TouchCU

Texas Christian University
April 23, 2014

[SOFTWARE REQUIREMENTS SPECIFICATION]

Version 4.1

Table of Contents

I. Revision Signatures

By signing this document, the team member is acknowledging that he/she has read through this document thoroughly and has certified that the information within this document is accurate and satisfies all requirements.

Name	**Signature**	**Date Signed**
Trenton Bishop		
Yizhou Hu		
Blake LaFleur		
Thales Lessa		
Matthew Spector		

II. Revision History

Version	**Changes**	**Edited**
v0.1	Initial Documentation Draft	31 October 2013
v0.2	Additional Refinements	05 November 2013
v0.3	Added System Architecture + Gesture Tables	06 November 2013
v0.4	Added Settings Prototype + Updated Use Case Scenarios	07 November 2013
v1.0	First Major Draft	08 November 2013
v1.1	Additional refinements + Updated logo	18 November 2013
v1.2	Added details to the use case scenarios	19 November 2013
v1.3	Changes based on Dr. Payne’s input	03 December 2013
v2.0	Iteration 2	30 January 2014
v3.0	Iteration 3	28 February 2014
v4.0	Iteration 4	25 March 2014
v4.1	Final Draft	23 April 2014

1 Introduction

1.1 Purpose

The purpose of this document is to provide all requirements for project TouchCU. This document will cover the functional and non-functional requirements of the project. The requirements were developed while working with the project advisor, Dr. Payne, and with input from the TouchCU team.

1.2 Intended Audience

This document is for the team developing TouchCU and is designed to be a reference for what is necessary to complete the project.

1.3 Overview

This document includes the following sections.

Section 2 - Project Overview: Outlines the scope and objectives of our project while also giving a background on the application that is being developed.

Section 3 – Overall Description: Outlines the user requirements, constraints, and the operating environment.

Section 4 – System Architecture: Illustrates the hardware and accompanying connections that will be needed.

Section 5 – External Interface Requirements: Describes the user, hardware, and software interfaces that the user will interact with.

Section 6 – Functional Requirements: Defines the functions of the system.

Section 7 – Non-functional Requirements: Defines the constraints of the system.

Section 8 - Glossary of Terms: Includes a list of abbreviations and their technical terms and their associated definitions.

2 Project Overview

2.1 Scope and Objectives

The massive growth of touch technology has created an increased demand for new and innovative ways for users to interact with their devices. TouchCU, a 2013-2014 capstone project, consists of a desktop application that turns any flat surface into a multi-touchscreen utilizing the Microsoft Kinect for Windows, a standard projector, and a Windows 8 PC. Research teams at Intel Labs and Ubi-Interactive have created similar products, with one supported by Microsoft and available commercially. TouchCU, while similar in concept, allows for a greater operating range and implements voice interaction.

2.2 Project Background

In the early days of touch technology, cell phones gave up their buttons for touchscreens. As touch technology continues to improve it is simultaneously incorporated into almost everything around us. We see touch technology in everything from our refrigerators, to the media controllers in our cars. But more important than touch technology itself is the users' increased demand for new and innovative ways to interact with their devices. With that in mind, the concept of making any surface "touch-friendly" gave birth to the TouchCU project.

The Microsoft Kinect, a key component of the project, is a motion sensing device for both the Xbox 360 gaming consoles and Windows PC's. The Kinect features an RGB camera and a depth sensor that allows the capturing of body movements. When used with a projector and proper screen calibration, this allows body movements or "gestures" to be interpreted by the Windows 8 OS as input.

Ubi is a current product on the market that functions similarly to TouchCU. This product is made available by Ubi-Interactive and uses the same concept and hardware (Kinect, Projector, a flat surface, Windows 8 computer) to register touch input and gestures with the Windows 8 OS.

3 Overall Description

3.1 Product Description

TouchCU is a system designed to turn any flat surface into a touchscreen. It utilizes a standard projector, Microsoft Kinect for Windows, and a Windows 8 PC. The system consists of a standalone application that interfaces with the Microsoft Kinect for Windows. The Kinect sends the data it receives to our standalone application. Our application will send touch input information to the Windows 8 OS. This allows the user to control and interact with a Windows 8 PC on a projected surface.

