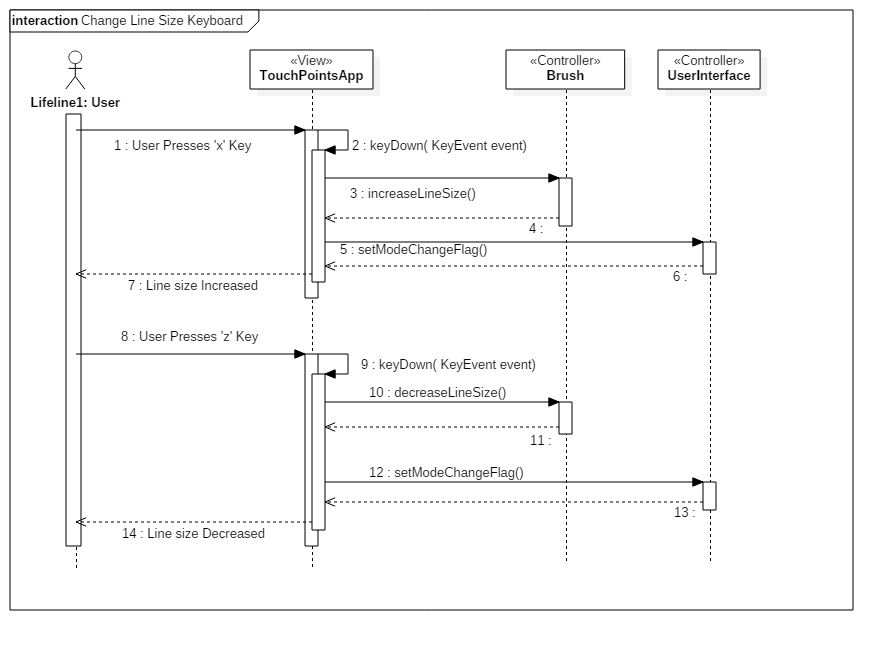
#546

****

****

****

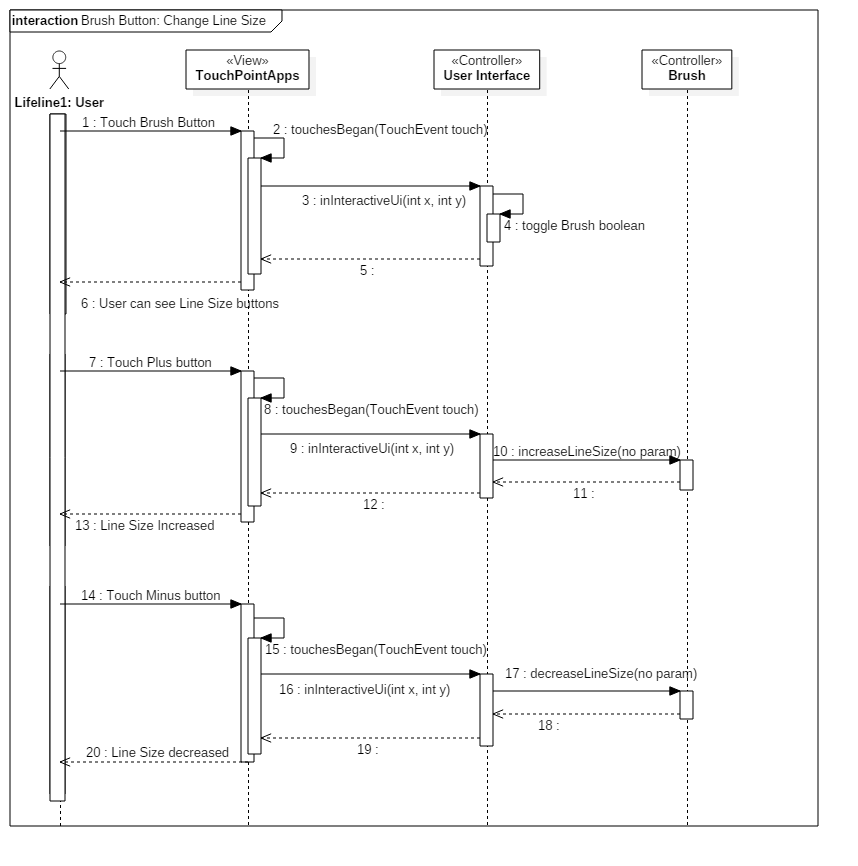
#538

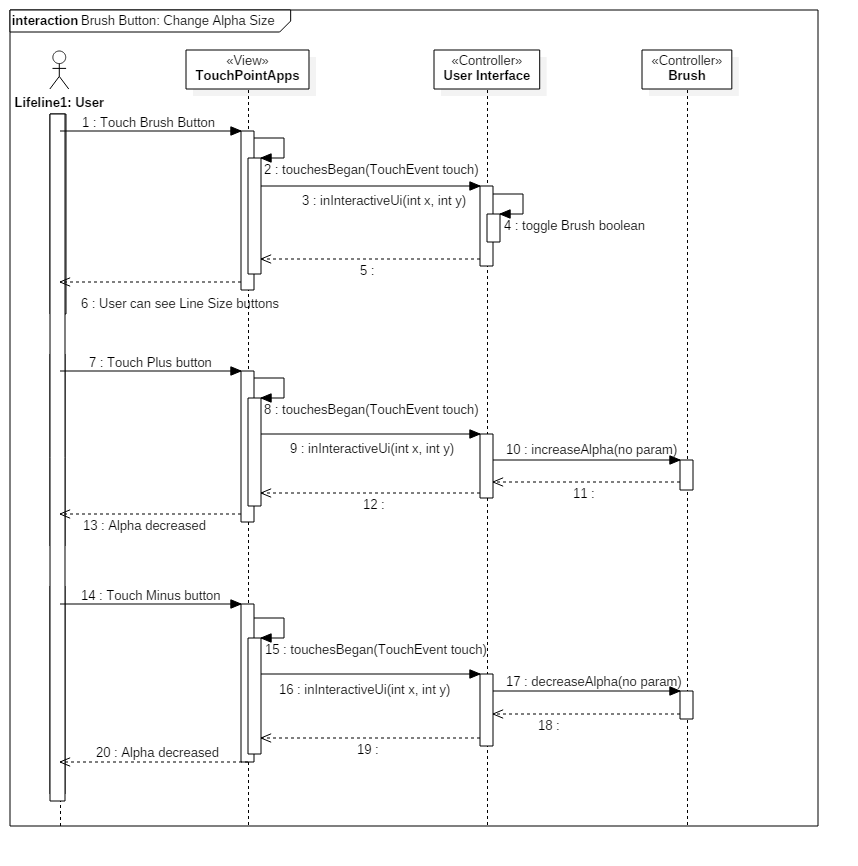


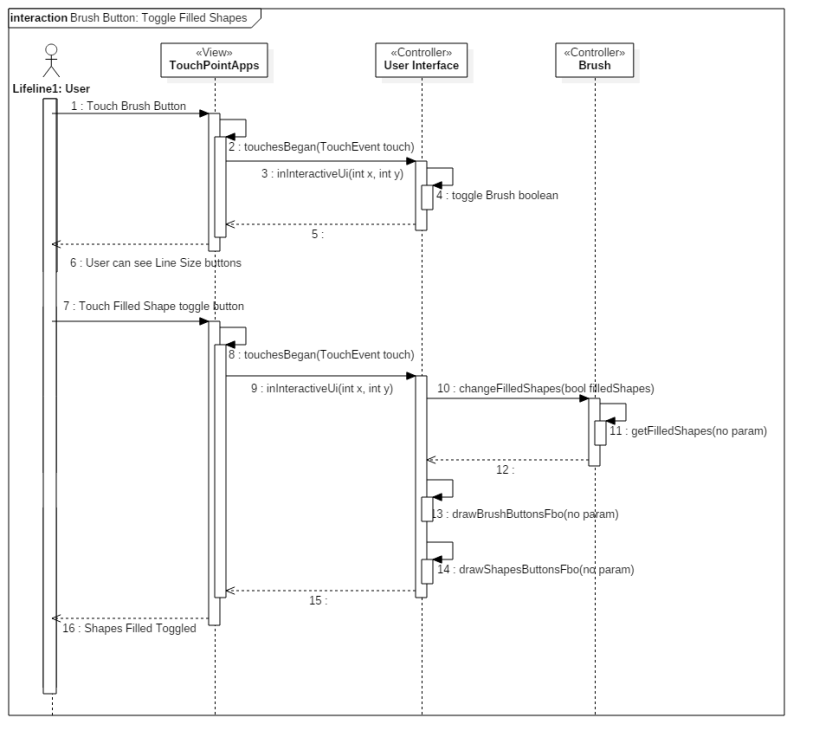
#735

****

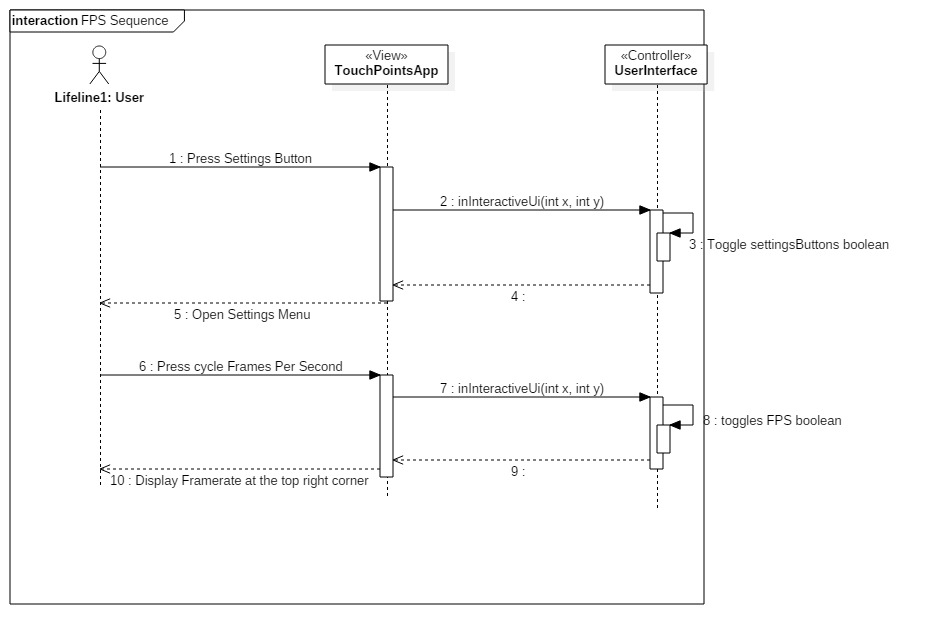
#732

****

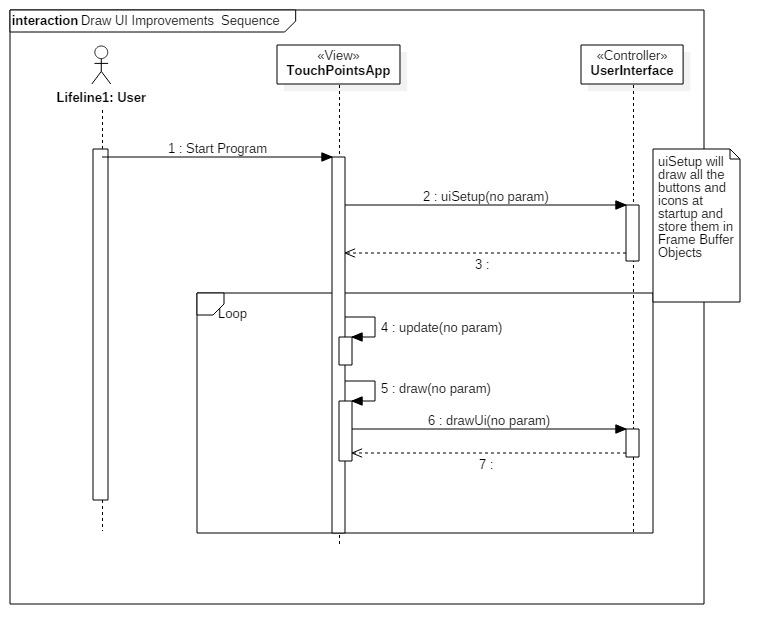
****

****

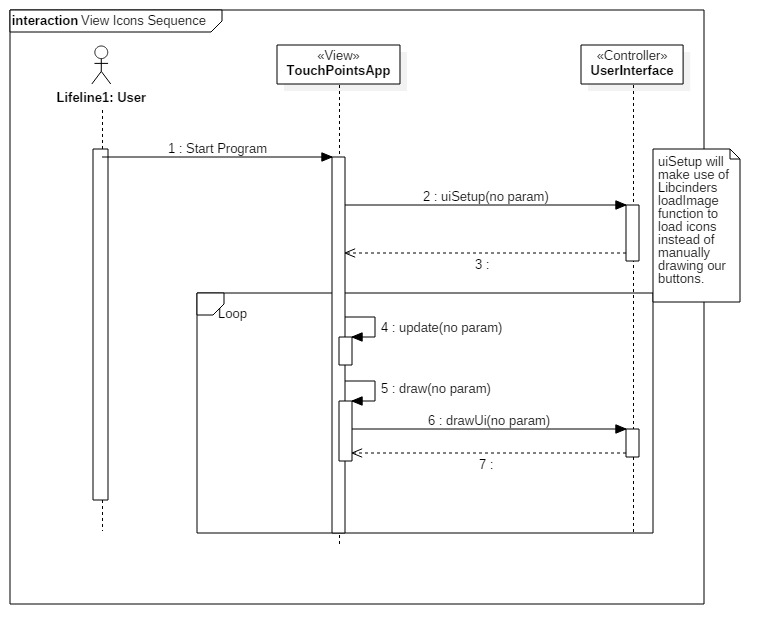
