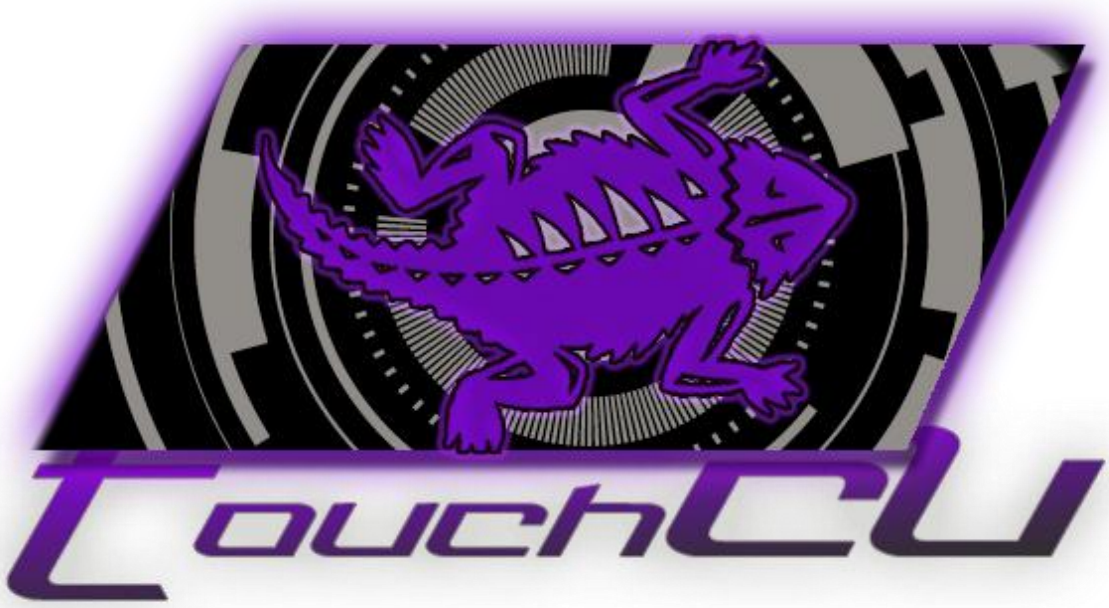
TouchCU

Texas Christian University
April 23, 2014



[DESIGN DOCUMENT]

Version 4.1
©2013-2014 Computer Science Department, Texas Christian University

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	12 November 2013
v0.2	Additional refinements + Updated Logo	18 November 2013
v0.3	Added details to the use case scenarios	19 November 2013
v0.4	Added diagrams	25 November 2013
v0.5	Added Sequence Diagrams + Additional Tweaks	03 December 2014
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 detail the design of project TouchCU. Various diagrams, tables, and pictures are included to help create a clear understanding of the system. This document will also cover the interactions that take place with the system.

1.2 Project Overview

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.

1.3 Overview

This document includes the following sections.

Section 2 – Design Constraints: Constraints of the system and the development environment.

Section 3 – System Architecture: Overview of the hardware and software architecture.

Section 4 – UML Models: A look at the models created for project TouchCU.

Section 5 – User Interface: The GUI that will be used by the user.

Section 6 - Glossary of Terms: Includes a list of abbreviations and their technical terms and their associated definitions.

2 Design Constraints

2.1 Assumptions and Dependencies

We are assuming that the user will have the following:

- A computer 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 (Xbox Kinect 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.

2.2 General Constraints

- Kinect Limitations:
  - Maximum Kinect capture rate of 30fps.
  - Maximum distance from the Kinect to the screen.
  - Maximum/Minimum size of the projected image.
  - No objects can be in front of the screen during calibration.
- Computer Limitation:
  - Must be running Windows 8 or higher.

2.3 Development Methods

TouchCU will be developed using agile methodologies in order to meet the requirements set forth by Dr. Payne and our group. The team will be developing our application for the Windows platform.

Programming Environment

- Microsoft Windows 8 x64 Professional
- Visual Studio Pro 2012
- Kinect Studio for Windows v1.8.0
- Kinect Developer Toolkit SDK

Support Environment

- Core FTP Lite 1.3c
- Subversion
- Tortoise SVN 1.8.2
- GroupMe
- Doodle

Other Software

- Adobe Photoshop CS 4
- Microsoft Office 2010
- Google Drive
- Camtasia 8
- iMovie
- Handbrake Video Converter 0.9.9

3 System Architecture

3.1 Diagram

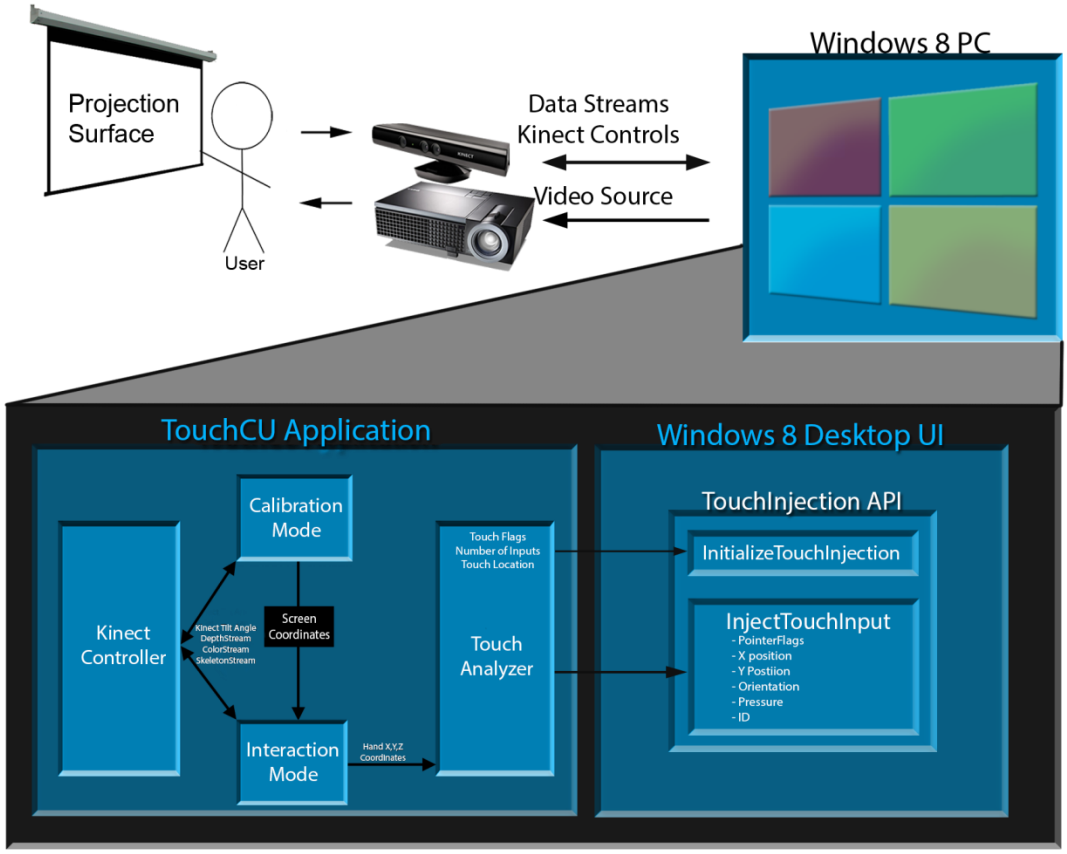


Figure 1

3.2 Projection Surface

TouchCU will be compatible with any flat, non-reflective surface that can properly display a projected image. The projection surface must be no closer than 7 feet and no further than 11 feet away from the Kinect.

3.3 Kinect

TouchCU will be compatible with the Microsoft Kinect for Windows. The Xbox Kinect is not supported.

3.4 Projector

TouchCU is compatible with any projector. A resolution of 1920x1080 is recommended for best performance, if 1920x1080 is not possible, choose a resolution as close to this as possible. The projector can use either analog or digital input as long as it supports the required resolution.