3.2 User Requirements

TouchCU will allow a user to accurately interact with their Windows 8 PC by using any flat surface as a touchscreen. The application will be user-friendly, require minimal training, employ standard/well known touch gestures and require minimal set up.

3.3 User Characteristics

TouchCU is designed to be used by anyone who wants to interact with their computer on an external surface. The user will utilize the projected image of the computer's desktop to interact with the system, giving it commands as if it were a touchscreen. This interaction will encompass several touch commands as well as gestures, which will enable the user to control the computer.

3.4 Constraints

- Time Constraints:
  - Limited by the school year (May 2014).
- Kinect Limitations:
  - Maximum Kinect capture rate of 30fps.
  - Maximum/Minimum distance of 13'/2.6' from the Kinect to the screen.
  - Maximum/Minimum size of the projected image. (17.6'/3.5' diagonal screen size)
  - No objects can be in front of the screen during calibration.
- Computer Limitation:
  - Running Windows 8 or higher.

3.5 Operating Environment

TouchCU will be a software application that can be installed on any computer running Windows 8 or later. The Kinect Driver v 1.8.0 must be installed along with the .NET Framework 4.5 or higher. The computer must meet the minimum requirements for using the Microsoft Kinect, which are:

- minimum of a dual-core processor @ 2.66 GHz
- 2 GB of RAM
- dedicated USB 2.0 port

3.6 Assumptions and Dependencies

We are assuming that the user will have the following:

- A Windows 8 PC that meets the minimum requirements for using the Microsoft Kinect.
- A standard projector that is capable of projecting a resolution of 1920x1080.
- A Microsoft Kinect for Windows (Kinect for Xbox not supported).

Due to limitations of the Microsoft Kinect for Windows, fast touch input may be skewed or inaccurate and will depend on the speed of the user interacting with the system.

4 System Architecture

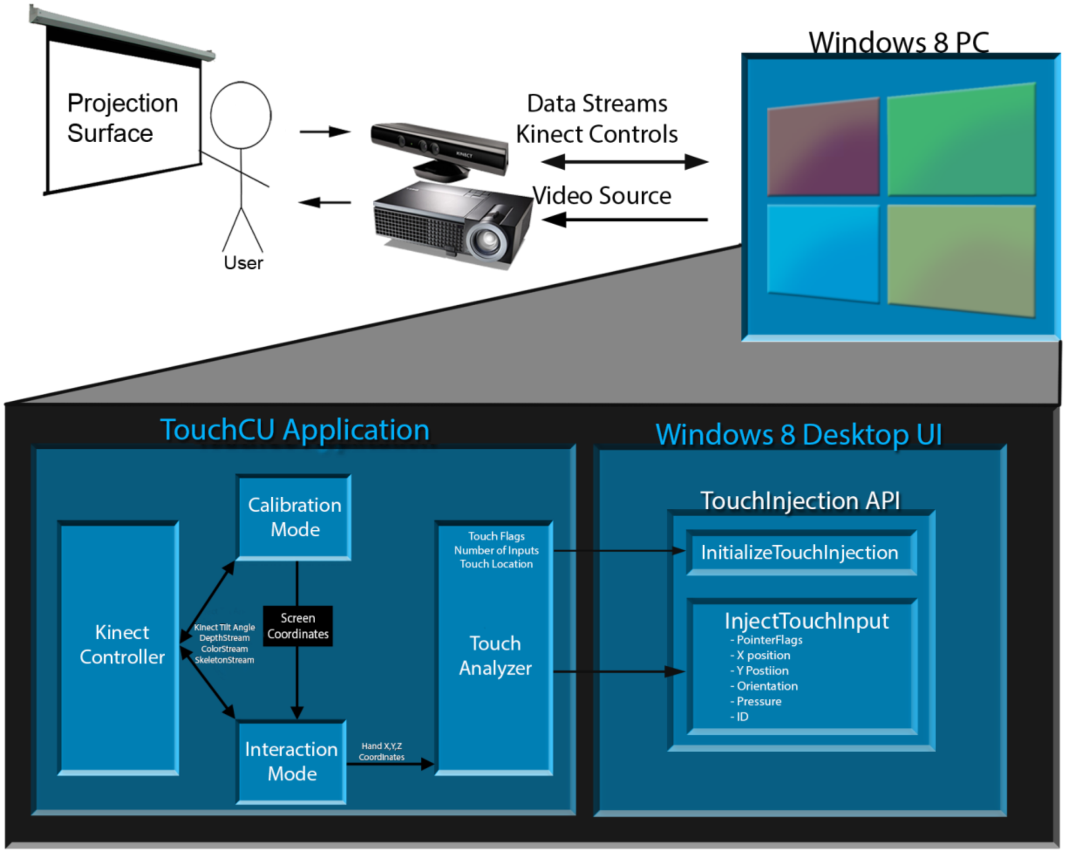

Figure 1

5 External Interface Requirements

5.1 User Interfaces

The user will interface with the TouchCU standalone application by doing the following:

