Operating System Controller With Hand Gestures

Leftovers

Alec Knight
Cameron Barker
Dewi Endah Kharismawati

Final Project Documentation

https://docs.google.com/document/d/14iX7JXiikL31QS4hVnrZcaHapPg8xa8sGvcTKQKu69A/edit?usp=sharing

Final Project Code

https://github.com/Zaphster/Gestures

December, 2017

Table of Contents

1. Executive Summary

The group thought it would be fun to be able to control a computer by using hands and gestures. Inspired by movies such as “Minority Report” and “Iron Man,” the group set out to create a system that would allow users to do just that. To run the application, users need a 3D camera called leap motion controller to captured users gestures. In the application, users enable to have multiple user profiles, which each profile able to store gestures and commands mapped to them. Users also able to create their own gestures and named them as they like. After gestures are created, users need to mapped each of them into the provided commands.

The application need leap motion 3D camera controller to work. Leap Motion controller will captured gestures performed, and streamed the sequence of frames. These frames gave all different attributes for each gesture. The data is extracted based on five hand characteristics, which are hand, finger, bone, palm, and vectors. The application collects these data and process it in the decision tree algorithm. The decision tree is self created. It finds all values of attributes associated with gestures. Then, it calculates the ideal split based on all attributes collected. Finally, it creates nodes that keep track of the attribute with its probability value, and makes new children nodes depend on that probability. Children node get the list of gestures that can only be accessed from where they are and continue the process until the decision tree is fully created.

Decision tree that has been built is used for gesture recognition. Data from each frame is given to the decision tree. At each node, the attribute that the node picked is checked. If the value in the frame matches value at a node, then the tree adds the children of that node to a list. The tree continue checking until the list has a gesture in it, indicating gesture was performed, or list is empty that indicating no gesture was found.

2. Acronyms and Terminology

Leftovers: Team name

TigerMotion: program name

OS: Operating System

SDK: Software Development Kit of the leap motions

JSON: JavaScript Object Notation

JavaFX: A tool for building user interfaces that work with Java

UI: User Interface

ERD: Entity Relationship Diagram

IDE: integrated development environment

Leap motion controller: 3D camera usb peripheral that can capture hand gestures and extracting informations about the gesture performed

Agile: A style of project management in which development is rapid, requirements are revisited and refined often, and issues are addressed quickly.

3. Problem Statement

3.1. Need

With the creation of virtual reality and motion capture, a new world is ready to be stepped into. This futuristic idea has been seen in some science fiction movies, such as Iron Man, Matrix Revolutions, and Minority Report, where people move their hand and computer understands what the user wants with precision. The Leftovers became interested in this area and wanted to develop touchless OS control using hand gestures that utilizes a 3D camera.

In addition, based on Medical Daily, a typical computer mouse and keyboard have way more germs than a typical toilet seat. Based on their data, toilet seats have 49 germs in average, whereas keyboards have around 3300 germs and mice have 1700 germs in average. Public computers are even worse germ magnets. Therefore, touchless interactions between humans and computers is needed. Especially in the medical field where doctors, nurses, and patients need to keep their hands clean and sterile as much as possible. Touchless interactions are also needed in situations where a user has soiled their hands, doesn't want to soil their computer peripherals, and needs to use the computer.

3.2. Objectives

The main goal for this project is to bring this science fiction idea to life by developing an application that enable users to control their Operating System with their gestures. This application will enable users to interact with their computers in a cool way, without touching mouse, keyboard, or screen. In addition, the application is expected to be easy to use. Thus, all ages can enjoy the benefits.

4. Problem Analysis and Research

4.1. Problem Analysis

From the concept of need and objective from previous sections, there are several problems that need to be addressed on. To capture the gesture, the application will need leap motion 3D camera. How does this 3D camera works? How does it captures the gesture? How does the controller can recognize the gesture? These questions drives into other questions such as what algorithm to use for an effective and efficient result? What language to use to develop the program?