#733



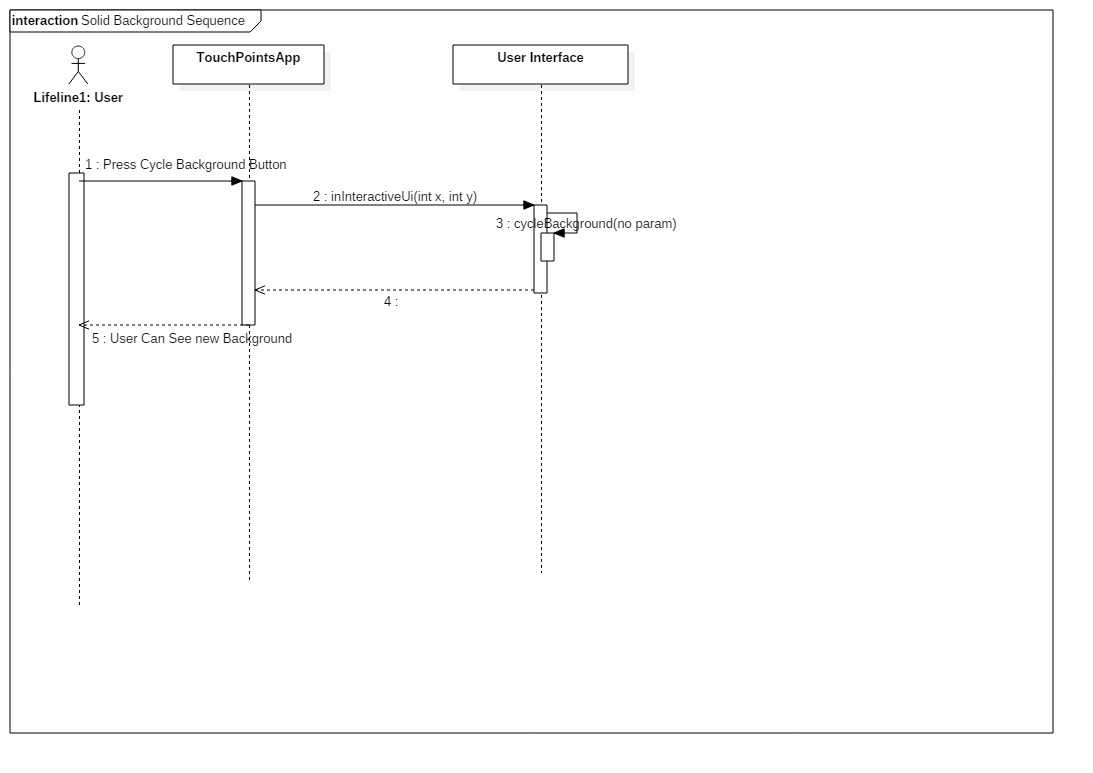
#731



#730

****

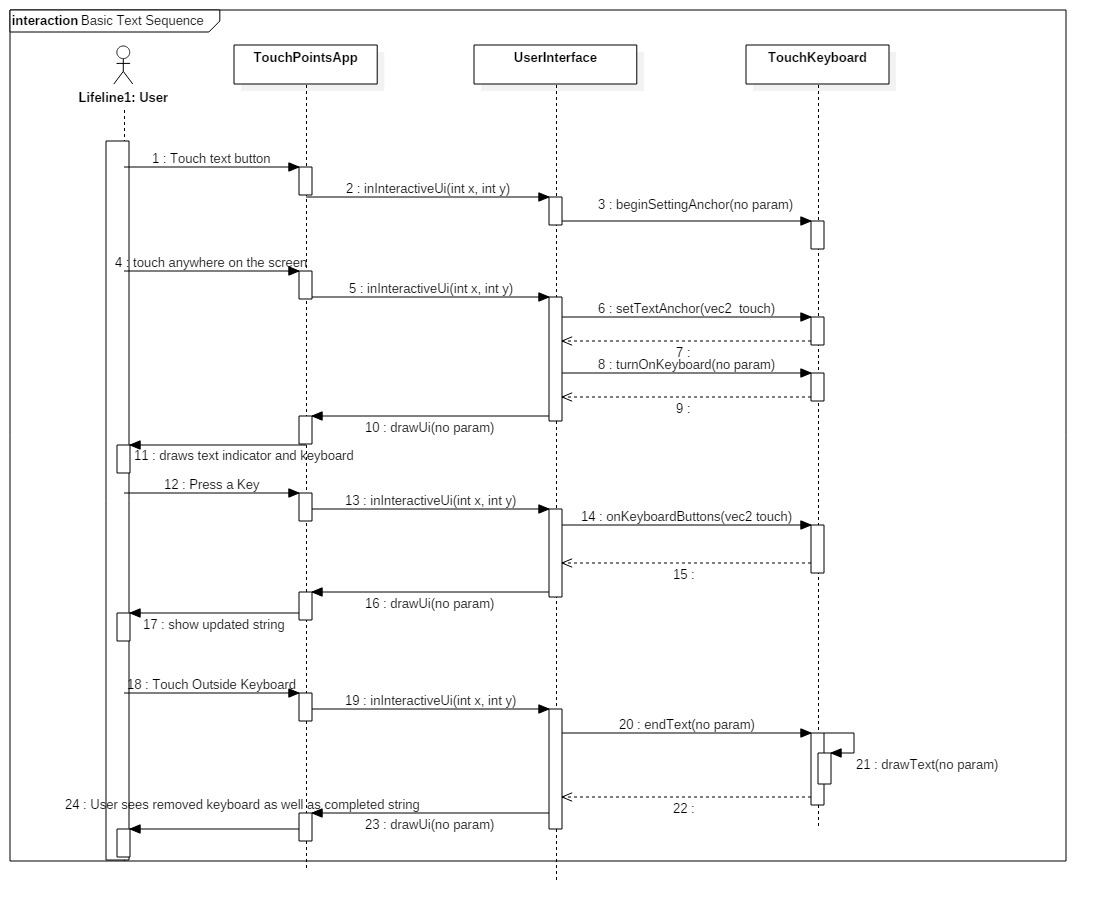
#702



#700



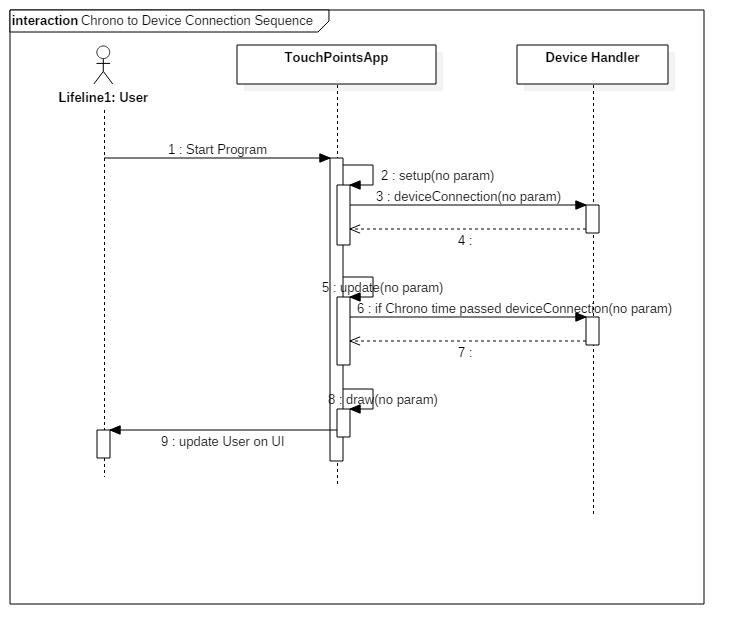
#698



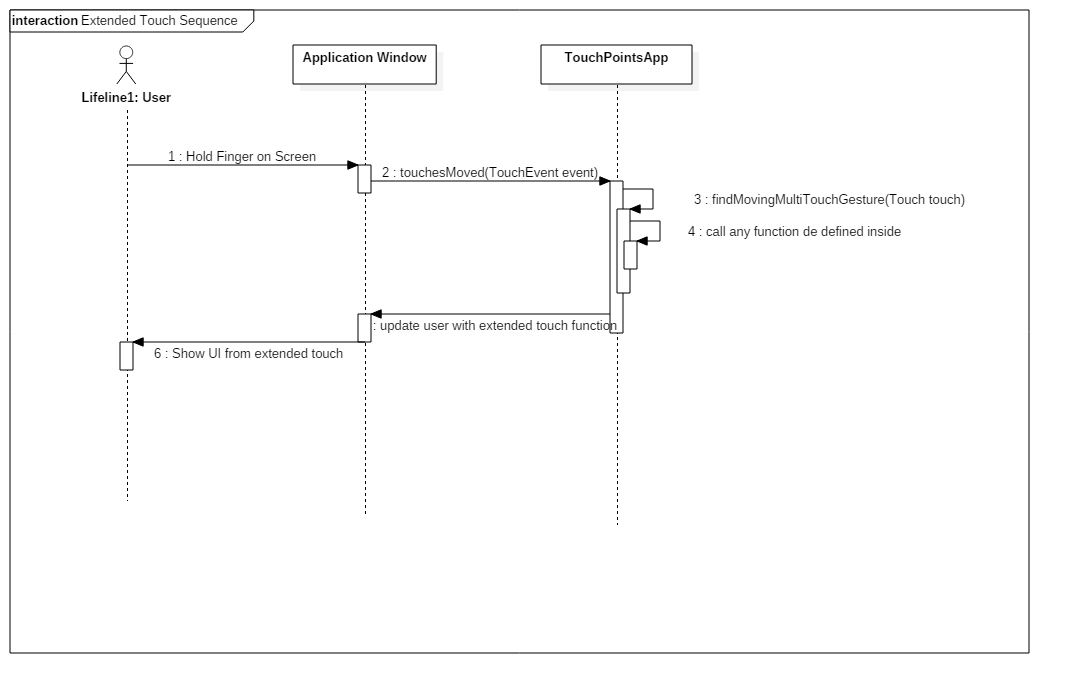
#678



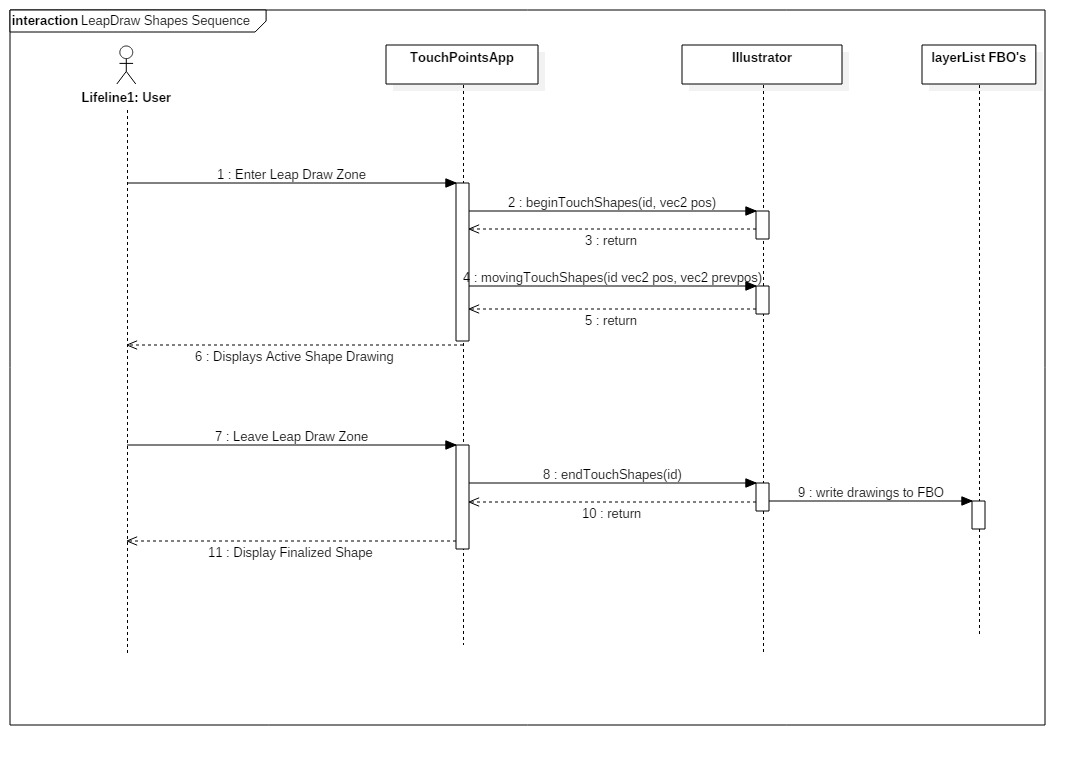
#669



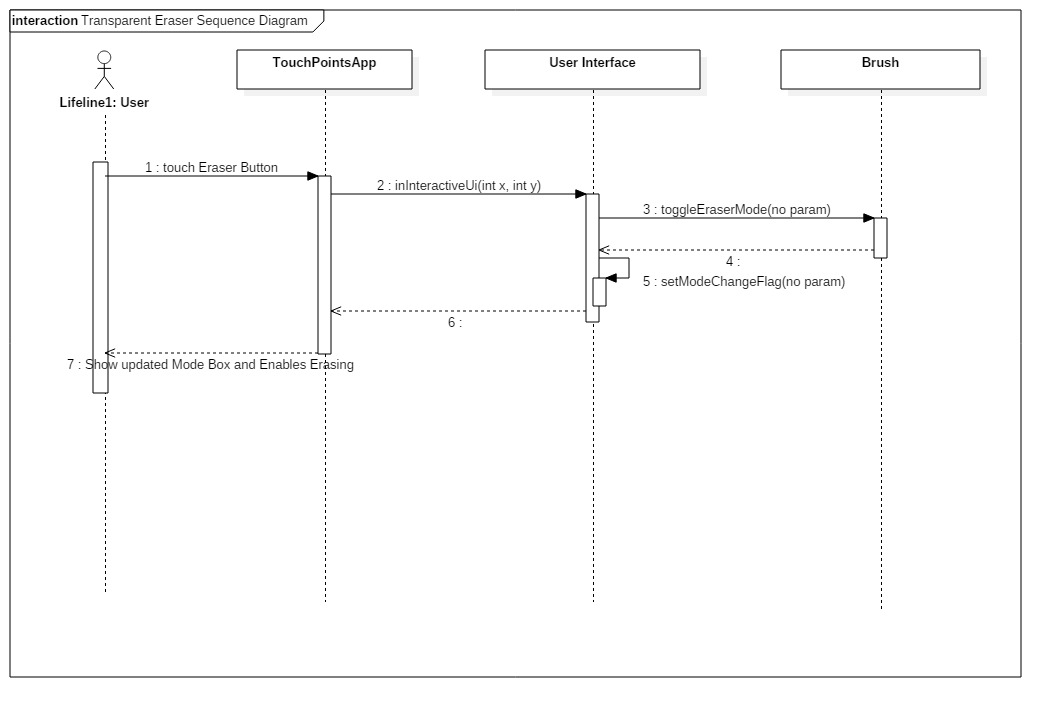
#668



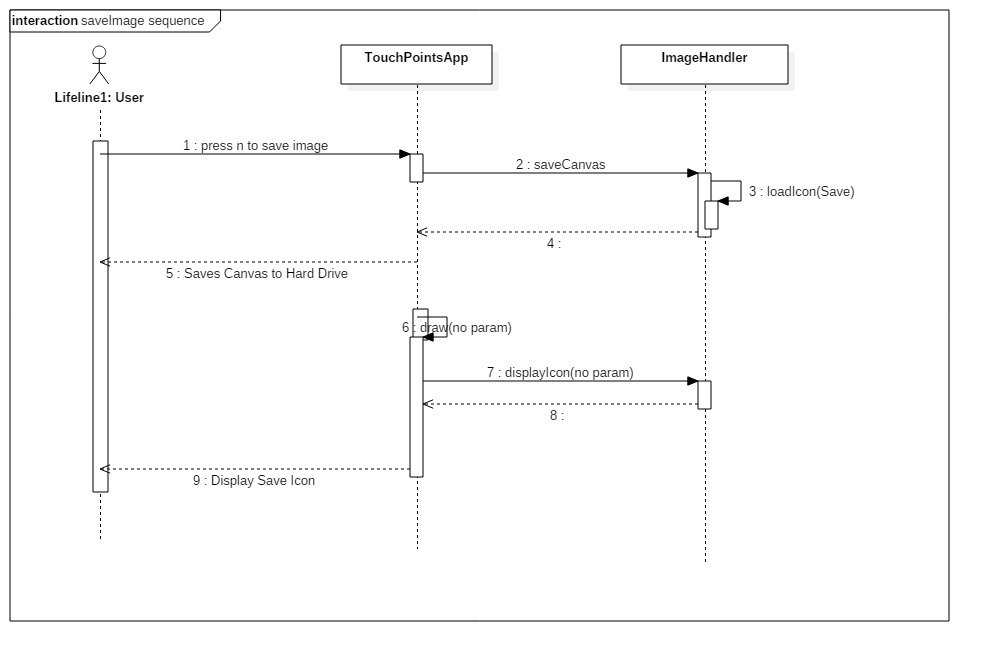