1. Open the TouchCU application.
2. Follow on screen instructions to begin calibration.
3. Upon successful calibration, the user will be prompted with a success message and will minimize to the taskbar.
4. The user will now be able to interact with their system via the "touch" screen.

5.2 Hardware Interfaces

1. The application will run locally on the user's machine.
2. The application will interact with a single Kinect via USB.
3. The computer's desktop will be projected onto a flat surface via a projector. The video output to the projector can be analog or digital.

5.3 Software Interfaces

The software interface will seamlessly integrate into the Windows 8 operating system. TouchCU will have two user interfaces.

1. Allows system to be calibrated prior to use or during operation.
2. Allows user to change settings, re-calibrate, and disable features.

6 Functional Requirements

6.1 General Requirements (GEN)

GEN 1
The application shall provide for any flat, non-reflective surface, meeting size and distance constraints, to be utilized as a touchscreen.

GEN 2
The application will receive hand movement input from the Microsoft Kinect for Windows.

GEN 3
The application will provide for the recognition of hand and touch gestures.

GEN 4
The application will interpret position and movement data from the Kinect and relay touch and gesture input to the OS.

GEN 5
The application must be able to calibrate the size and placement of the screen surface.

6.2 Standalone Application Requirements (SAR)

SAR 1
The user will be able to calibrate the system upon application startup or by manually choosing the calibrate option from the settings menu.

SAR 2
The user will be able to access a settings menu in order to disable air gestures and enable voice commands.

SAR 3
The user will be able to access a debugging mode that shows hand location, XYZ coordinates of each touch point, and icon showing whether each hand is touching the screen. The user will also be able to interact with an object for testing out various gestures.

SAR 4
The application must be able to determine the hand position to allow for the proper location of the touch input.

SAR 5
The application must be able to support gestures from two hands.

6.3 Gesture Requirements (GR)

GR 1 (Tap)
The user will be able to tap the screen to simulate a left-click from a mouse.

GR 2 (Double-Tap)
The user will be able to double-tap the screen to simulate a double click from a mouse.

GR 3 (Hold)
The user will be able to tap and hold the screen to simulate a right-click from a mouse.

GR 4 (Drag)
The user will be able to tap and hold the screen while moving in any direction to simulate moving of an object on the screen.

GR 5 (Zoom)
The user will be able to use both hands and move either farther apart or closer together to simulate making an object larger or smaller on the screen.

GR 6 (Rotate)
The user will be able to use both hands to emulate a clockwise or counter-clockwise motion to move the object around a center point.

GR 7 (Swipe Left)
The user will be able to use one of their hands in mid-air to move a short distance to the left to simulate using the left arrow on the keyboard.

GR 8 (Swipe Right)
The user will be able to use one of their hands in mid-air to move a short distance to the right to simulate using the right arrow on the keyboard.

6.4 Voice Command Requirements (VCR)

VCR 1 (Addie)
The user will say **"Addie"** + *command* + *action word* aloud to initiate the voice recognition process.

VCR 2 (Open)
The user will say *Addie* + **"Open"** + *action word* aloud to open the window of the previously said action word.

VCR 3 (Close)
The user will say *Addie* + **"Close"** + *action word* aloud to close the window of the previously said action word.

VCR 4 (Start Menu)
The user will say *Addie* + *Open/Close* + **"Start Menu"** aloud to open or close the Windows Start Menu.

VCR 5 (Window)
The user will say *Addie* + *Close* + **"Window"** aloud to close the active window.

VCR 6 (My Documents)
The user will say *Addie* + *Open* + **"My Documents"** aloud to open the user's Documents.

VCR 7 (Settings)
The user will say *Addie* + *Open* + **"Settings"** aloud to open the TouchCU settings menu.