Based on the question and problem above, the team members divide some research question to be studied. The result are provided in the next sections.

4.2. Research

4.2.1. Cameron Barker

Programming Language for Leap Motion Operating System Control

For the application that the Leftovers are building, a Leap Motion controller will be used to capture a user's hand movements which will control cursor movements, allowing the user to interact with the operating system without using a physical mouse. The application will need to be built using a programming language, and as such this paper will explore possible languages and attempt to determine the best fit.

Ultimately, the Leftover's desire is to have an application that integrates with the Leap Motion controller and is usable on any operating system. As such, the chosen language should work with Leap Motion and be easily portable (without writing separate versions) to any operating system. Ideally, the language should be fairly easy to use in a group work environment, should be easily understood, and should make application development the main focus instead of dealing with nuances of the language or annoying bugs such as segfaults. In order to reduce misunderstandings when various group members are working on the code, a strong typing language is preferred. This will ensure that the number of bugs due to unexpected return types is minimized.

Leap Motion, as per their developer website, is designed to be compatible with several different languages: Javascript, Unity game engine (with UnityScript Similar to Javascript), C#, C++, Java, Python, Objective-C, and Unreal. That gives a good group of languages to choose

from. Each language will be examined individually and their benefits and drawbacks will be compared.

Right off the bat, based on the environments or the style of the language, it is apparent that three of these languages won't work: Javascript, Unity, and Unreal. Javascript is a web browser based language. It can run outside of the web browser environment, but that separate environment must be specifically created. The Leftover's created application should not depend on a browser being open and the Leftovers don't have the time or the desire to create an environment for Javascript to run in. These concerns remove Javascript from the pool of possible options. Unity and Unreal are both 3d game engines. If the created application were utilizing the Leap Motion controller to move around in or manipulate a 3d environment, both Unity and Unreal would be excellent choices. Since the goal is to simply control a cursor in the regular operating system environment, Unity and Unreal are not ideal.

C#, C++, and Objective-C are all languages based on the original C language. C has the advantage of being very close to the hardware and as such it has great performance. But C is not directly supported by Leap Motion (although one could argue that since C++ is close enough to C, and indeed near pure C can be written in C++, that Leap Motion supports C). Unlike C, C# has "strong type checking, array bounds checking, detection of attempts to use uninitialized variables, and automatic garbage collection" [1]. These characteristics make C# fairly easy to program in for programmers. The group can maintain a focus on the application itself instead of worrying about what is causing that segfault or whether they freed that memory. C# was designed to be deployed in distributed environments and be portable, fitting the needs of working on multiple operating systems. So far C# appears to fit the needs for the application.

C++ "has imperative, object-oriented and generic programming features, while also providing facilities for low-level memory manipulation" [5]. C++ is an extension of C with high level aspects for program organization. C++ uses a class object oriented system, has strong typing, and focuses on efficiency. It can be compiled on most systems with GCC, making it fairly portable. The typing system is implicitly strong, but can be overridden if explicitly told by the programmer. C++ supports exception handling, but has no garbage collection. Overall C++ is a decent choice, but the lack of garbage collection and the fact that the typing system can be overridden makes it not as ideal for ease of programming in a group setting.

"Objective-C is a general-purpose, object-oriented programming language that adds Smalltalk-style messaging to the C programming language" [4]. Objective-C "is a 'strict superset' of C," and as such the non object-oriented aspects of the language are exactly the same as C. There is no garbage collection, and segmentation faults are still a problem. For the object-oriented aspects, rather than calling a method on an object, objects are sent messages. The object may or may not respond to the message, based on whether there is an implementation in place for the message. Objective-C messages are not strictly typed. As for portability, Objective-C can be compiled to any system with GCC, just like C++. That requires compiling at the installation phase for users who would install the application. Overall Objective-C seems like a weak choice due to the lack of garbage collection, the possibility of segmentation faults, and the lack of strong typing.