#666



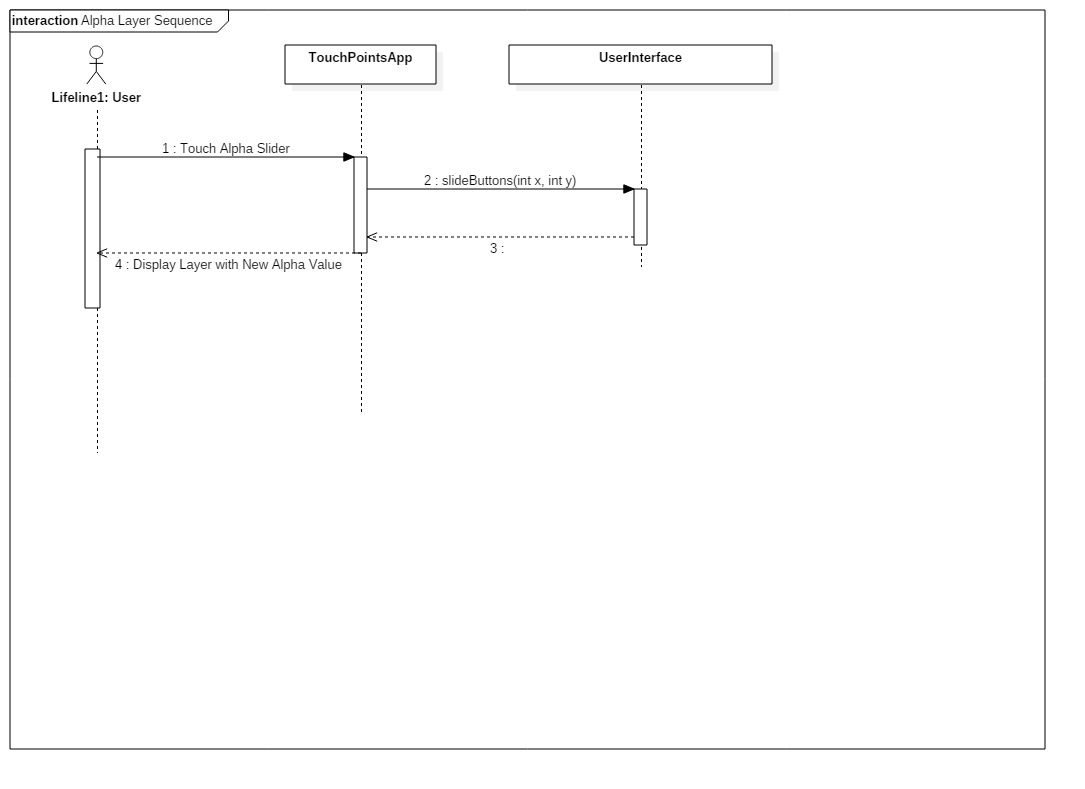
#665



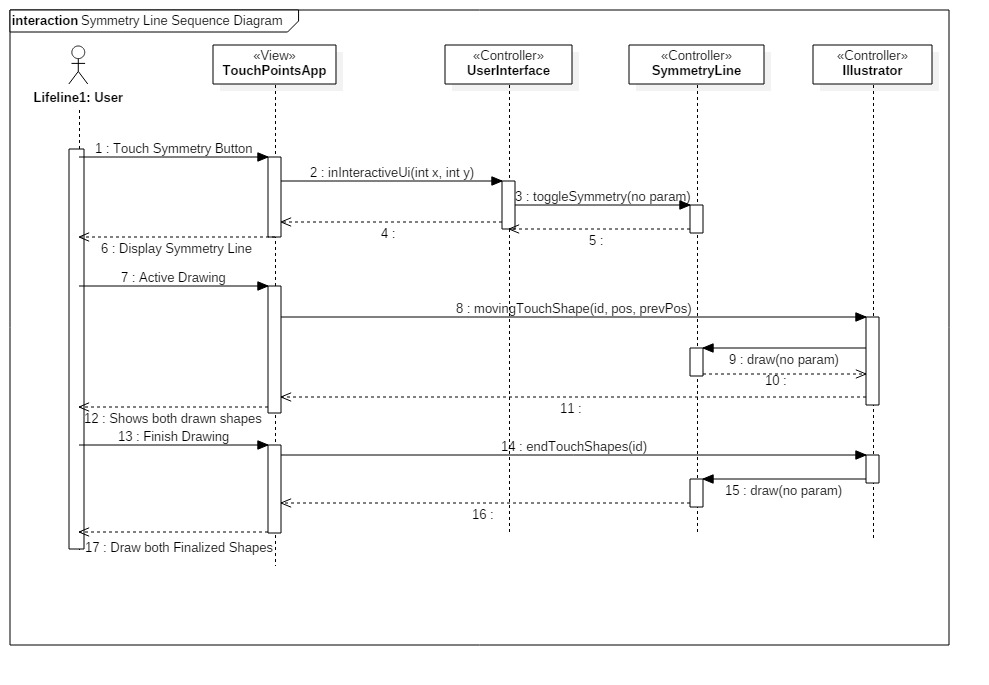
#664



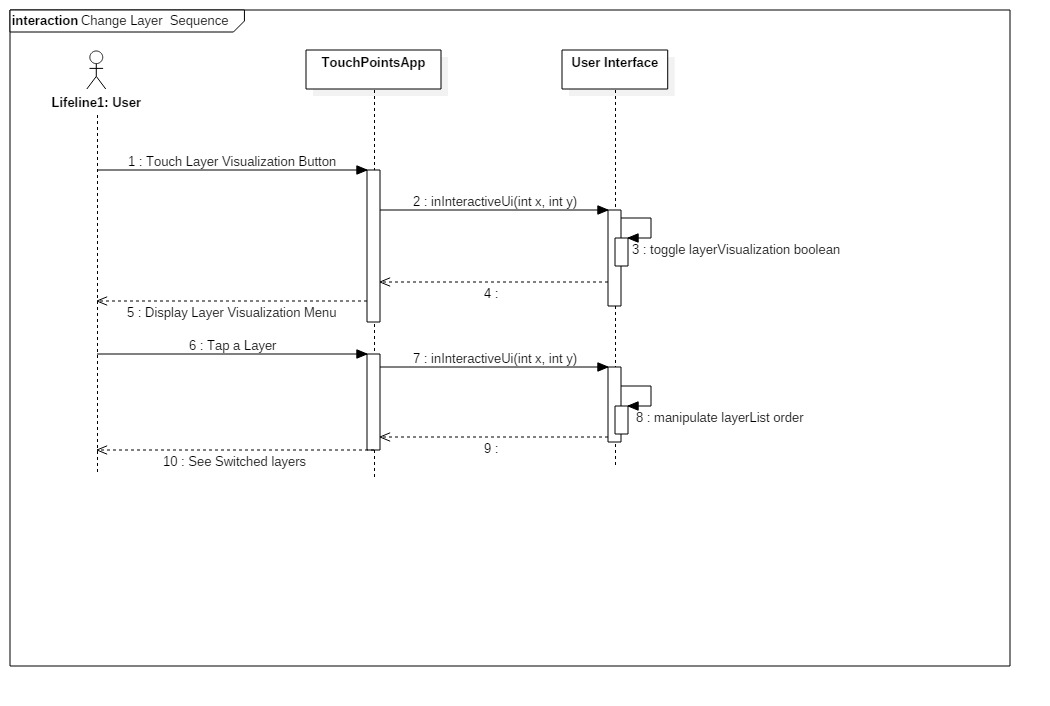
#662



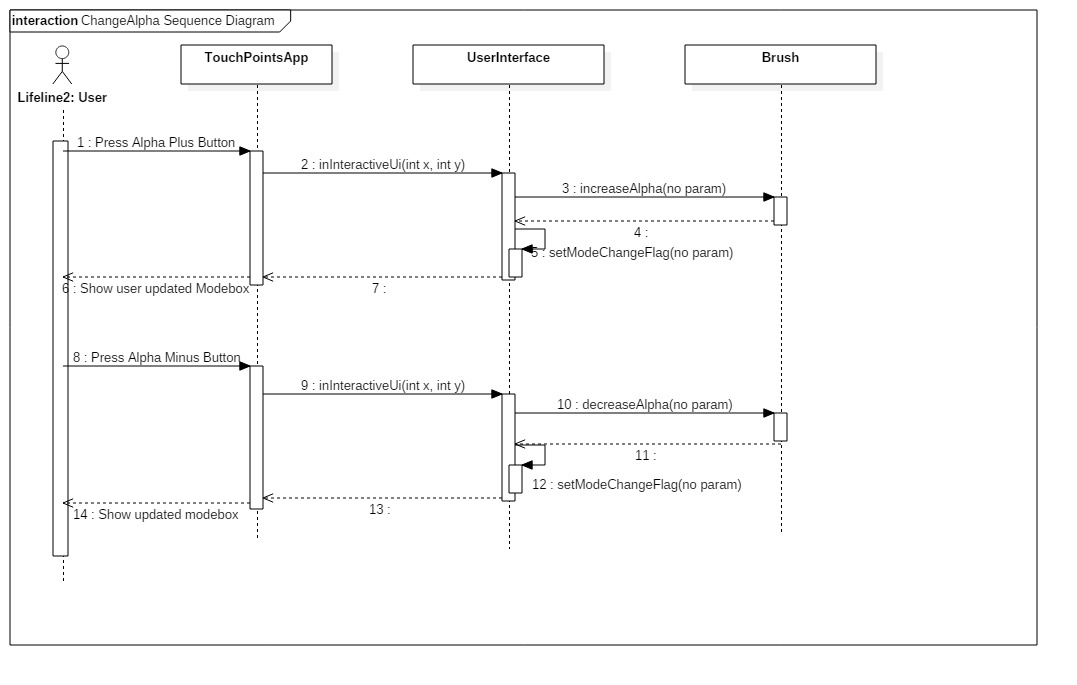
#630



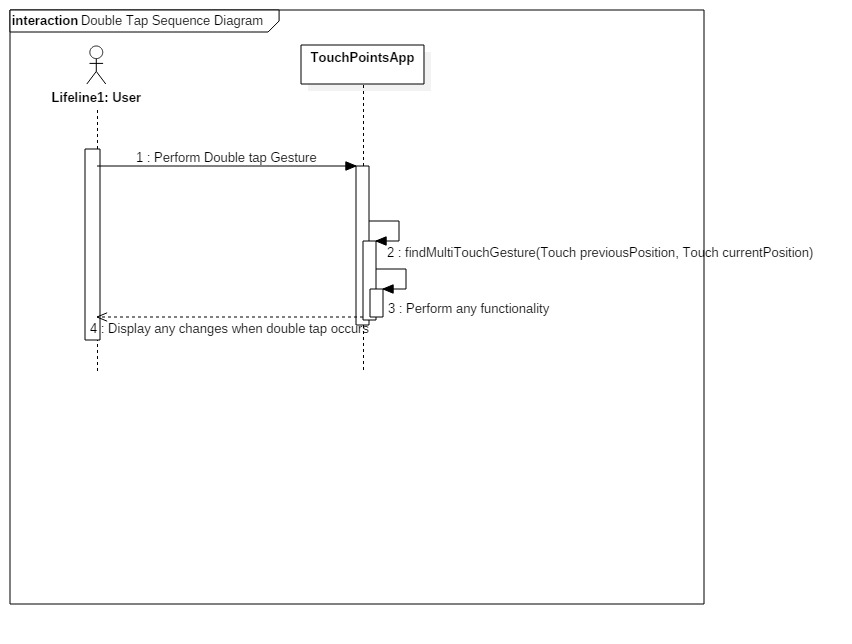
#627

-------------

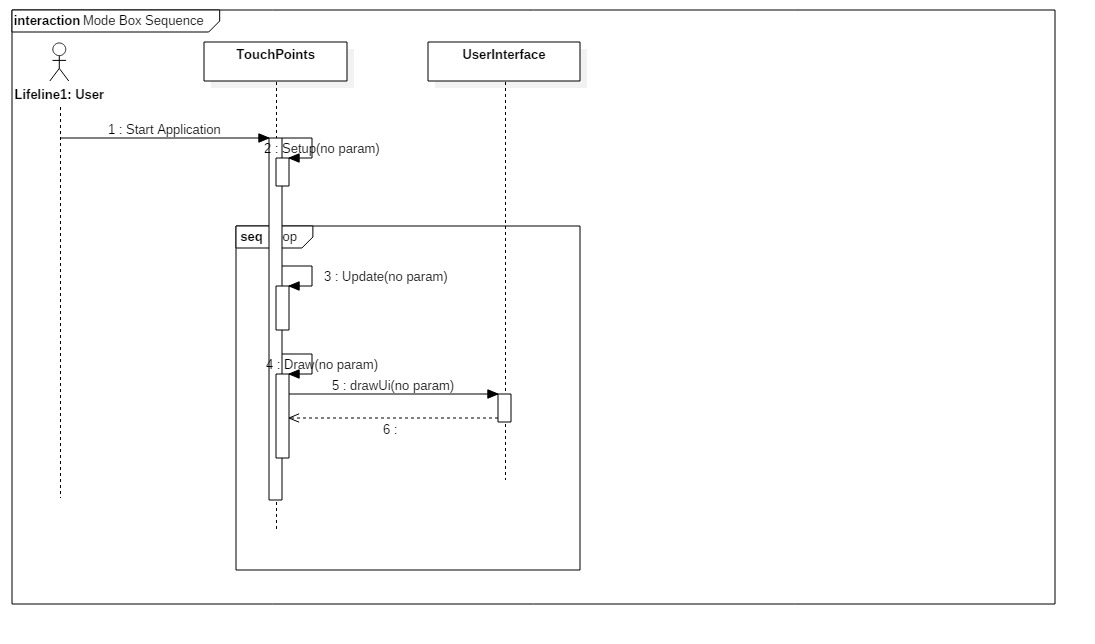
#626



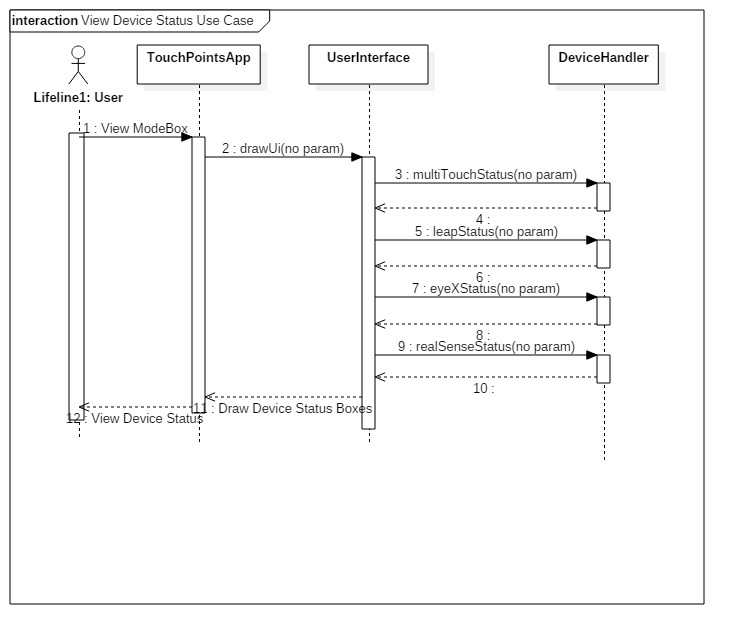