3.5 Computer

TouchCU will be compatible any computer running Windows 8 or higher (This includes 32-bit and 64-bit versions of the OS). 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 Deriving Touch Injection Parameters

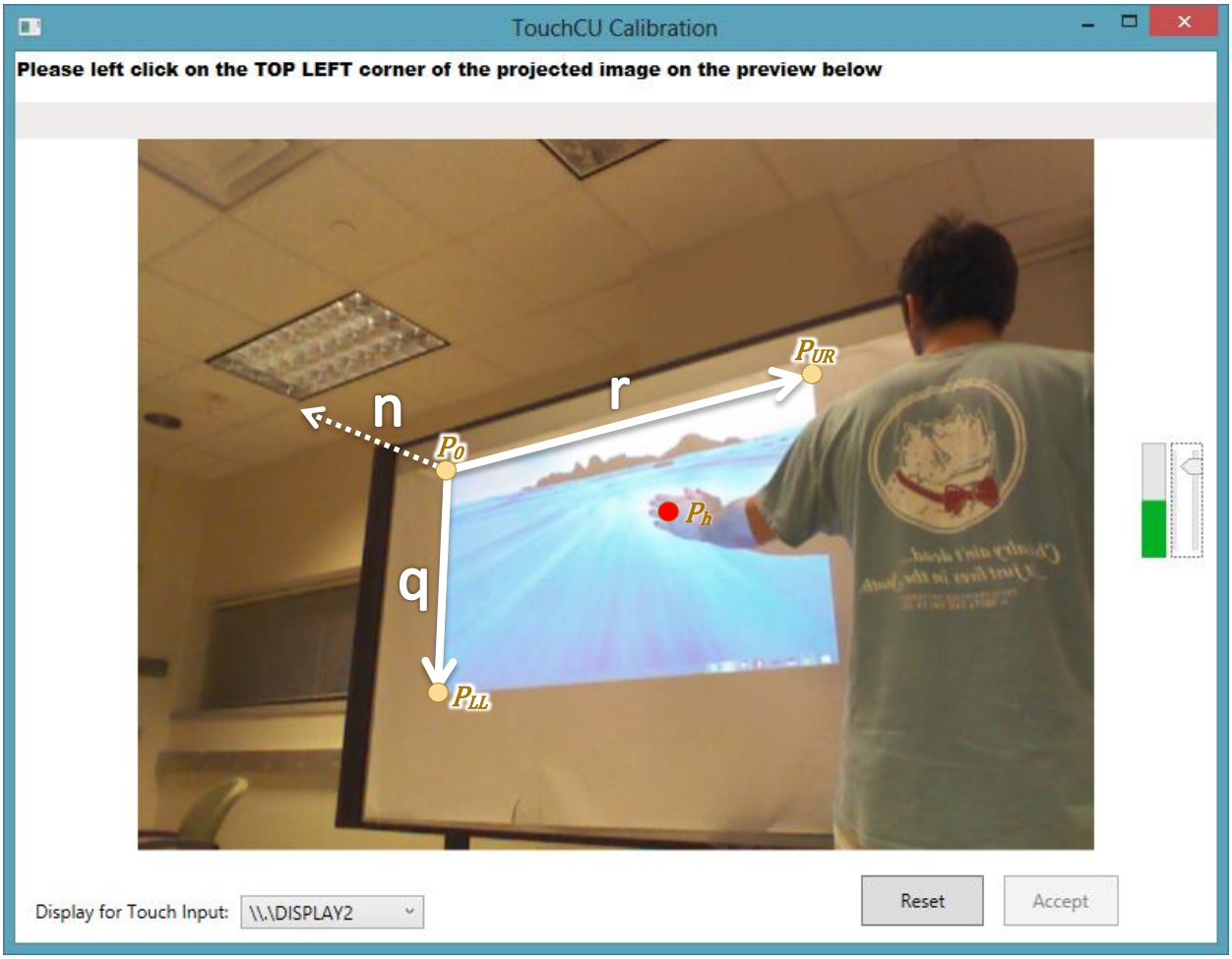


Figure 2

Every point in the space that the Kinect can see is represented by three Cartesian coordinates (x,y,z). To identify the "touch" surface, these Cartesian coordinates are used to represent a vector from the Kinect's origin (it's camera/depth detector) to that particular point as $\vec{u}, \vec{u} \in \mathbb{R}^3$.

Note: Figure 2 displays the view as seen from the Kinect, please note this is a mirrored image.

The upper left corner P_0 is used as a base point. Then, two vectors: $\vec{r}$ from the upper left corner to the upper right corner and $\vec{q}$ from the upper are calculated. $\vec{r}$ and $\vec{q}$ determines the plane the projected image is in. For any pixel in the image, it can be represented as$(a\vec{r} + b\vec{q})$ from P_0 , with $a, b \in \mathbb{R}, 0 \leq a, b \leq 1$.

$$\vec{r} = \overrightarrow{P_{UR}} - \overrightarrow{P_0}$$

$$\vec{q} = \overrightarrow{P_{LL}} - \overrightarrow{P_0}$$

The cross product, $\vec{n}$, of $\vec{r}$ and $\vec{q}$ is then taken. $\vec{n}$ points vertically out from the plane.

$$\vec{n} = \frac{\vec{r} \times \vec{q}}{|\vec{r} \times \vec{q}|}$$

For any hand position in the 3D space, the difference between the input point and the UL corner point can be represented by a linear combination of $\vec{r}$, $\vec{q}$ and $\vec{n}$.

$$\overrightarrow{P_h} - \overrightarrow{P_0} = x\vec{r} + y\vec{q} + z\vec{n}$$

The coefficients x, y and z are determined by this formula:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = (\vec{r} \quad \vec{q} \quad \vec{n})^{-1}\left(\overrightarrow{P_h} - \overrightarrow{P_0}\right)$$

- Screen Injection $(x_0, y_0) = (width \cdot x\,, height \cdot y)$
- Distance from hand to screen $d = z$