VCR 8 (Debug)
The user will say *Addie* + *Open/Close* + **"Debug"** aloud to open or close the TouchCU debugging overlay.

7 Non-Functional Requirements

7.1 Performance Requirements (PR)

PR 1

The application must be able to convert Kinect output data into Windows touch input data within 500 milliseconds.

PR 2

The application must have a touch accuracy of within three centimeters.

PR 3

The application must be able to run stably for as long as the computer is powered on.

8 Glossary of Terms

- **Audio Stream –** Data stream from the Kinect that allows for high-quality audio capture that can be used for voice commands.
- **Color Stream –** Data stream from the Kinect that acts as an RGB web camera.
- **Depth Stream –** Data stream from the Kinect that creates a depth map from the distance of the sensor to the object.
- **EOD** - End of Day.
- **HDMI -** High-Definition Multimedia Interface.
- **IDE -** Integrated Development Environment.
- **Kinect -** A motion sensing input device by Microsoft for the Xbox 360 video game console and Windows PCs. Based around a webcam-style add-on peripheral, it enables users to control and interact with PC through a natural user interface using gestures and spoken commands. The Kinect recognizes 20 joints on the human body at a capture rate of 30 Hz.
- **Kinect Studio -** Debugging tool for the Kinect sensor that allows the depth and color streams to be viewed in real time.
- **NTASC -** North Texas Area Student Conference.
- **Rich Interaction –** Supports multi-touch.
- **SDK -** Software Development Kit.
- **Skeletal Stream –** Data stream from the Kinect that tracks up to 20 joints on the human body.
- **SRS -** Student Research Symposium.
- **TBD -** To Be Determined.
- **TCU -** Texas Christian University.
- **TouchCU -** The team and project name for the 2013-2014 senior design team.
- **USB -** Universal Serial Bus.

9 Appendix

9.1 Appendix A: Gesture Tables

Single Hand Gestures

Name of Gesture	How it's Performed	What it's Used For
Tap (GR 1)	Tap an item on the screen once.	Simulates a left-click from a mouse.
Double-Tap (GR 2)	Tap an item on the screen twice.	Simulates a double left-click from a mouse.
Hold (GR 3)	Tap an item on the screen and hold.	Simulates a right-click from a mouse.
Drag (GR 4)	Tap and hold the screen while moving in any direction.	Simulates moving an object on the screen.

Two Hand Gestures

Name of Gesture	How it's Performed	What it's Used For
Zoom (GR 5)	Both hands will be placed on the screen and move either farther or closer apart.	Simulates making an object larger or smaller on the screen.
Rotate (GR 6)	Both hands will be placed on the screen to emulate a clockwise or counter-clockwise motion.	Simulate moving the object around a center point.

Air Gestures

Name of Gesture	How it's Performed	What it's Used For
Swipe Left (GR 7)	One hand in mid-air will move a short distance to the left.	Simulates using the left arrow on the keyboard.
Swipe Right (GR 8)	One hand in mid-air will move a short distance to the right.	Simulates using the right arrow on the keyboard.

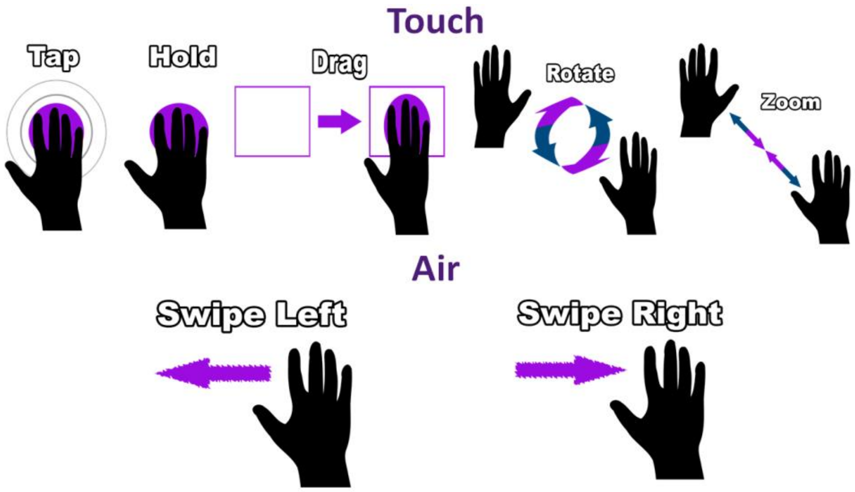

Figure 2

9.2 Appendix B: Voice Command Tables

Trigger Word		
Name of Command	How it's Performed	What it's Used For
Addie (VCR 1) [**ad**-ee]	User will say "Addie" aloud followed by a command word + action word.	Initiates the voice recognition process.