#624



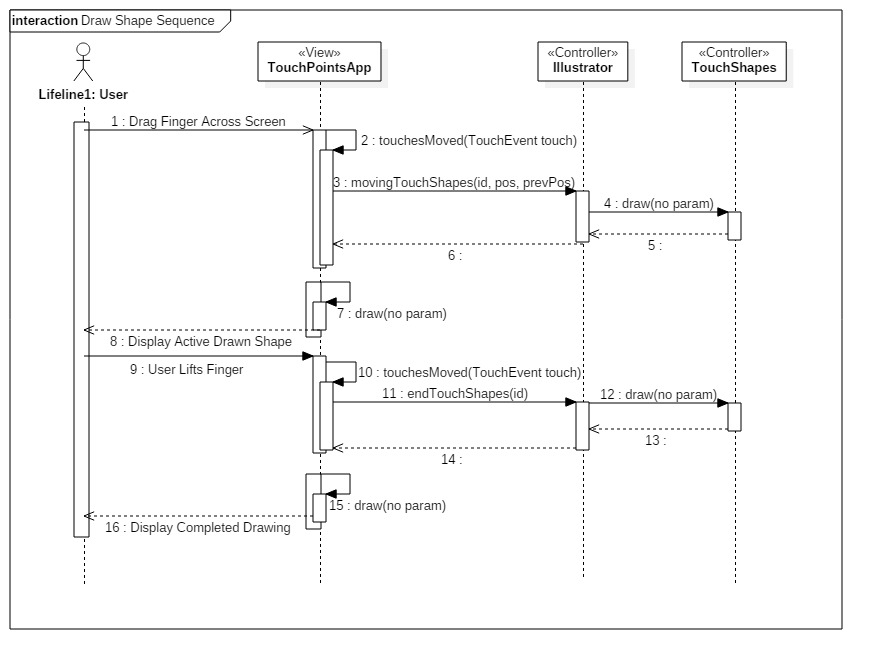
#592



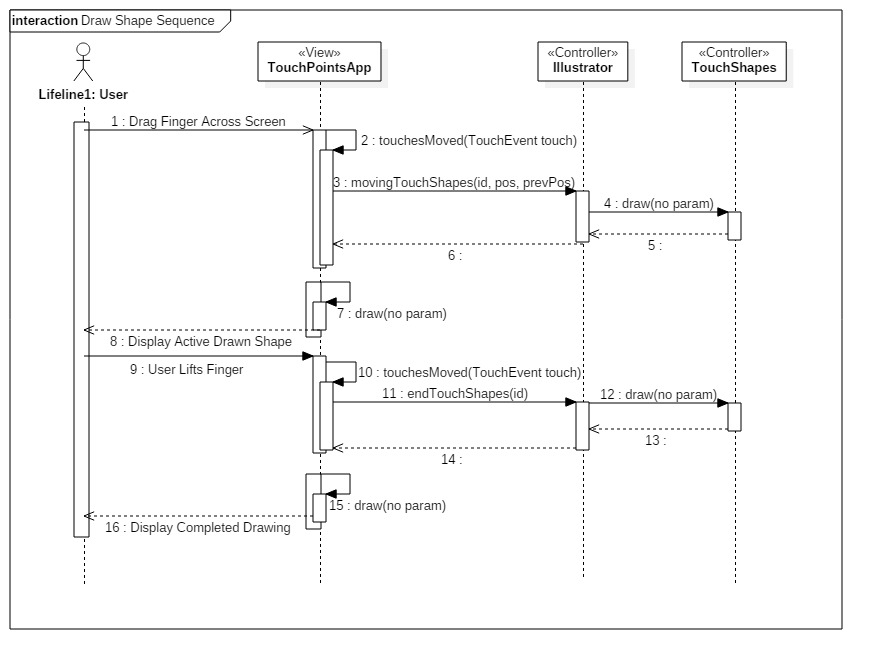
#591



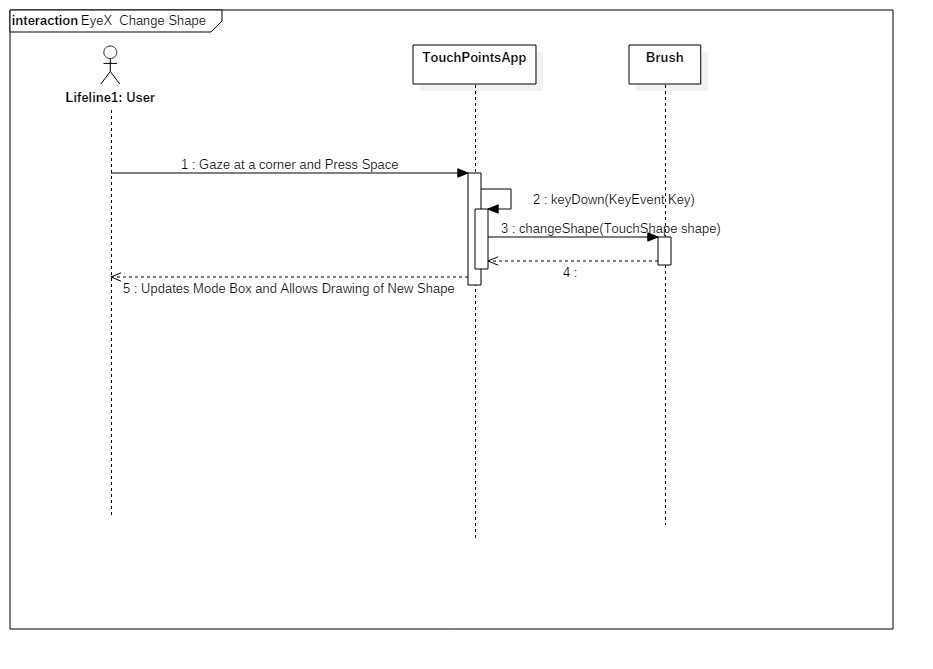
#563



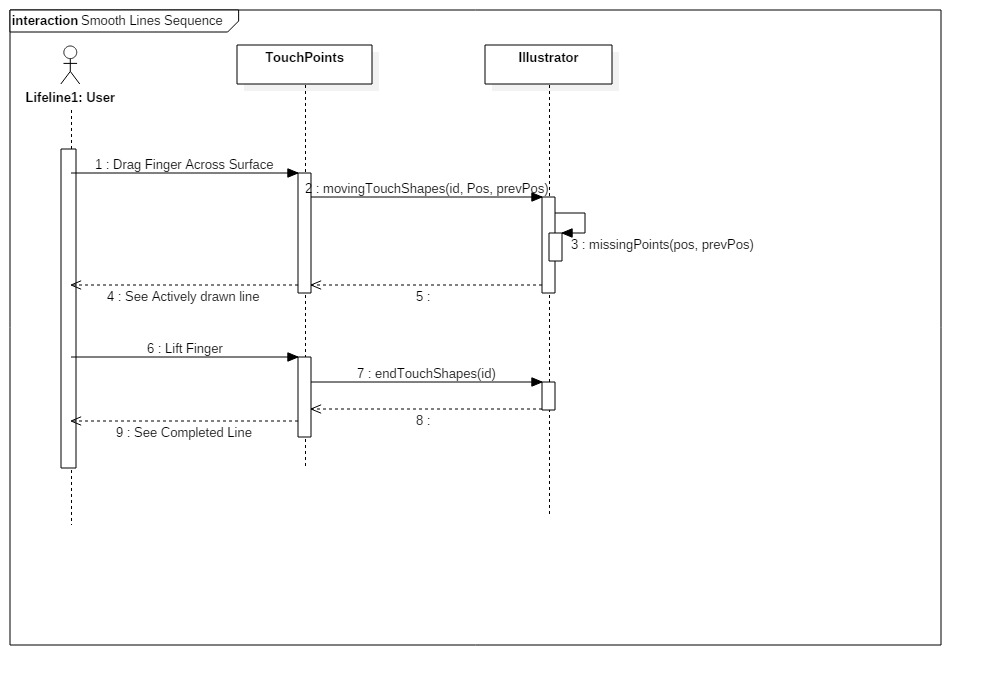
#562



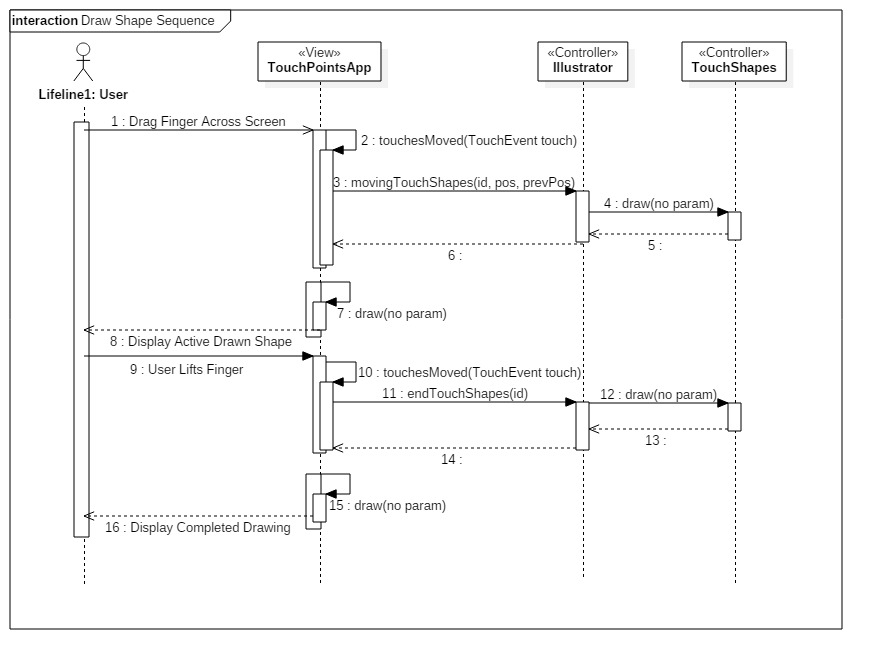
#560



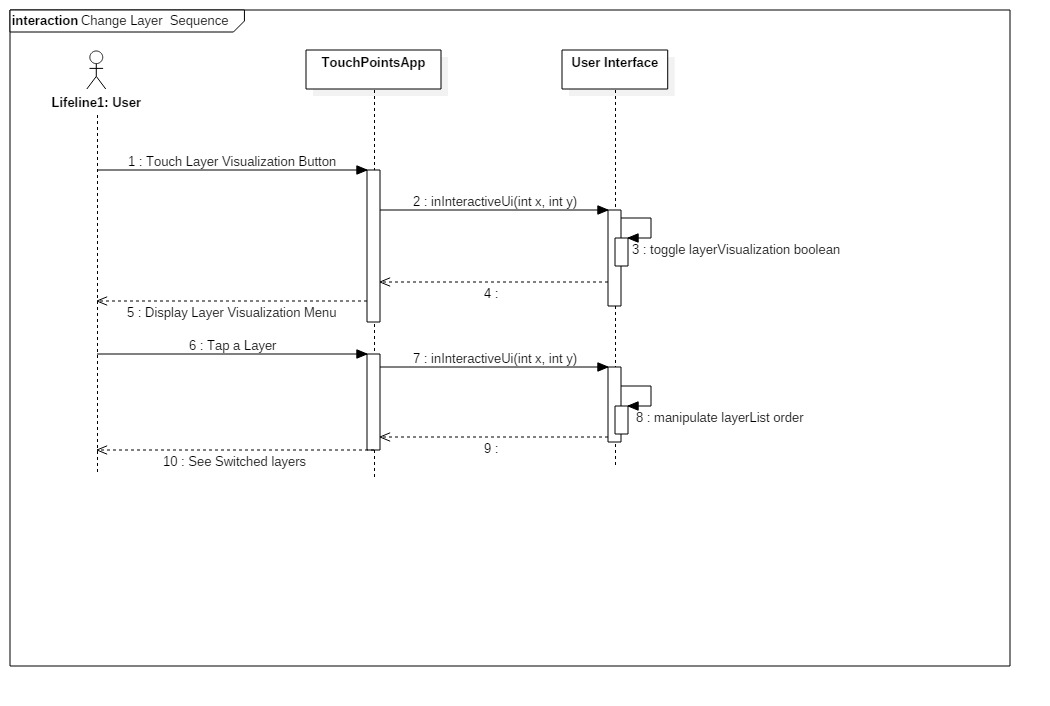
#553



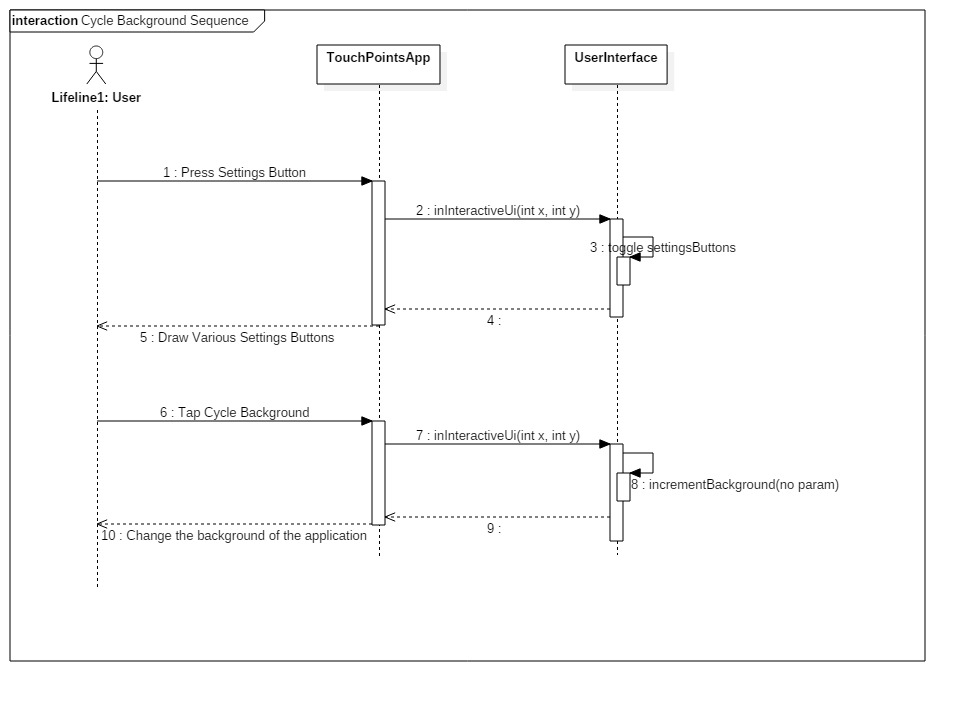