After looking at the various C-like languages, if one is chosen it would be C#. The features in C# outshine the features in the other C choices.

"Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers 'write once, run anywhere' (WORA)" [6]. Java is compiled to bytecode that can run on a Java Virtual Machine (JVM) on any system without recompiling, making it very portable. Since Java is widely used, the vast majority of systems already have JVMs installed. Java utilizes an automatic garbage collector, C++ like syntax, and strong typing. The syntax and garbage collection make programming more about the development of the application at a high level instead of dealing with memory and other low level concepts. The typing system helps the programmers in a group setting to understand the overall program and prevents misconceptions. Java is a very strong choice for the application.

"Python features a dynamic type system and automatic memory management and supports multiple programming paradigms, including object-oriented, imperative, functional programming, and procedural styles" [7]. Python does have a garbage collector, and it focuses on being easy to read and on reducing the number of lines of code required to express concepts. The group would benefit from an easy to read programming language, and the garbage collection ensures that the group can focus on higher level concepts. There are python interpreters for many operating systems, so python is portable. So far Python seems to fit our criteria. However, the dynamic type system is not ideal based on the premises laid out. Python is strongly typed, so

types are enforced, but the dynamic typing means that any errors due to mistyping will be found at runtime instead of compile time. This forces the programmers to check for runtime typing errors by actually running the program, and as such some type errors may not be caught until it is too late. Python is a fairly good choice, but not the best.

Three of the languages that Leap Motion supports had obvious features or environments that were not compatible with the desired application. Of the remaining five, three were based on C, with improvements. C# was determined to be the best of the C extensions, with strong typing, portability (with separate compilation on the machine being ported to), object orientation, garbage collection, and none of the headache that comes with typical C programming. Python was portable, easy to read and develop, and focused on doing more with less code. Python suffered from a dynamic typing system that is checked at runtime, and as such falls short of C#. Java really shines with ultimate portability, compile time strong typing system, garbage collection, and ease of understanding the code in a group dynamic where many people will look at the code. Based on all considerations, Java wins the hunger games of the languages and will be utilized by the Leftovers to build the application.

4.2.2. Alec Knight

Machine Learning Research

Currently one of the biggest subjects in computer science is machine learning. The Leftover's current project entails recognizing gestures from input data using machine learning. There are many different ways in which machine learning can help in pattern recognition. There are also various algorithms that could be employed to implement machine learning within a program. There are many different problems that machine learning can help solve and pattern recognition is a common one. Some of the main machine learning algorithms are decision tree learning, discriminant analysis, and neural network based learning. These algorithms are classified as supervised algorithms predicting categorical labels and are typically used to put units of data into particular categories. These algorithms, if given hand data, would then be able to classify the data as matching a particular gesture or not matching any gestures. There are other types of machine learning algorithms for pattern recognition but supervised algorithms predicting categorical labels is the primary category which will be researched.

Discriminant analysis is a way of discovering features that characterize or separate two or more classes of objects or events. Discriminant analysis uses sets of observations or measurements compiled into a data set called the training set. The training set is used to help find a good predictor for the data. The goal is to be able to find a predictor good enough such that the predictor can be given a single unit of data, not from the training set, and with high probability categorized that data correctly. There are many different forms of discriminant analysis such as linear or quadratic and each are different in how they assume the data in the training set to be distributed. However they are each functionally identical. Discriminant Analysis is used in many different applications. In bankruptcy prediction, face recognition, though it's typically used to reduce the number of features present in a face to make it more manageable during classification, and in biomedical studies to help access patient severity and help to assess prognoses.