4 UML Models

4.1 State Diagram

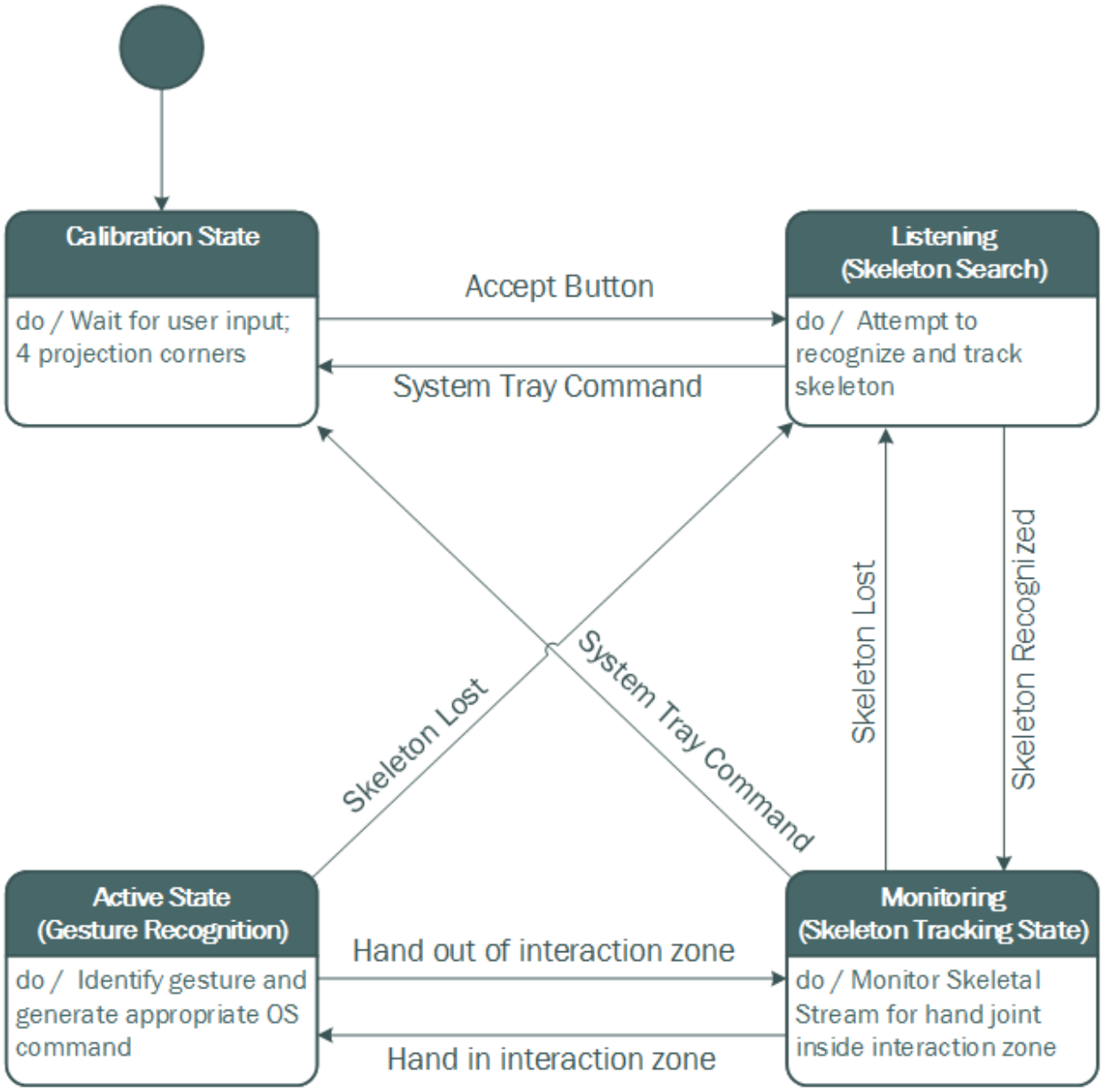


Figure 3

4.2 Class Diagrams

4.2.1 Display

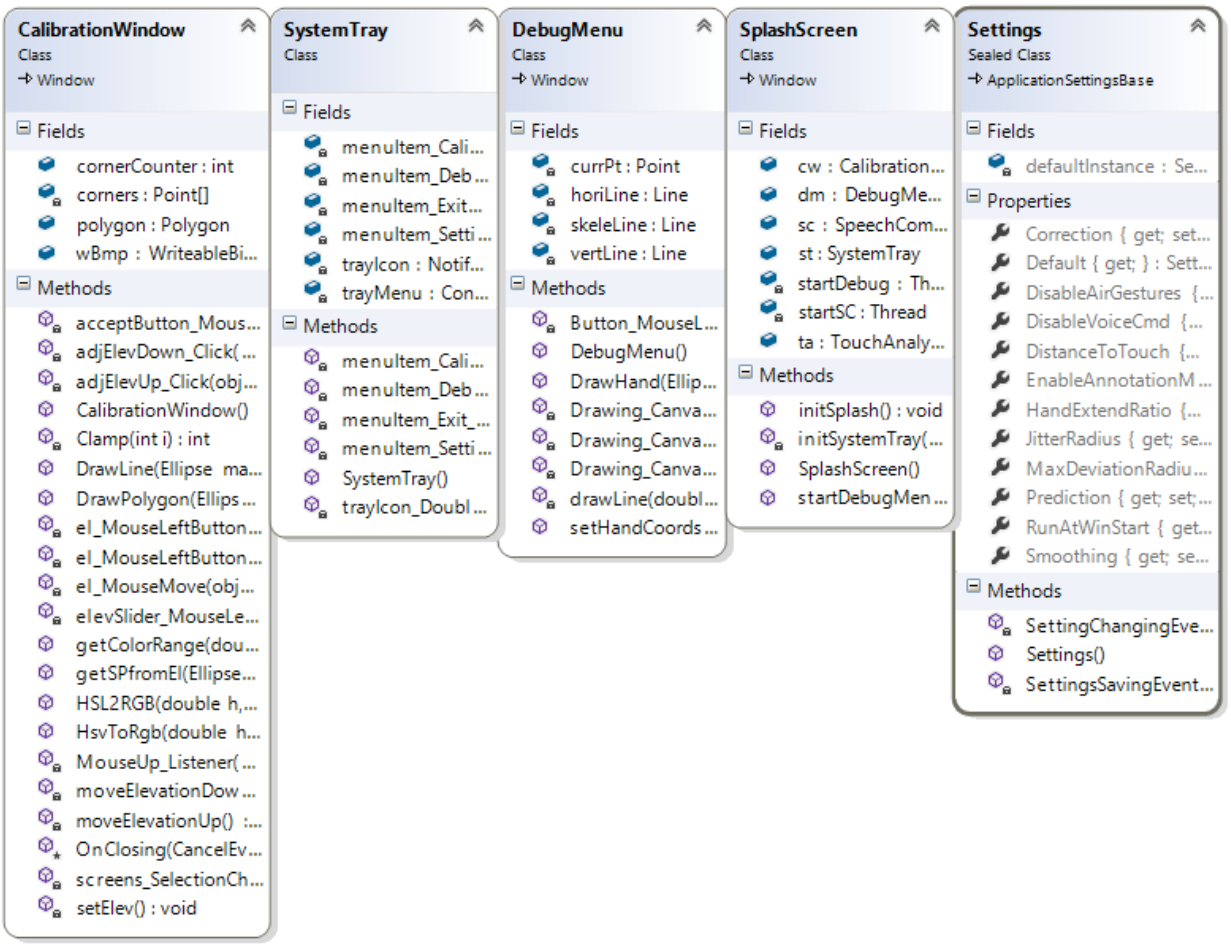


Figure 4

4.2.2 Speech

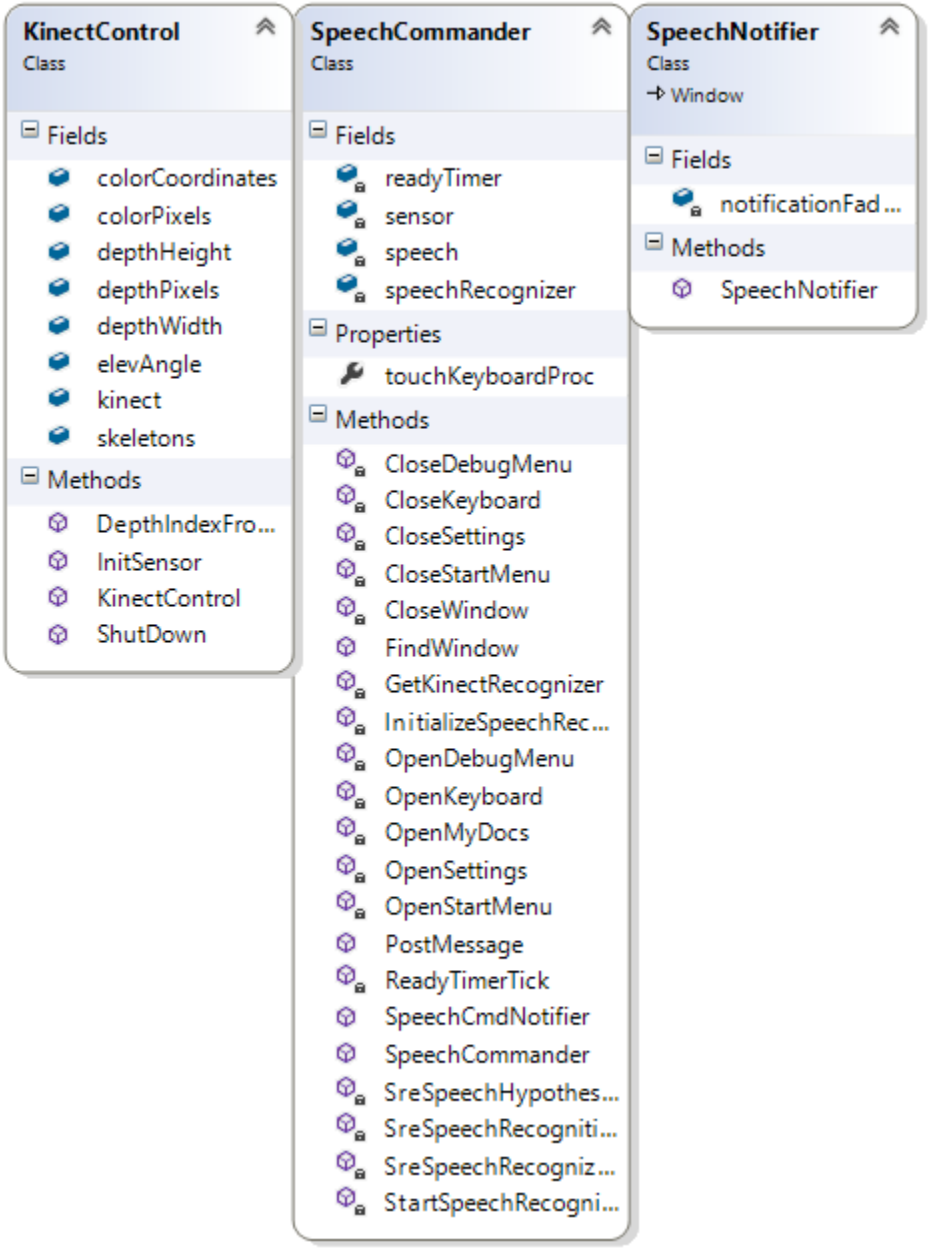


Figure 5

4.2.3 Touch