#552



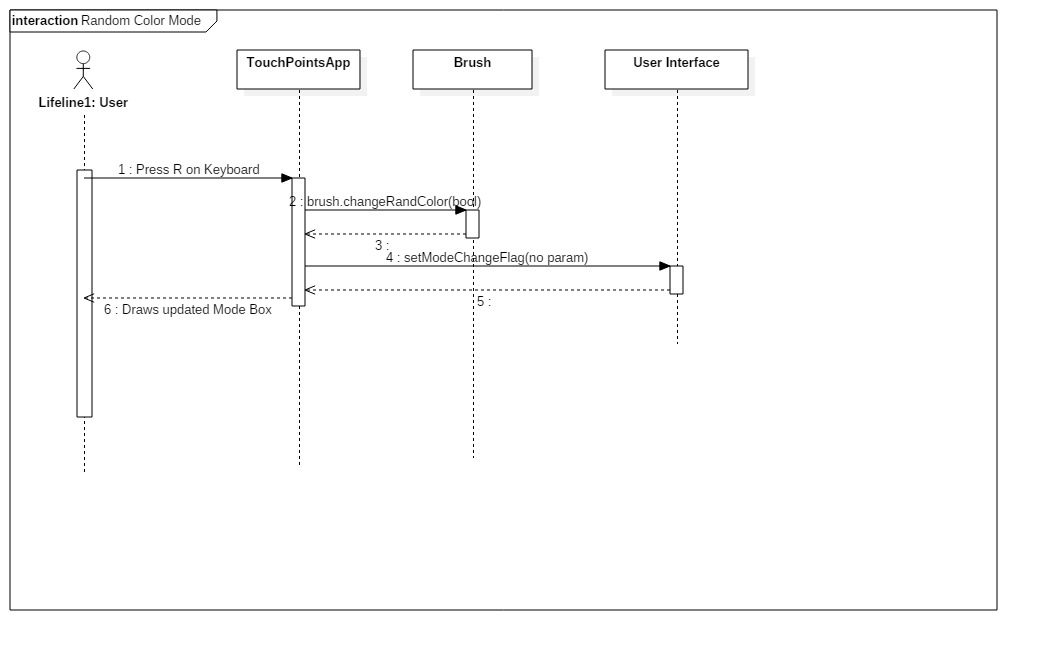
#550



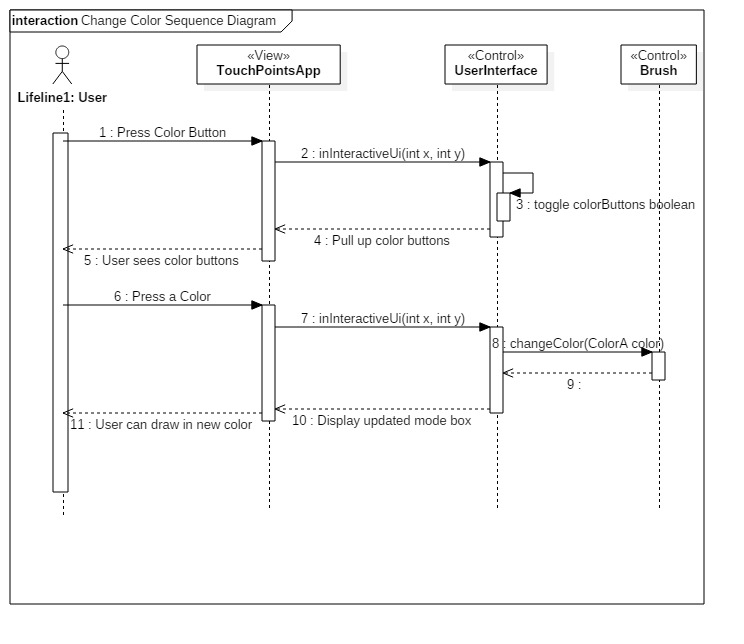
#549



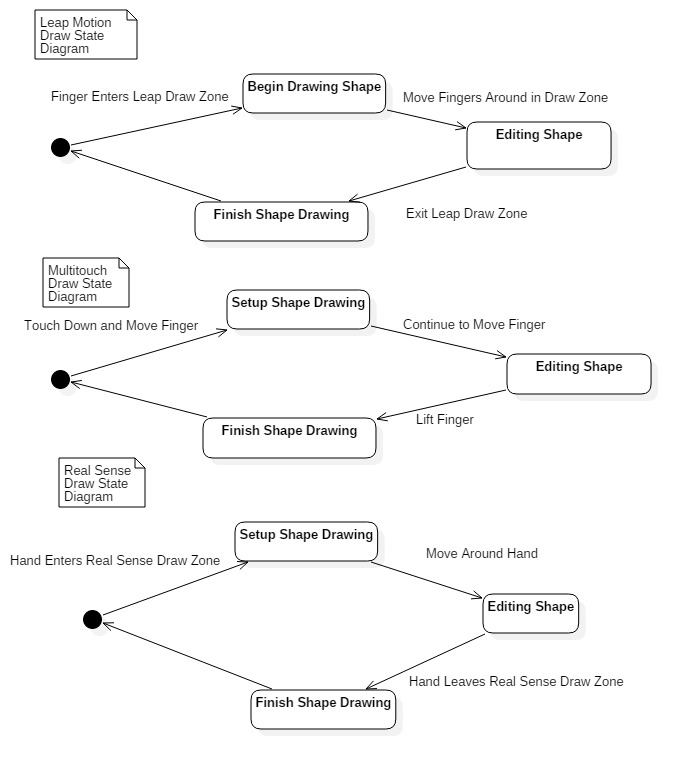
#537



#498

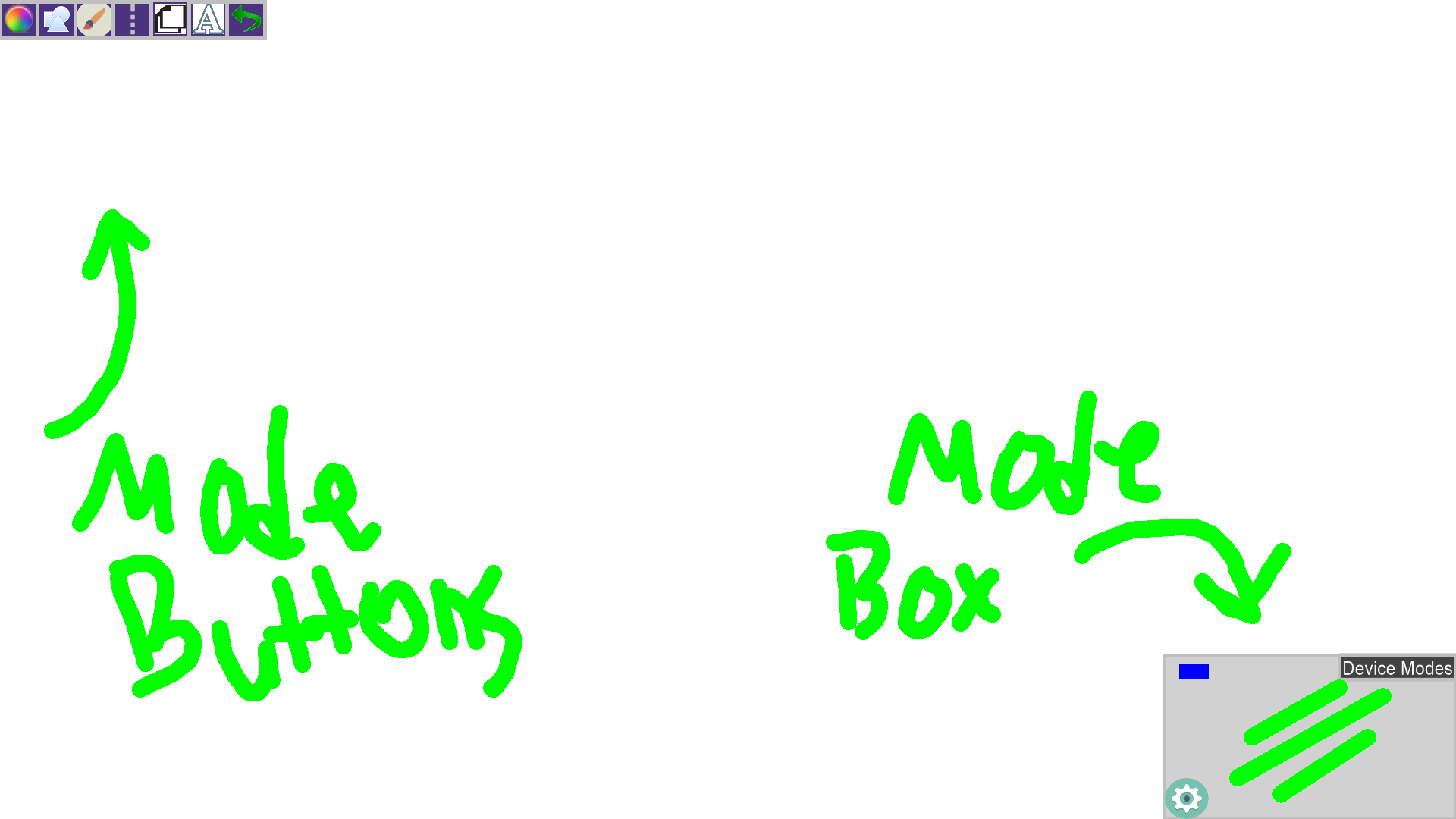


State Chart Diagrams

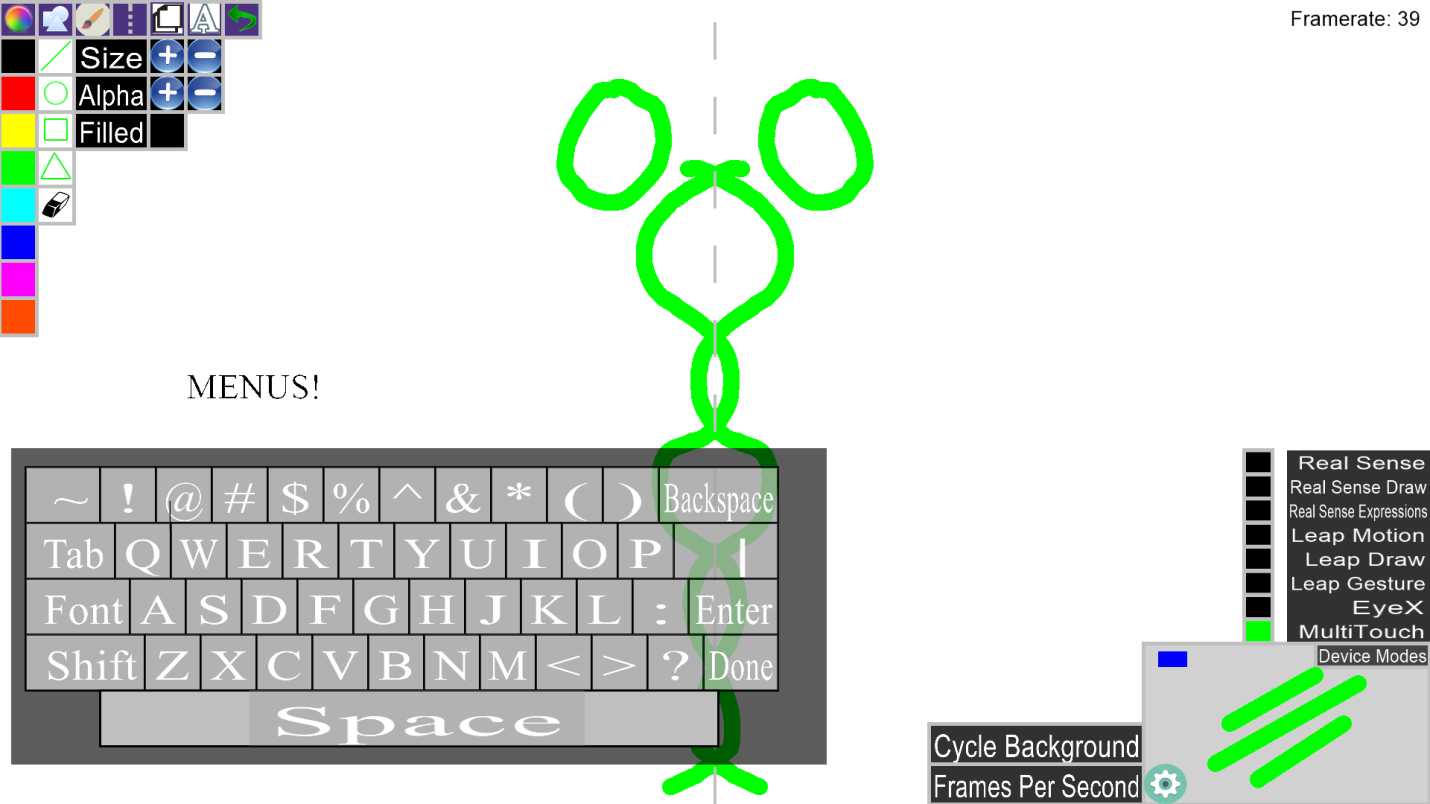


## 

## Appendix B - User Interface Design

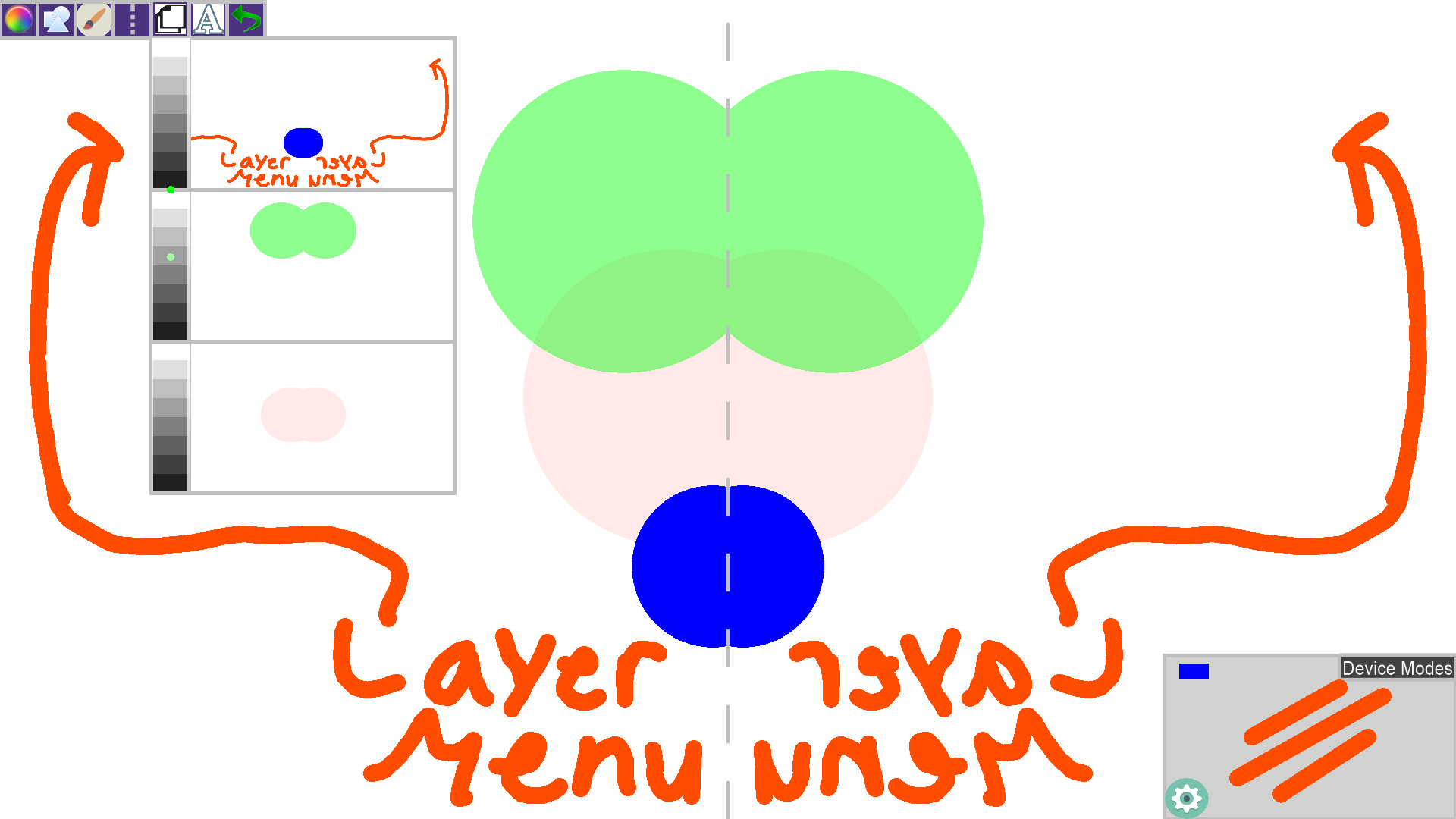


Here is our basic user interface for the Multitouch Device. You can see the mode buttons in the top left, as well as the mode box in the bottom right. The mode box shows a blue rectangle indicating that the multitouch device is plugged in and on. You can see there are three green lines, showing us what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