Decision tree learning is another type of machine learning algorithm. This algorithm uses decision trees to help it make classifications about objects. The algorithm uses decision trees to map observations about an item to a conclusion about the item's target value. Usually the decision tree model classifies objects by creating a tree with the leaves representing the classification and the branches representing the conjunction of features that lead to that classification. For example if a decision tree is given features of a person it may be able to

classify that person into some category such as boy or girl, young or old, etc. There are many different types of decision trees used in machine learning and each has their unique uses. Classification trees are used to predict which class the input data belongs to. Regression trees are used when the outcome predicted is a real number such as a price. The learning part of decision tree learning takes place when a decision tree is constructed from a training set and then that tree is used to help make classifications of objects. Currently the Kinect Development Kit created by Microsoft uses of form of decision tree learning in order to create new gestures recognizable by its Kinect camera. Despite its uses there are also some drawbacks to decisions trees. Decision trees tend to be not as accurate as other approaches. Implementation decision tree algorithms tend to be slow or hard to compute. Some problems are not represented well in terms of a decision tree.

Another form of machine learning is neural networks. Neural networks are loosely based on the neurons in the brain and the way in which they are interconnected which why they have the name neural networks. A neural network is a interconnected system of simple units typically called neural units. Each neural unit is connected with many other neural units. Each unit is used to compute a simple function. Each unit computes using a summation function but there may also be a limiting or threshold function on the unit as well. The neural units contain parameters called weights which can change how the data is computed in each. The system itself is self-learning and trained and is usually composed of multiple layers. Each layer sends it data to the next layer. The multiple layers allow for a single layer to focus on a specific job. For example the first layer can be designated the input layer, the next layers are used to find the specific output being looked for and the last layer can be used as the output layer.

Learning in neural networks can be approached in several different ways. Supervised learning is used when there is example sets of data that can be fed into the system. This is usually done in the form of pairs which consist of a input and the desired output. The network must then find a function that map the pairs together. The function that the network finds can then be used to help map new examples not found in the training data. Another type of learning uses unsupervised learning which does not require a training set of data. In unsupervised learning the network is given unlabeled data rather than the paired data that was given in supervised learning. In unsupervised learning the network tries to find a way to map the input like in supervised training. However, unlike supervised training, the network, first, tries to reduce the input as

much as possible. Using the reduced input a mapping function is created and the network tries to generate output similar to the input using the mapping function. There is also reinforcement learning which concerns how software acts in an environment such as a game. Currently there are many systems that are based on neural networks and are used for things such as function approximation, classification, and even robotic decision making. Deep learning is another example of machine learning that is similar to a neural network.

Deep learning is a type of machine learning algorithm that is similar to a neural network. Deep learning uses multiple layers composed of nonlinear processing units sometimes called neurons. Each layer in the system uses the output from the previous layer. At each layer the input is transformed by the processing units that make it up. Each processing unit has a weight that is determined through either supervised or unsupervised training. The weights, like in a neural network, are the parameters that are contained by each processing unit. The main difference between neural networks and deep learning is that deep learning usually has far more hidden layers then neural networks . Neural networks have up to two layers in between the input and output layers called the hidden layers. These layers are what transform the input data into the correct format for the network to output. Deep learning has more hidden layers then neural networks. Deep Learning usually solves problems in the same way that neural networks do. The Deep Learning algorithm is given data the is either labeled or unlabeled and the goal is that the algorithm can find a function that maps the input data to the correct output data. There are currently many uses for Deep Learning such as automatic speech recognition, image recognition, and other types of pattern classification.