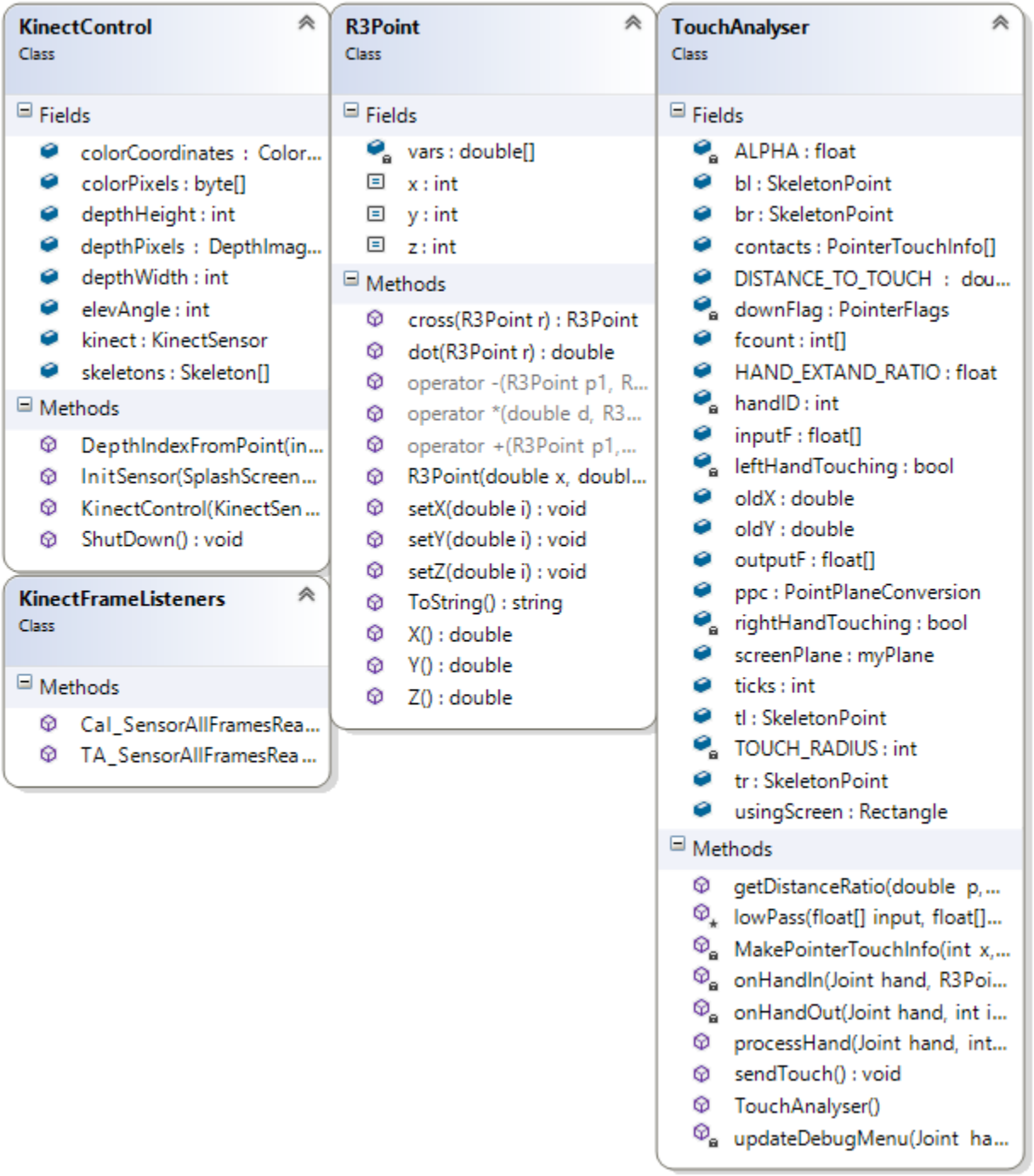


Figure 6

4.3 Sequence Diagrams

4.3.1 Screen Calibration

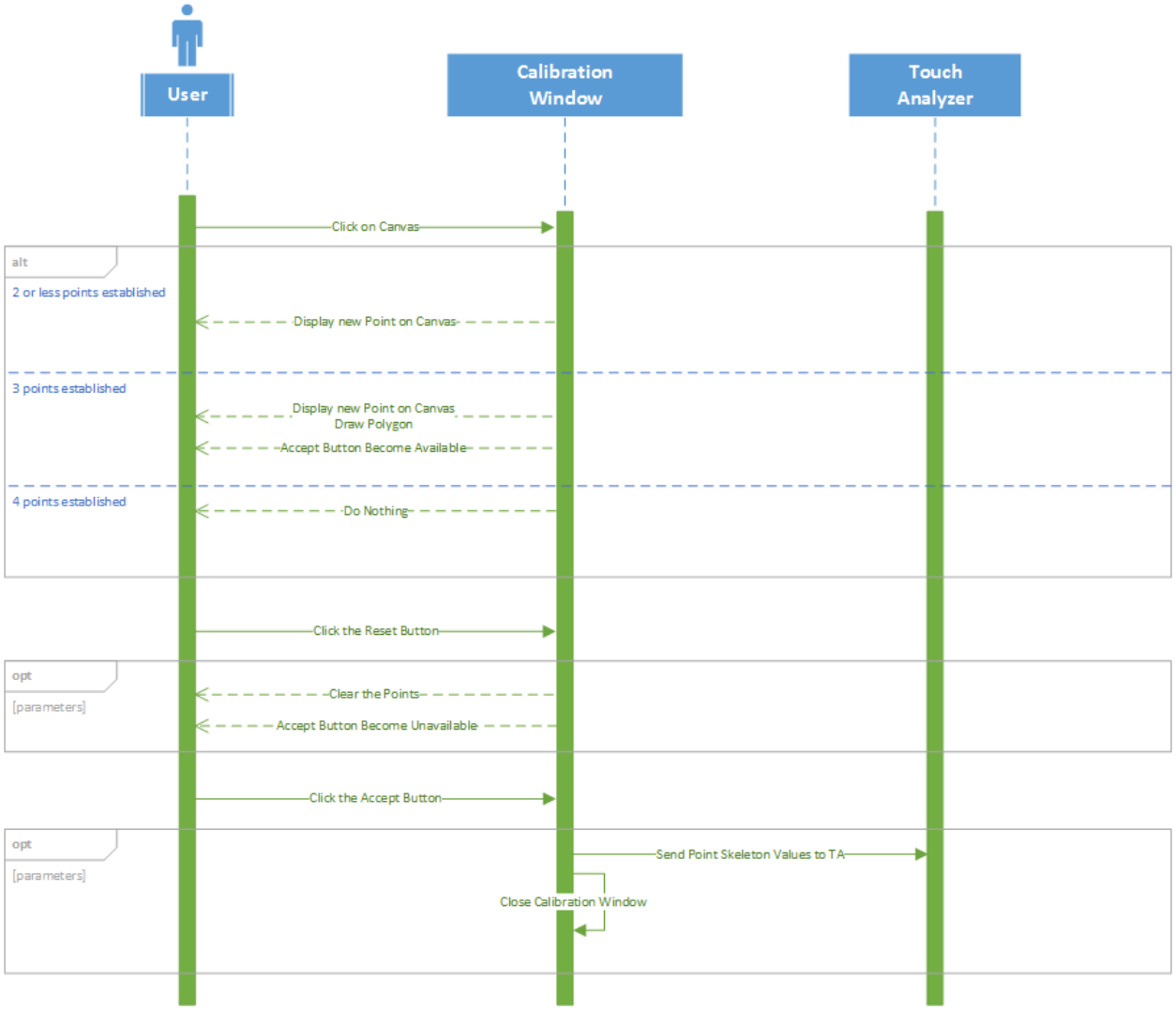


Figure 7

4.3.2 Air Gesture Recognition

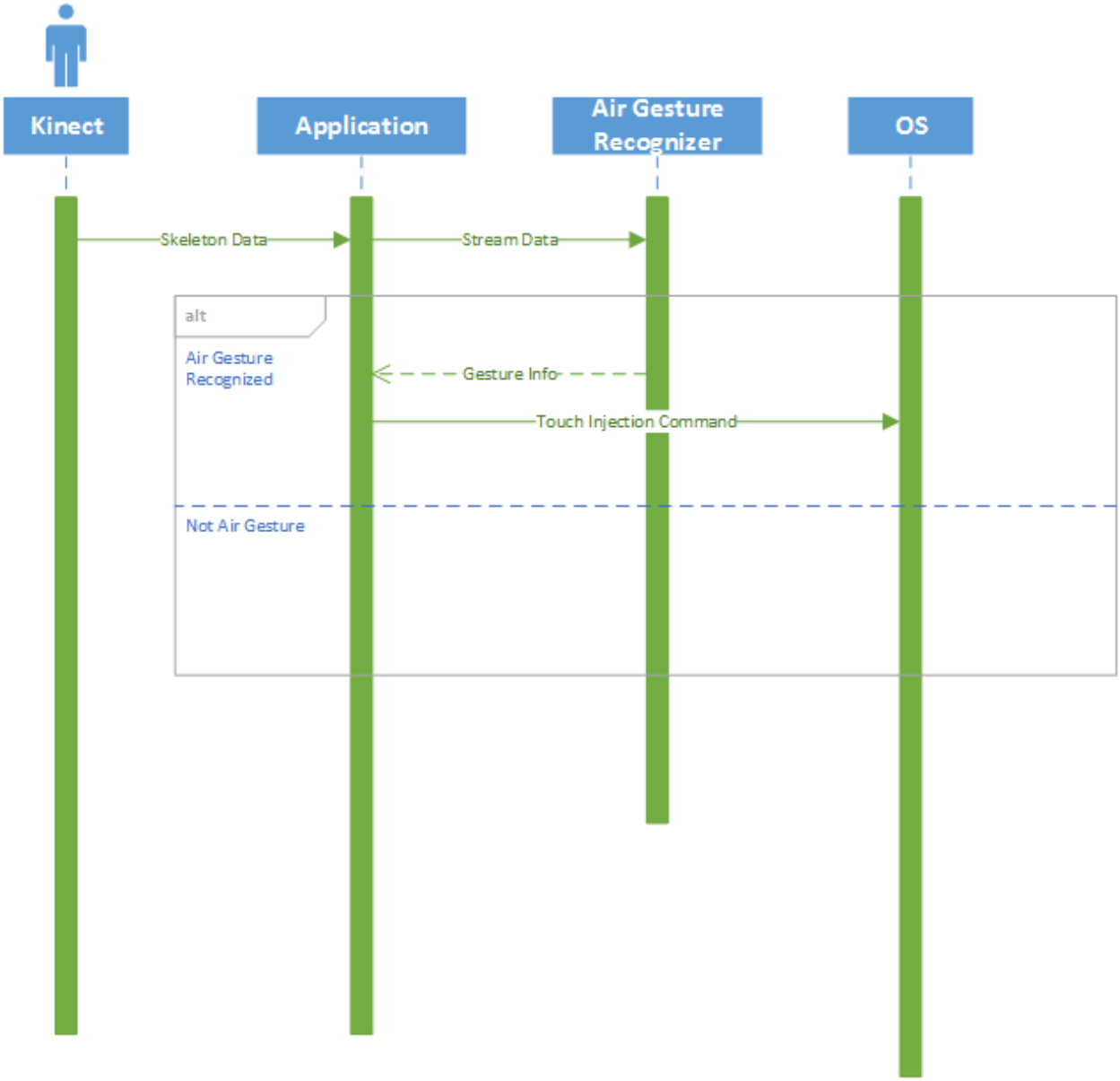


Figure 8

4.3.3 Accessing Settings Menu