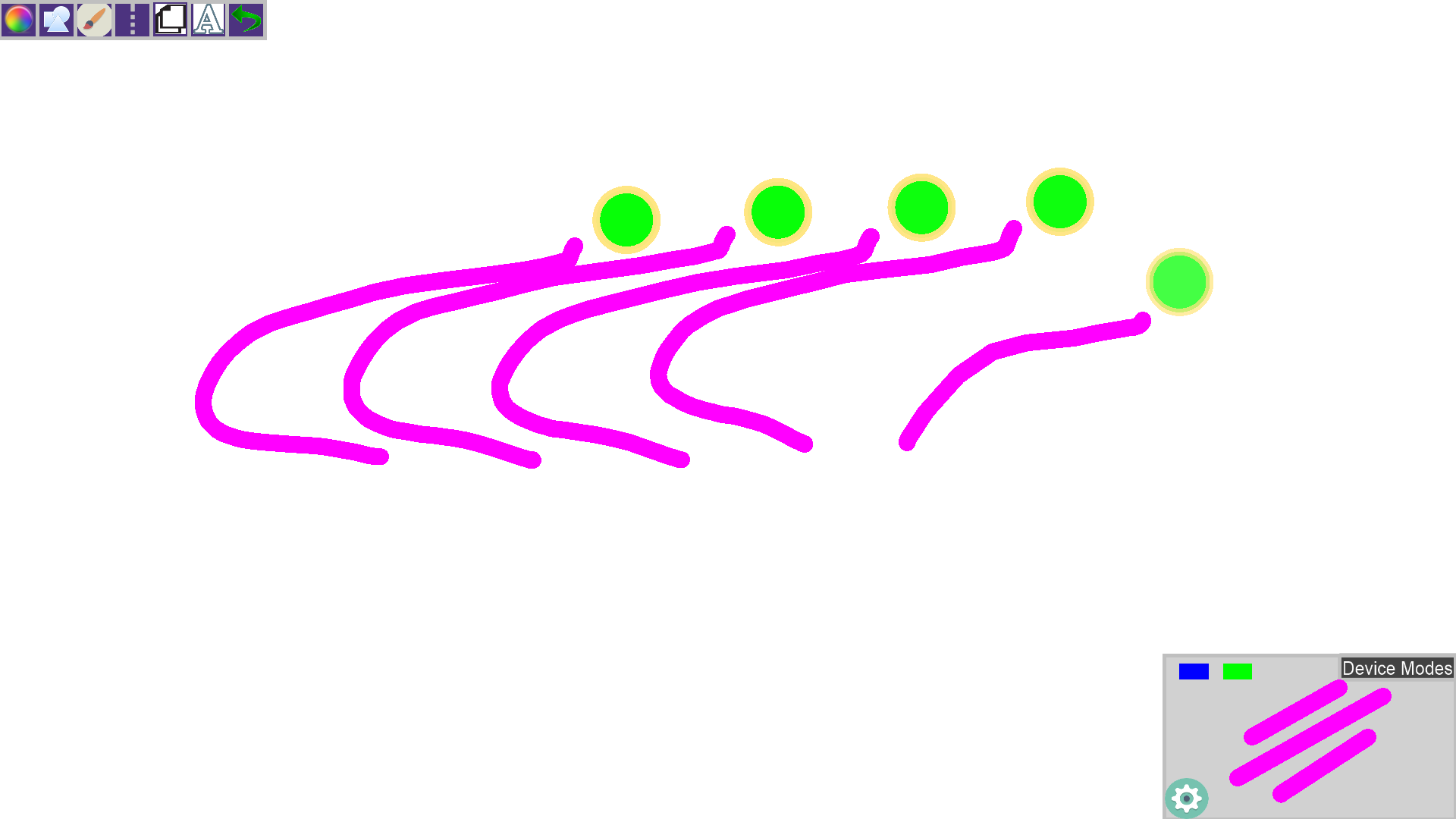


Here we can see various menus open, such as the color menu, the shapes menu, the settings cog (bottom left mode box) and the device modes menu (top right mode box). In addition we can see how the symmetry line looks (dotted line down the center of the screen) as well as the implemented keyboard for basic text. The framerate is also displayed in the top right corner (if you turn it on).

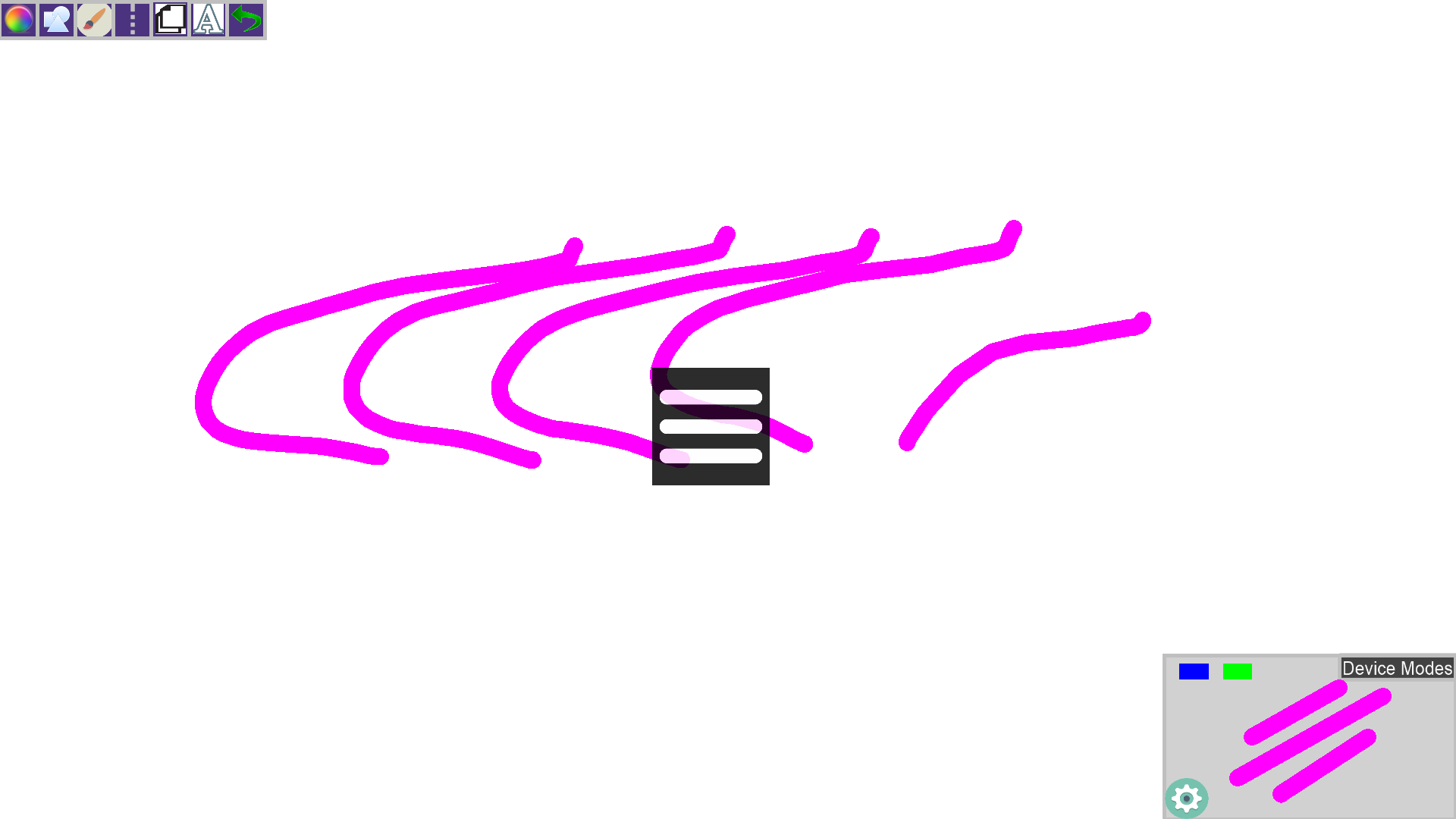
The device modes menu located above the mode box is where the user can select different devices to use, and enable or disable certain functionality.



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.



Here we can see the Leap Motion drawing in action. We display green circles with this cream colors outlines to show where your fingers are in space. For the real sense the indicator is Yellow with a cream colored outline.



Performing Gestures with the leap or real sense device will call up different icons, the one displayed here shows that you changed to line shape.



For the EyeX we simply have two little circles that will jump around the application (displayed above the mode box here). It will pull up the mode box and mode buttons based on where the circles are located. As you can see, they are currently close enough to pull up the mode box.

**Appendix C - Sprint Review Reports**

**Sprint 1**

Date: 1/31/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 8:53PM

End time: 9:00PM

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

* User Story #538 - Change Line Size
* User Story #537 - Implement Random Color Mode
* User Story #498 - Change Line Color
* User Story #??  - Eraser Mode

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* N/A
  + N/a
  + How this should be reflected on the user story definition in Mingle:
    - N/A
* …

**Sprint 2**

Date: 02/14/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 7:15

End time: 7:30

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

* User Story 549
* User Story 552
* User Story 562
* User Story 563
* User Story 553
* User Story 550
* User Story 560
* User Story 548
* User Story 546
* User Story 554

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

N/a. All Accepted.

* User Story #???
  + N/a
  + How this should be reflected on the user story definition in Mingle:
* …

**Sprint 3**

Date:2/27/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 10:40

End time:10 : 55

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

* User Story #623
* User Story #624
* User Story #625
* User Story #626
* User Story #629
* User Story #630
* User Story #631
* User Story #633
* User Story #627
* User Story #628
* User Story #632
* User Story #634
* User Story #635
* User Story #636

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* User Story #???
  + N/a
  + How this should be reflected on the user story definition in Mingle:
* …

**Sprint 4**

Date: 3/13/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 9:25

End time: 9:40

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

* User Story #627
* User Story #623
* User Story #630
* User Story #628
* User Story #624
* User Story #636
* User Story #629
* User Story #626
* User Story #634
* User Story #633
* User Story #632
* User Story #625
* User Story #631
* User Story #635

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

* User Story #???
  + N/a
  + How this should be reflected on the user story definition in Mingle:

**Sprint 5**

Date: 4/2/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 2:00

End time: 2:25

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

* User Story #675
* User Story #671
* User Story #670
* User Story #669
* User Story #667
* User Story #678
* User Story #668
* User Story #666
* User Story #665
* User Story #664
* User Story #662

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

* User Story #672
  + Was rejected because developer found that it cannot yet be integrated. The Orion update for the leap motion device is currently only available in the unity engine.
  + How this should be reflected on the user story definition in Mingle:
    - The acceptance criteria is updated to “Update to new Orion API and SDK when it is released for C++”

**Sprint 6**

Date: 4/17/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:10:30

End time: 10:55

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

* User Story #559
* User Story #649
* User Story #698
* User Story #699
* User Story #700
* User Story #701
* User Story #702
* User Story #703
* User Story #706
* User Story #707
* User Story #708

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

N/a. All Accepted.

* User Story #???
  + N/a
  + How this should be reflected on the user story definition in Mingle:
* …

**Sprint 7**

Date: 5/1/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:11:55

End time: 12:10

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

* User Story #701
* User Story #700
* User Story #699
* User Story #698
* User Story #702
* User Story #708
* User Story #707
* User Story #706
* User Story #703
* User Story #649
* User Story #559

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Spring Planning meeting.

N/a. All Accepted.

* User Story #???
  + N/a
  + How this should be reflected on the user story definition in Mingle:
* …

## Appendix D - Sprint Retrospective Reports

**Sprint Retrospective 1**

Date: 1/31/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:9:00

End time:9:20

What went wrong?

* Did we do a good job estimating our team's velocity?
  + No. We only did 20 points.
* Did we do a good job estimating the points (time required) for each user story?
  + After setting up libcinder and getting our first user story completed we estimated the other user story fairly well, but still not great.
* Did each team member work as scheduled?
  + Yes we worked as scheduled, including over time!
  + We had a small hiccup for our sprint planning meetings, started later than planned.
* Magnum
  + Failed as a good development environment.
  + Spent many hours working on compiling
  + Spent even more hours trying to get working samples
  + Spent some time on working examples.
  + Eventually it was scrapped for libcinder.

What went right?

* After switching to libcinder our velocity skyrocketed. After looking at a few examples we managed to get our user stories completed in a great amount of time.
* We did learn a lot about C++ and the various devices from Professor Ortega and online videos.
* Great communication between product owner and developers.
* All team members are working hard and on time, very easy to get a hold of.

How to address the issues in the next sprint?

* How to improve the process?
  + Sprint planning with real user stories from day 1.
  + Already have a dev environment set in stone (no magnum, switched to lib cinder), so it should be easier to develop user stories.
  + Point Estimation - Longer Sprint planning II and doing it properly with correct user stories.
* How to improve the product?
  + Smooth the larger lines.
  + Implement LeapMotion Device.

**Sprint Retrospective 2**

Date: 02/14/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:7:30

End time:7:45

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Estimations were fairly accurate, but,  Sunday, the last day of the sprint,  we spent a little more time than anticipated finishing our work before the meeting.
* Did we do a good job estimating the points (time required) for each user story?
  + Andrew - Underestimated the line smooth functionality. Took a bit longer than anticipated.
  + Garrett - Underestimated drawing a smooth line. Took extra time fixing performance issue.
* Did each team member work as scheduled?
  + Yes!
* Design philosophy meeting got pushed back.
* Performance issues with drawing lines (multitouch).
* EyeX is not as precise as previously thought.
* Worked separately on different devices.

What went right?

* Got the devices to work.
* Got our user stories completed.
* Product owner was impressed by how much we got done.

How to address the issues in the next sprint?

* How to improve the process?
  + More meetings between Garrett and Andrew to structure the code better for the multiple devices
* How to improve the product?
  + Need to improve performance issues (Andrew)
  + Implement a good user Interface.
  + Implement Shape Inheritance

**Sprint Retrospective 3**

Date: 02/27/2016

Attendees:Garrett, Andrew

Start time:10:55

End time: 11:15

What went wrong?

* Did we do a good job estimating our team's velocity?
  + We got everything done, maybe have worked a bit extra to account for unexpected meetings.
* Did we do a good job estimating the points (time required) for each user story?
  + Feedback implementation took a bit longer than anticipated.
* Did each team member work as scheduled?
  + We had a few hiccups the days where we had meetings. We worked later a lot later in the day when the meetings got pushed back.
  + Sprint review got slightly postponed.

What went right?

* Code integration went well.
* Finally learned a few core components of framebuffers which will surely help in the following sprints.
* Much better time estimations.
* Git integration is a lot more smooth now.