Command Words		
Name of Command	How it's Performed	What it's Used For
Open (VCR 2) [**oh**-*puh* n]	User will say "Open" aloud followed by an action word.	Used to open the following action word.
Close (VCR 3) [klohz]	User will say "Close" aloud followed by an action word.	Used to close the following action word.

Action Words		
Name of Command	How it's Performed	What it's Used For
Start Menu (VCR 4) [stahrt **men**-yoo]	User will say "Open/Close Start Menu" aloud.	Opens or closes the Windows Start Menu.
Window (VCR 5) [**win**-doh]	User will say "Close Window" aloud.	Closes the active window.
My Documents (VCR 6) [mahy **dok**-*yuh-muh* nts]	User will say "Open/Close My Documents" aloud.	Opens the user's Documents.
Settings (VCR 7) [**set**-ings]	User will say "Open/Close Settings" aloud.	Opens the TouchCU settings menu.
Debug (VCR 8) [dee-**buhg**]	User will say "Open/Close Debug" aloud.	Opens or closes the TouchCU debugging overlay.

9.3 Appendix C: Use Case Diagram

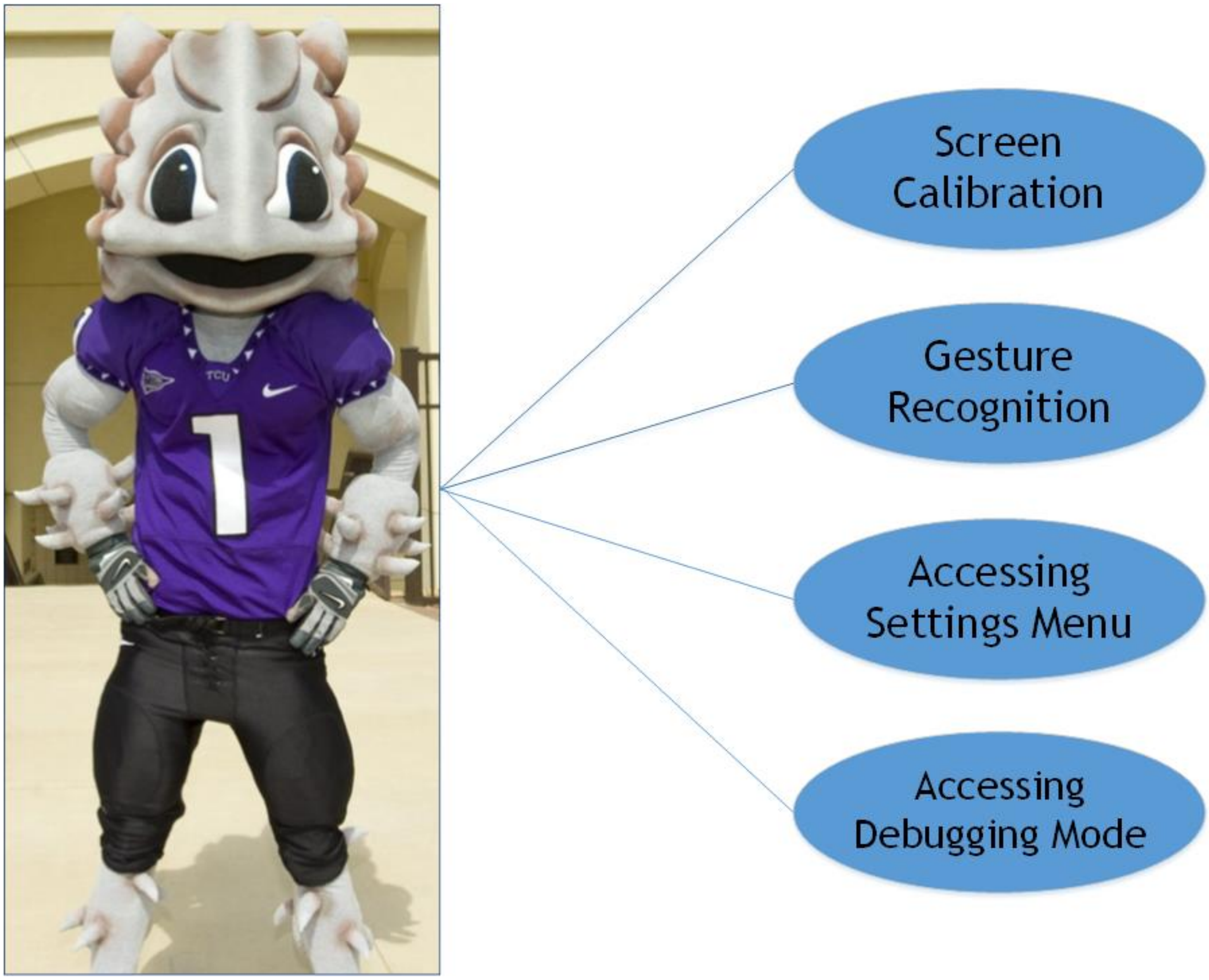

Figure 3

9.4 Appendix D: Use Case Scenarios

Screen Calibration	
Actors	User
General Goal	Determine the size and position of the screen on the surface that is being used and calculate proper screen coordinates.
Pre-conditions	• TouchCU application must be running. • One Microsoft Kinect for Windows is connected to the computer via USB and facing the surface that will be used. • Windows Desktop is projected onto the screen surface.
Triggers	The application is started on a computer for the first time or the user has decided to re-run the calibration wizard.
Course of Events	1. Manual Calibration (Multi-point Calculation Method) • The application will prompt the user to place their mouse pointer on each of the four screen corners. • Takes the coordinates and saves the data in a local file for future use. 2. The application will minimize and will begin to take input from the Kinect.
Alternate Paths	Application will use existing calibration settings.
Post-Conditions	Application moves onto Monitoring State.

Gesture Recognition (Monitoring State)	
Actors	User
General Goal	Recognize user movement and determine which gesture was performed.
Pre-conditions	Application is in the Monitoring State, successful post-calibration.
Triggers	The data stream input to the Microsoft Kinect for Windows contains movement from either one of the hands being tracked.