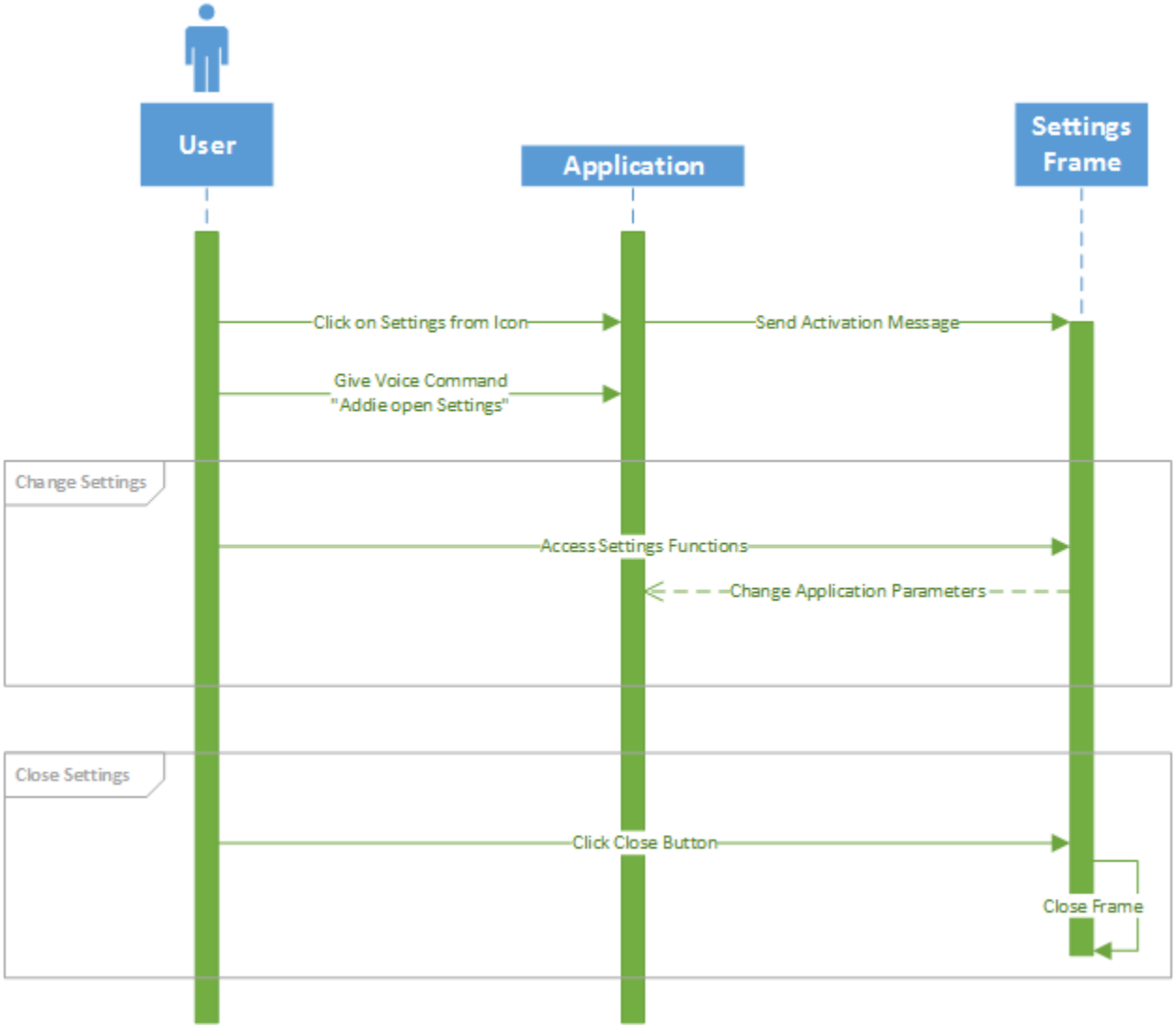


Figure 9

4.3.4 Accessing Debugging Mode

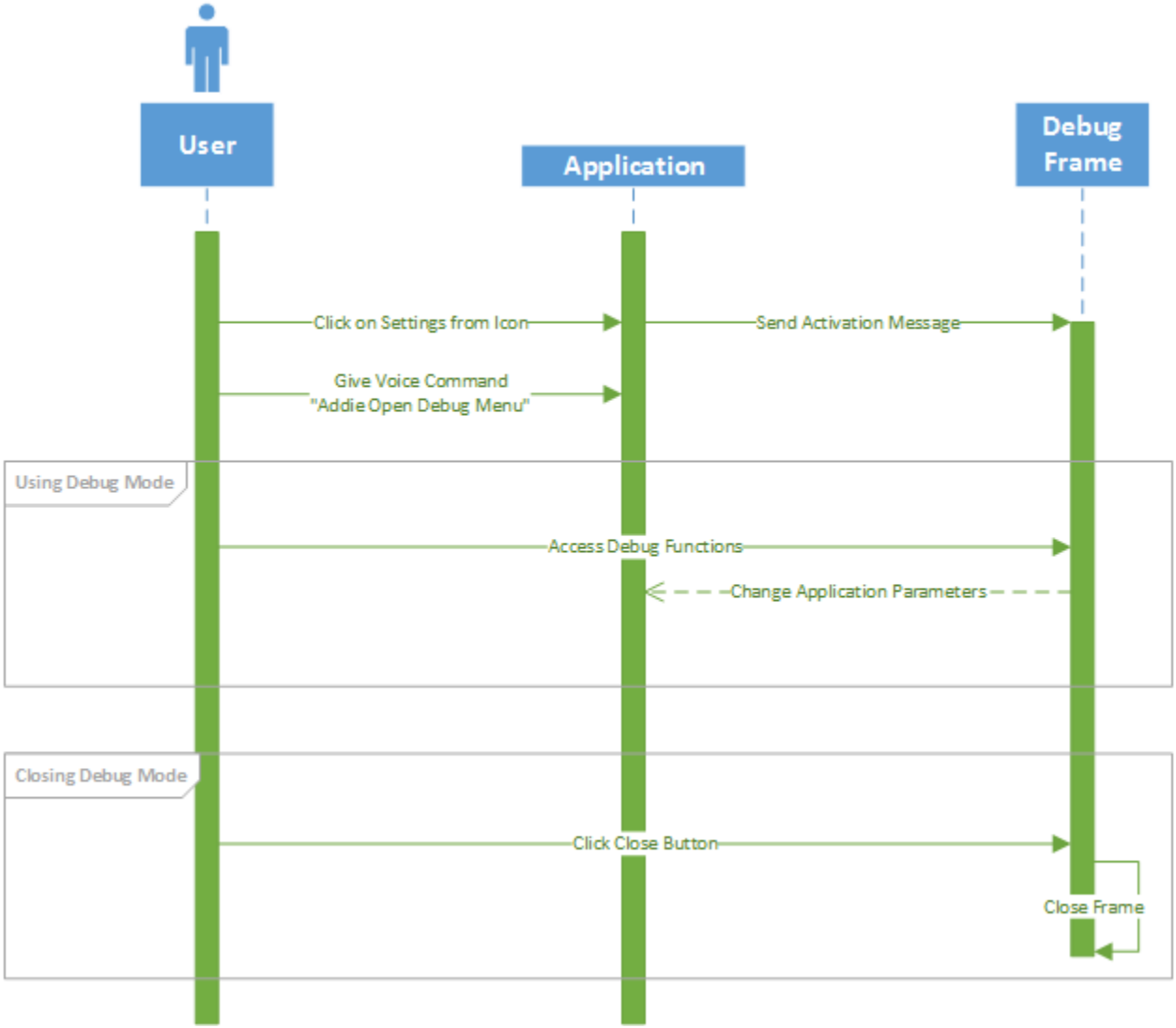


Figure 10

4.3.5 Voice Command Recognition

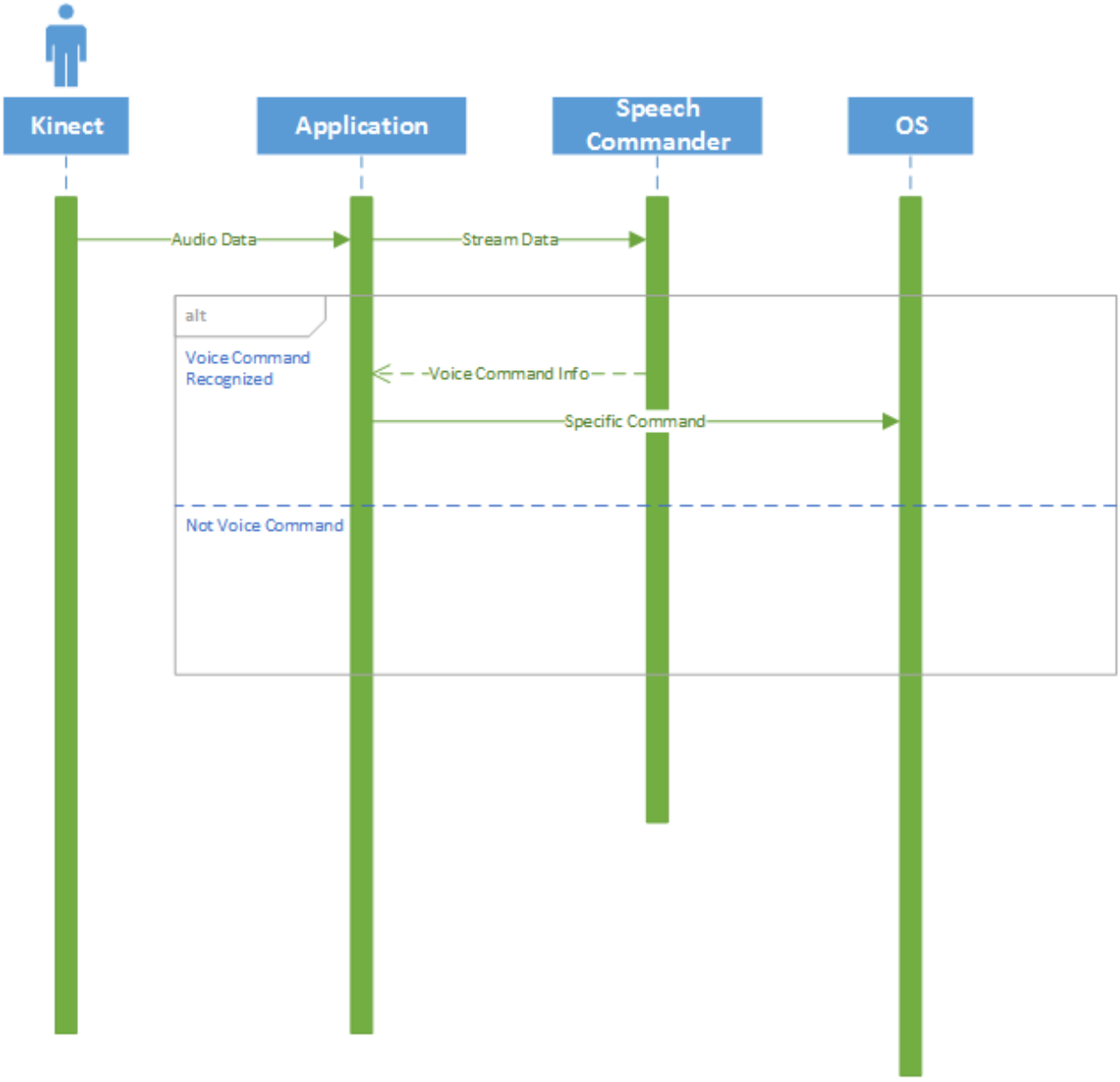


Figure 11

5 User Interface

5.1 Overview

The user will interface with the TouchCU desktop program by doing the following:

1. Open the TouchCU application.

Figure 12

2. Once the program has opened successfully, the user will receive a welcome message instructing them to begin calibration.