How to address the issues in the next sprint?

* How to improve the process?
  + Try to get all our meetings scheduled ahead of time. They need to be at least known at the start of the sprint, even if the schedule is slightly malleable.
  + Should have design meetings in previous sprints.
* How to improve the product?
  + Implement alpha shading to our shape colors.
  + Get icons for our buttons and feedback popups.
  + Improve device status so it can be dynamic (currently only leap is dynamic)
  + Turn off gestures while drawing with leap.
  + Add UI for Layering
  + Implement ‘Modes’ for using specific devices a certain way.

**Sprint Retrospective 4**

Date: 3/13/2016

Attendees:Garrett, Andrew

Start time: 9:40

End time:9:55

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.
  + LibUSB provided a couple of unsuspecting hiccups. Won’t be our end all be all solution.
* Did each team member work as scheduled?
  + Yes.
* Lib USB caused a lot of performance issues.

What went right?

* Our presentations went well
* Completed all of our user stories on time, even with the event of our checkpoint I

How to address the issues in the next sprint?

* How to improve the process?
  + Try to find a way to better communicate with the OpenHID Lab.
  + Continuously refactor code.
    - Ensure code is as separate as possible
    - This should be an ongoing process.
* How to improve the product?
  + Refactor / Separate out code.
  + Draw shapes with leap motion.
  + Gamification elements for kids?
  + Add RealSense Device.
  + Continue searching for good ‘device detection’ paths.
  + Interactive UI for Leap Motion
  + Continue working on the user guide.
  + Improve LIBUSB Performance Issues.

**Sprint Retrospective 5**

Date: 4/2/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time: 2:20 PM

End time: 2:55 PM

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes, some slight overtime due handling new device.
* Did we do a good job estimating the points (time required) for each user story?
  + Fairly well.
    - User Story 670 Implement Undo Button took a bit longer than expected.
    - Some overtime with setting up real sense.
* Did each team member work as scheduled?
  + Yes.
* Again the product owner had to push back the meeting despite sending out a message and Gmail meeting early in the sprint. The product owner did get sick however, so this may have been the cause for the postponement. Hopefully it doesn’t continue to happen.

What went right?

* We finished everything on time despite some very interesting bugs that popped up (With some slight overtime)
* We Integrated a new device into the program (Intel RealSense Camera)
* Cleaned up how devices are handled.

How to address the issues in the next sprint?

* How to improve the process?
  + We should write the User Guide before writing the code (So we know what exactly the user should experience). We should update it after completion still to ensure that it turned out as planned.
* How to improve the product?
  + Continue working on the user guide.
  + Improve LIBUSB Performance Issues (Add Chrono).
  + Better Icons for Buttons
  + Improve how UI is drawn (performance wise) - Low priority.
  + RealSense - Additional Functionality
  + Program Startup Display Image

**Sprint Retrospective 6**

Date: 4/17/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:10:55

End time: 11:10

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes. We estimated perfectly this time. We finished on time.
* Did we do a good job estimating the points (time required) for each user story?
  + Yes. All went about the correct time.
* Did each team member work as scheduled?
  + Yes.
* Again the product owner had to push back the meeting due to being out of town this week. Had to postpone to Saturday.

What went right?

* We finished everything on time (no real overtime except a video the product owner wanted us to do).
* Added more functionality for new real sense device.
* Improved performance dramatically by reducing the amount of device checks.

How to address the issues in the next sprint?

* How to improve the process?
  + We should change our sprint review schedule because the product owner seems to have problems arriving on fridays.
* How to improve the product?
  + Continue working on the user guide.
  + Better Icons for Buttons (found Icons, need to implement them now).
  + Improve how UI is drawn (performance wise, draw into frame buffers once) - Low priority.
  + Program Startup Display Image
  + Refactor some more code
    - Leap Motion and EyeX out of Touch Points
    - Ensure that ‘Update’ does all the updates and ‘Draw’ does our ui and canvas drawing.
  + Real Sense ‘Asynchronous’ Implementation (May improve real sense performance).
  + Buffer Multitouch input.

**Sprint Retrospective 7**

Date: 5/1/2016

Attendees:Garrett, Andrew, Francisco Ortega

Start time:12:10

End time: 12:45

What went wrong?

* Did we do a good job estimating our team's velocity?
  + Yes. We managed to finish our tasks early and had enough time to finish touching up our user guides as well as old documentation.
* Did we do a good job estimating the points (time required) for each user story?
  + Yes.
* Did each team member work as scheduled?
  + Yes.
* This sprint we decided to set the EoS Meetings to sunday at noon. This means we didn’t wait up friday for our product owner because he is normally available sundays. Worked much better!

What went right?

* Setting up the meeting on sunday.
* Our Testing was good. Retesting old features, most of them seemed to work as intended but we did find a few bugs that we managed to fix.
* New documentation is looking good!

How to address the issues in the next sprint?

* How to improve the process?
  + Continue to change day for our EoS Meetings.
  + Do documentation right the first time!
* How to improve the product?
  + Refactor more code. Move Leap Motion, EyeX, and Multitouch out of TouchPointsapp.
  + Buffer Multitouch Input.
  + Implement Threading
  + Gamification elements added into a tutorial system.
  + How to improve it for multi modal ouputs (iPad, VR, etc).
  + Speach recognition
  + Active Pen.

## Appendix E - Sprint Planning Meetings

**Sprint Planning 1**

Date: 1/19/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 4:30

End time: 5:30

After discussion, the velocity of the team were estimated to be ??.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

* User Story #497 up to User Story  (Learning C++, Technologies, and Magnum).

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

* Both Garrett and Andrew were assigned the same user stories, to learn C++, Magnum, download and play with the technology’s provided including Microsoft Kinect, Tobii eyeX, Acer Multitouch Screen, Leap Motion Controller. In addition to reviewing the previous semesters Framework and using that to test some of the technologies, so we are familiar with the environment.

**Sprint Planning 2**

Date: 1/31/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 9:20

End time: 11:33

After discussion, the velocity of the team were estimated to be 49.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* User Story 549
* User Story 552
* User Story 562
* User Story 563
* User Story 553
* User Story 550
* User Story 560
* User Story 548
* User Story 546
* User Story 554

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

Andrew :

#549 - 1 points - Change Background Color

#552 - 3 points - Create Circle Shape

#562 - 2 points - Implement Triangle Shapes.

#563 - 1 Points - Implement Rectangle Shapes.

#553 - 5 points - Smooth Lines

#550 - 8 points - Add Multiple Layers

#560 - 5 points - Read EyeX Eye Locations.

Garrett:

#548 - 8 points - Draw lines with leap motion

#546 - 8 points - Find working leap motion gestures

#554 - 8 Points - Save a File

**Sprint Planning 3**

Date: 2/14/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 7:45

End time: 8:40

After discussion, the velocity of the team were estimated to be 46 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* 558
* 590
* 591
* 592
* 548
* 546
* 554

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

Andrew :

#558 - 5 Points - Design Proper Feedback for Multitouch

#590 - 5 Points - Improve Performance

#591 - 5 Points - Design UI for Multitouch

#592 - 8 Points - Implement feedback for multitouch

Garrett:

#548 - 8 points - Draw lines with leap motion

#546 - 8 points - Find working leap motion gestures

#554 - 8 Points - Save a File

**Sprint Planning 4**

Date: 2/27/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 11:15

End time: 12:45

After discussion, the velocity of the team were estimated to be 47 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* #623 - 2 Points - Create Multitouch ReadMe
* #625 - 2 Points - Provide leapmotion read me for user
* #632 - 5 Points - Implementation of UI Menu
* #629 - 5 Points - Disable Gestures while drawing.
* #623 - 5 Points - Implement Double Tap Multitouch Gesture
* #634 - 3 Points - Enable device’s connection status
* #626 - 5 Points - Enable Alpha Coloring for Shapes
* #627 - 5 Points - Create layer visualization menu
* #628 - 3 Points - Design Device Modes
* #636 - 3 Points - Develop interactive paint device modes
* #633 - 2 Points - Implement icons for gestures
* #631 - 2 Points - Refactor “Touch” code
* #630 - 5 Points - Create Vertical Symmetry Line
* #635 - 3 Points - Work with intel camera

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

Andrew :

* #623 - 2 Points - Create Multitouch ReadMe
* #631 - 2 Points - Refactor “Touch” code
* #624 - 5 Points - Implement Double Tap Multitouch Gesture
* #626 - 5 Points - Enable Alpha Coloring for Shapes
* #627 - 5 Points - Create layer visualization menu
* #628 - 3 Points - Design Device Modes
* #630 - 5 Points - Create Vertical Symmetry Line

Garrett:

* #625 - 2 Points - Provide leapmotion read me for user
* #629 - 5 Points - Disable Gestures while drawing.
* #632 - 5 Points - Implementation of UI Menu
* #636 - 3 Points - Develop interactive paint device modes
* #634 - 3 Points - Enable device’s connection status
* #633 - 2 Points - Implement icons for gestures
* #635 - 3 Points - Work with intel camera

**Sprint Planning 5**

Date: 3/13/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 9:55

End time: 11:05

After discussion, the velocity of the team were estimated to be 49 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics) (UPDATING DURING SPRING BREAK WITH USER STORY NUMBERS)

* Implement Override Modes
* Add layer transparency
* Proximity menu for leap motion
* LibUSB Research
* Implement Time Machine
* Transparent Eraser
* Leap Motion Shapes
* Real Sense Functionality
* Libcinder has Multitouch research
* Additional Multitouch Gestures
* Transparency on saves
* Add ‘Cool’ Shader functionality.

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

Andrew :

* 5 - Add Layer Transparency #662
* 5 - Implement Override Modes #678
* 3 Add Transparency on Save #664
* 3 Make eraser erase ‘transparent’ #665
* 3 Leap Motion Shapes#666
* 3 Add MultiTouch Gestures#668
* 2 ‘Research Libcinder Has Multitouch’ #667

Garrett:

* 10 - RealSense Implementation (Will be split into smaller User stories before end spring break)
* 5 - Create Shader
* 5 - Proximity Menu for Leap Motion
* 2 - LibUSB Brainstorm
* 3 - Set up time machine functionality or “undo” functionality

**Sprint Planning 6**

Date: 4/2/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 2:55

End time: 3:30

After discussion, the velocity of the team were estimated to be 45 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* 3 - Add Realsense to Override Mode - #700
* 10 - Add additional functionality for Real Sense - #708
* 3 - Implement Additional Gestures for Real Sense - #703
* 2 - Add Chrono to Device Detection - #699
* 2 - Find Icons - # 701
* 5 - Implement Solid Background - #702
* 10 - Implement Basic Text - #698
* 3 - Integrate Real Sense into Default Mode - #707
* 2 - Integrate Real Sense into Devicehandler - #706
* 2 - Design Start up Image - #649
* 3 - Reset Default Mode on Device Disconnect - #559

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

Andrew :

* 5 - Implement Solid Background - #702
* 2 - Find Icons - # 701
* 3 - Add Realsense to Override Mode - #700
* 2 - Add Chrono to Device Detection - #699
* 10 - Implement Basic Text - #698

Garrett:

* 10 - Add additional functionality for Real Sense - #708
* 3 - Integrate Real Sense into Default Mode - #707
* 2 - Integrate Real Sense into Devicehandler - #706
* 3 - Implement Additional Gestures for Real Sense - #703
* 2 - Design Start up Image - #649
* 3 - Reset Default Mode on Device Disconnect - #559

**Sprint Planning 7**

Date: 4/17/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 11:10

End time: 11:40

After discussion, the velocity of the team were estimated to be 45 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* 3 Icons Implemented
* 2 Draw UI Improvements
* 3 Compress Brush Buttons
* 1 Add Frames Per Second Counter
* 3 Add Start Up Image
* Develop Leap Motion and Real Sense User Guide - 10
* Develop Multitouch and EyeX User Guide - 10
* Ensure Bugless Leap Motion and Real Sense Interaction  - 5
* Ensure Bugless Multitouch and EyeX Interaction - 5

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

Andrew :

* 3 Draw UI Improvements - #731
* 3 Icons Implemented - #730
* 2 Add Frames Per second counter- #733
* 10 Develop Multitouch and EyeX User Guide - #734
* 5 Ensure Bugless Multitouch and EyeX Interaction - #736

Garrett:

* 5 Compress Brush Buttons - #732
* 3 Add Start up Image - #735
* 10 Develop Leap Motion and Real Sense User Guide - #737
* 5 Ensure Bugless Leap Motion and Real Sense Interaction - #738

**Sprint Planning - For Product Backlog**

Date: 5/1/2016

Attendees: Garrett, Andrew, Professor Ortega

Start time: 12:45

End time: 1:30

After discussion, the velocity of the team were estimated to be NO points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority. (See below for specifics)

* #753 Implement Threading
* #754 Add Active Pen Support
* #755 Implement Speech Recognition
* Implement Gamified Tutorial

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

Andrew :

* N/a

Garrett:

* N/a

# 

# References

Special thanks to Alain Galvan from the OpenHID for helping us with the new Graphics Programming space we were working.

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

https://libcinder.org/docs/

http://developer.tobii.com/eyex-sdk/

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

https://developer.leapmotion.com/documentation/cpp/index.html