Course of Events	1. Application determines if a gesture was received. 2. Application moves into Active State. 3. Application generates the appropriate OS input command based on gesture. 4. Application sends input command to the OS.
Alternate Paths	If the hand does not generate a recognizable gesture, nothing will be done and application will go back to the Monitoring State.
Post-conditions	When one gesture is successfully processed and sent to the OS, the application will go back to the Monitoring State.

Accessing Settings Menu	
Actors	User
General Goal	Provide a screen within the application that shows the user a menu where they can select certain settings to change how the application runs.
Pre-conditions	System is up and running. Application is currently running and in either the Listening or Monitoring State.
Triggers	User will right-click the TouchCU icon from the System Tray and select settings or the user will say “*Addie* + *Open* + *Settings*”.
Course of Events	The settings frame appear with the following checkboxes: • Disable Air Gestures • Disable Voice Commands • Enable Advanced Settings The settings frame also allows the user to set the following smoothing parameters: • Smoothing • Correction • Prediction • Jitter Radius • Max Deviation Radius • Distance to consider a touch • Hand Extension
Alternate Paths	The settings menu will not be present unless the icon is clicked on in the System Tray or the user says “*Addie* + *Open* + *Settings*”.
Post-conditions	Application will be running regularly.

Accessing Debugging Mode	
Actors	User
General Goal	Provide a screen within the application that shows the user detailed information on the input streams the Kinect is receiving.
Pre-conditions	System is up and running. Application is currently running and in either the Listening or Monitoring State.
Triggers	User will select the option from the settings menu or the user will say “*Addie* + *Open/Close* + Debug”.
Course of Events	The debug frame appear with the following: • Point Tracker ○ Screen overlay showing current touch data and location ○ Provides feedback on user touch interactions ○ Supports up to two touch points • Manipulation ○ Supports testing of single and two hand gestures ○ Provides feedback on slide, rotate, and transform • Close Debug ○ Closes the debugging mode
Alternate Paths	The debugging screen will not be present unless the option is selected from the settings menu or the user says “*Addie* + *Open/Close* + Debug”.
Post-conditions	Application will be running regularly.