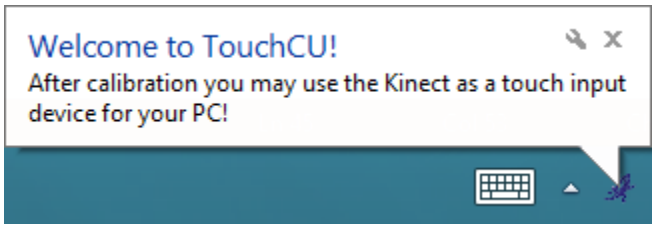


Figure 13

3. Follow on screen instructions to begin calibration

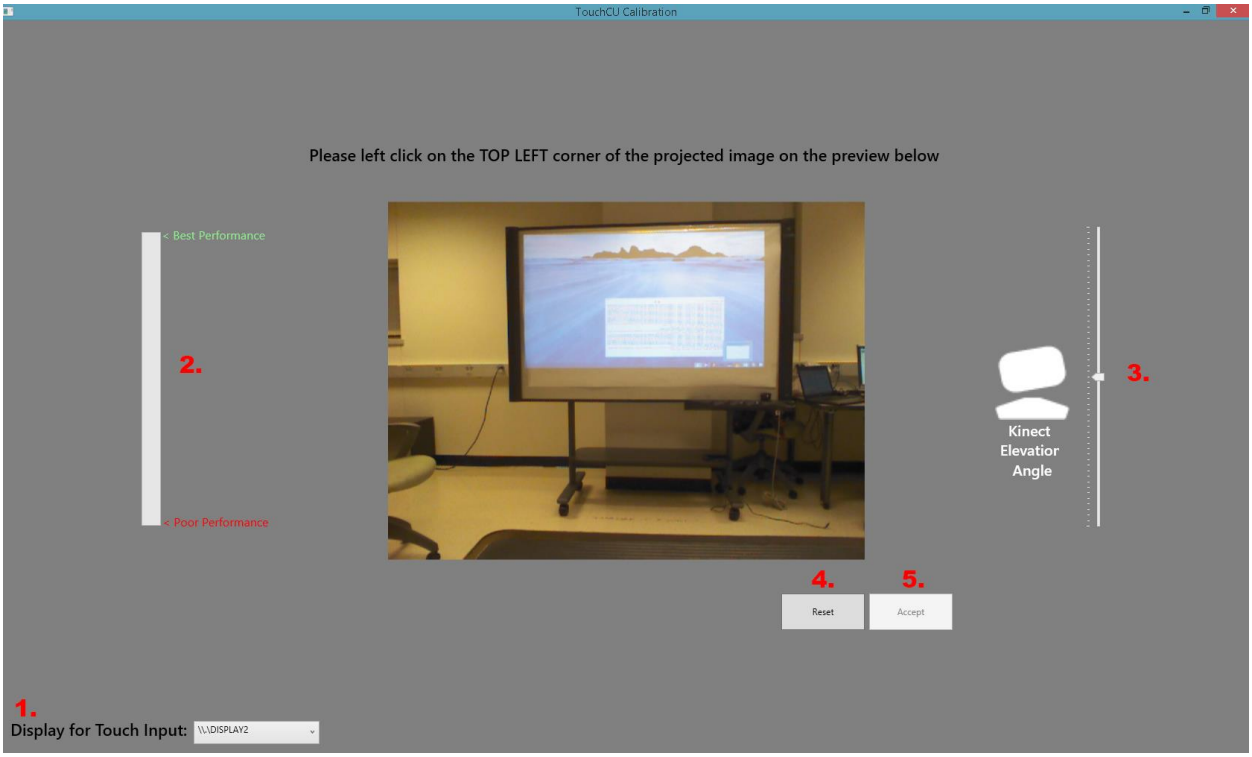


Figure 14

#	Name	Function
1	Current Display	Allows the user to set the active monitor for touch input.
2	Optimal Distance Progress Analyzer	Displays a progress bar that shows the optimal Kinect placement in respect to the projection surface.
3	Current Elevation Angle	Displays the current elevation angle of the Kinect.
4	Reset	Clears the current calibration screen and allows the user to re-calibrate.
5	Accept	Saves the current screen calibration.

4. Upon successful calibration, the program will minimize to the taskbar.

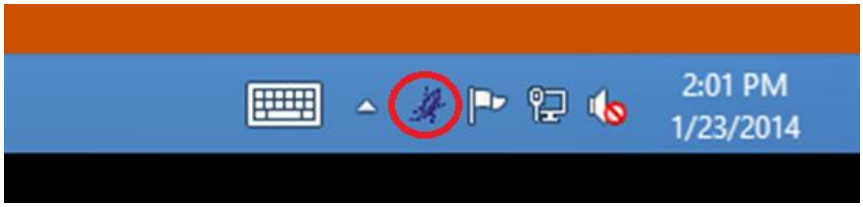


Figure 15

5. The user will now be able to interact with their system via the “touch” screen.
    a. The user now has the option to right-click on the TouchCU icon in their taskbar to access some additional options.

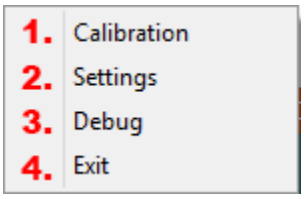


Figure 16

#	Name	Function
1	Calibration	Opens the calibration wizard for re-calibration.
2	Settings	Opens the TouchCU settings menu.
3	Debug	Enables a debugging overlay.
4	Exit	Exits the program.

    b. If the user clicks on Settings, they will see the following options.

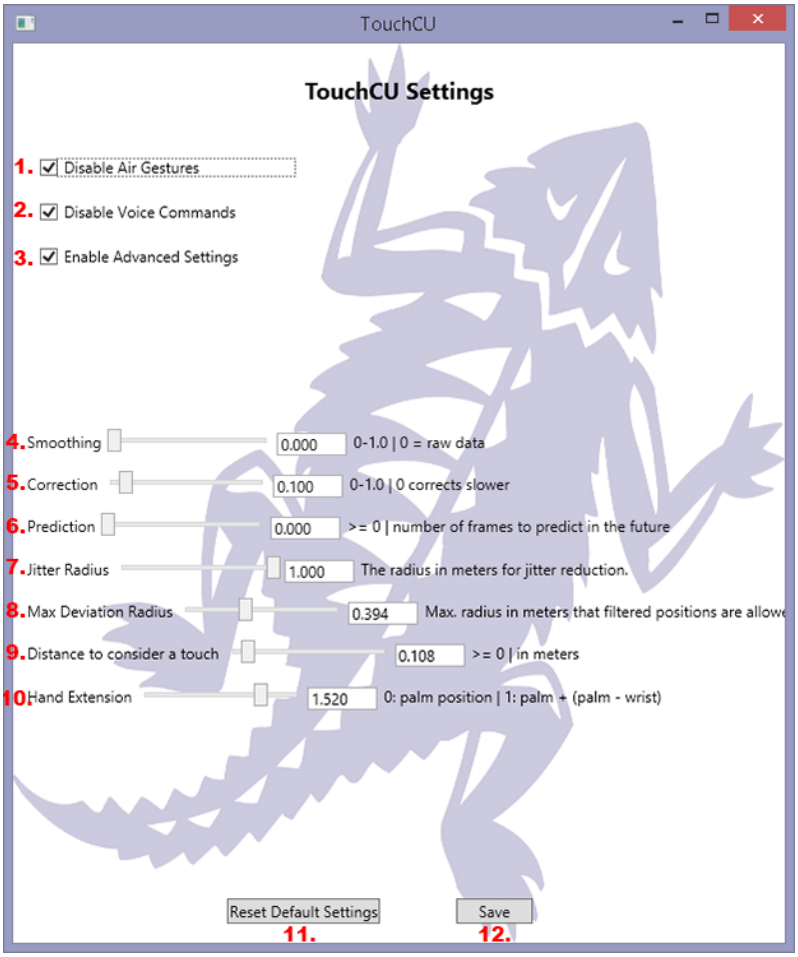


Figure 17

#	Name	Function
1	Disable Air Gestures	Disables all air gestures.
2	Disable Voice Commands	Disables all voice commands.
3	Enable Advanced Settings	Allows the user to change the advanced settings below.
4	Smoothing	Sets the amount of smoothing applied to the raw data. A value of zero does not alter the raw skeleton data. • A higher value ➔ more smoothing (Increased latency). • A lower value ➔ less smoothing (Lower latency).
5	Correction	Sets the amount of correction applied to the raw data. • A higher value ➔ Corrects towards raw data quicker. • A lower value ➔ Corrects slower and appears smoother.
6	Prediction	Sets the number of frames to be predicted.
7	Jitter Radius	Determines how aggressively to remove jitter from raw data. If a joint is tracked outside the radius, it is corrected and clamped to the radius. Measured in meters.
8	Max Deviation Radius	Sets maximum radius that filtered data can deviate from raw data. Used in conjunction with Jitter Radius to set the outer bounds of the jitter radius. Measured in meters.
9	Distance to consider a touch	Changes the depth of the touch interaction zone in reference to the touch screen surface. Starting from 0, where 0 is at the projection surface, this value increases the size (or thickness) of the users interaction zone. Measured in meters.
10	Hand Extension	This multiplier value is used to extend the touch point from the center of the palm outwards towards the fingertips. Specifically, the hand extension value (scalar) is multiplied against the wrist-to-palm distance to determine how far out to extend the touch point. For example: wrist-to-palm distance **X** hand extension = distance to extend touch point from palm center.
11	Reset Default Settings	Resets all settings back to their default options.
12	Save	Saves your current settings and closes the settings menu.

5.2 Debug Mode

The Debug part of TouchCU can be started by right-clicking the TouchCU system tray icon, and selecting Debug. It can also be started by using the appropriate voice command. The Debug application allows you to test and analyze your touch interaction while using TouchCU. Using this interaction feedback, you can adjust TouchCU's settings to enhance the accuracy and performance of TouchCU. The position of each hand is tracked in a 3D space using X, Y, and Z coordinates. The positive x-axis extends to the left, the positive y-axis extends upward, and the positive z-access extends outward in the direction that the Kinect is facing. X and Y are pixel coordinates and Z is meters from the projection surface.

Debug mode is composed of two windows. The debug output window, located at the top of the screen, and the interaction window. The debug output window displays the details of TouchCU's injected pointer information right alongside the Windows Pointer message.

The list menu populates touch interaction events that the application has received through the Windows message loop. These events include TouchDown, TouchUp, Hold, & Drag. The debug output window displays manipulation data upon completion of the appropriate manipulation.

Point tracker – Used to show current touch data and location and provides feedback on user touch interactions; up to two touch points.

Figure 18

Manipulation – Used to test single and two hand gestures while providing feedback on touch manipulations such as slide, rotate, and transform.

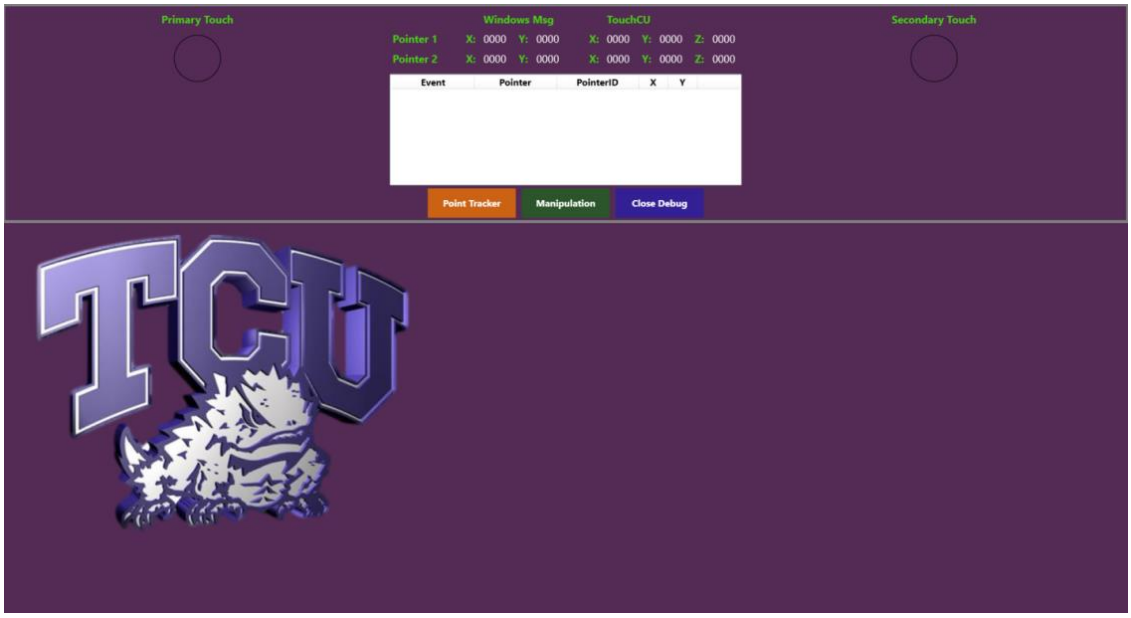


Figure 19

6 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.

7 Appendix

7.1 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.

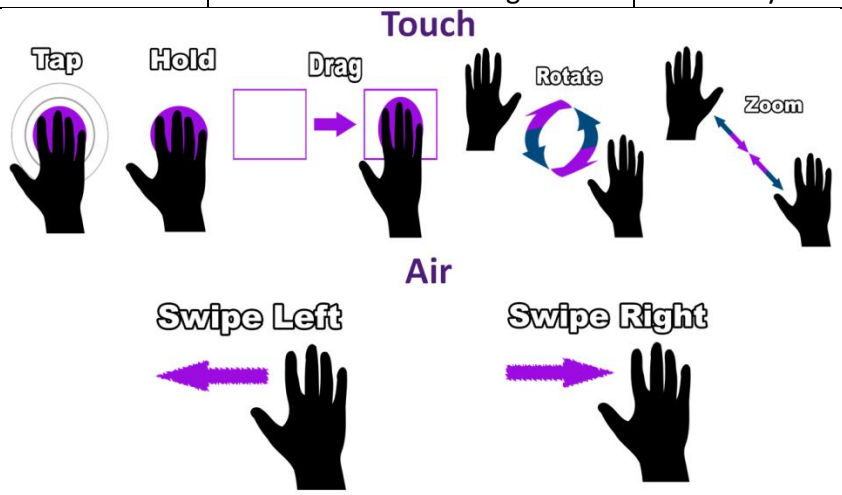


Figure 18

7.2 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 My Documents" aloud.	Opens the user's Documents.
Settings (VCR 7) [**set**-ings]	User will say "Open 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.

7.3 Appendix C: Use Case Diagram

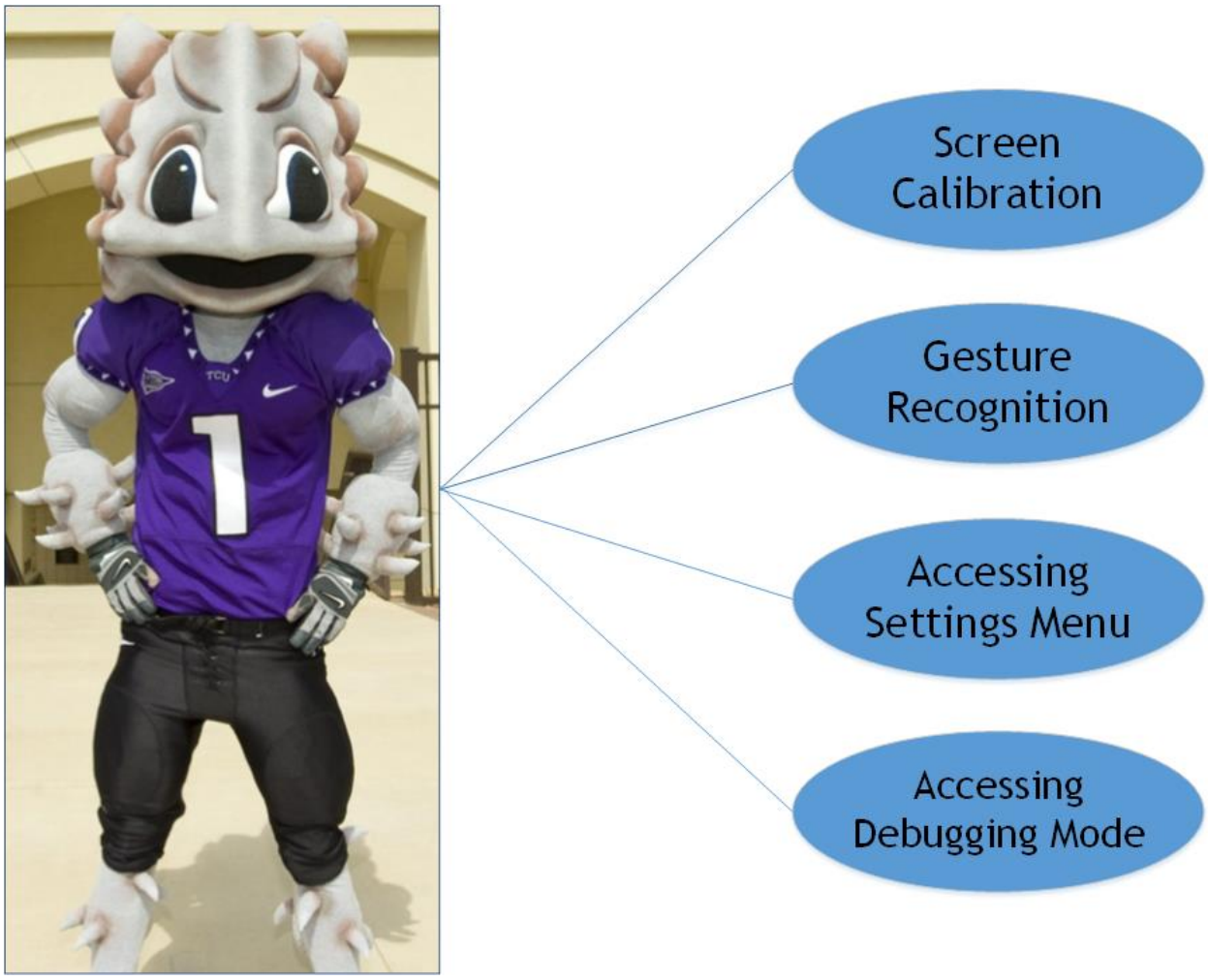


Figure 19

7.4 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.