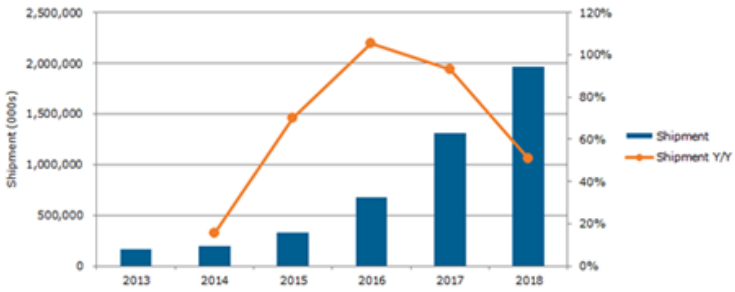
One of the primary uses of machine learning is for pattern recognition such as hand gesture recognition. There are many different methods that could be used to find hand gestures given the necessary hand data. The main goal of gesture recognition is to map a set of input data to a certain gesture. The use of machine learning algorithms would allow the user to create and store their own gestures that the program could learn to recognize. An example of this is the Microsoft Kinect. The Kinect uses decision trees to allow it to recognize gestures. The Kinect development software uses decision tree learning to allow developers to add gestures. The software allows a developer to store a gesture by creating a decision tree from examples of the gesture. The Kinect can then use that decision tree to determine whether the hand is in the correct position for that gesture.

Machine learning is a powerful tool for pattern recognition. The use of which can add a lot of functionality to software and hardware. There are many different algorithms that are classified under machine learning. Understanding the different types of machine learning algorithms is important. Some algorithms are more efficient handling certain problems than others are.

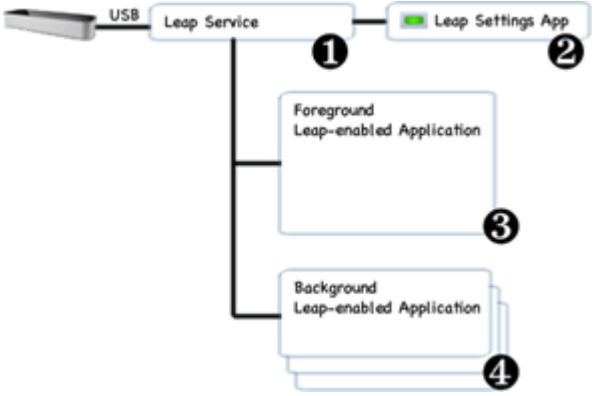
4.2.3. Dewi Kharismawati

Leap Controller and Trainer

Augmented reality head mounted displays (HMDs) enable users to navigate their computer hands free. One kind of HMD that closer to human life is through the hand gestures. Leftovers group interested in this area and wanted to develop touchless cursor controller using hand gesture that utilizing 3D camera. The main goal for the project is to build a system that is able to move and operate cursor pointer on the computer screen using user's 3D movement without touching mouse or screen. This futuristic idea has been seen in some science fiction movies, such as Iron Man, Matrix Revolutions and Minority Report, where people move their hand and computer understands what the user wants with precision. In short, the concept is transformed the hand gestures to be virtual mouse. As a result, computer user will experience a satisfactory advance technology that make their interaction with computer a lot easier. Also, the system will help people who has either physical or cognitive limitations to interact with the computer. Technology with gesture sensing is rapidly develop these days. In 2015, almost 330 million smart devices with gesture sensing is shipped [19]. From the chart, it shown that in 2018 it is predicted to have 500% increase in shipping compare to 2014. Based on this promising growth, my team is going to utilize a technology with gesture sensing to build the touchless cursor controller. After doing some research, the team agree to employed Leap Motion 3D camera technology in the development process. The main reason the team decide to use Leap Motion because it has zero latency and low processing power. Despite from the background, all members in this team is new to Leap Motion technology, and have limited knowledge on how it is actually work, from receiving the data, processing the image, giving the output. Furthermore, the team concern is that whether it is possible to work integrated with the Operating System since the program will make changes in the mouse input. Therefore, this research paper will be focused on which approach is the most efficient in the image processing techniques to get the most precise and accurate output to control the cursor on the computer.



Leap motion controller and Leap trainer is two important parts that act as a bridge between the human hand gestures and the computer. The Leap Motion controller has a main function to track and capture hand gesture, while Leap Trainer is the gesture recognition framework. Leap Motion controller is a small device that connected by the computer via USB. It able to capture hand movement up to 200 frame per second in 150^{o} scope with 8 cubic feet interactive 3D space[20]. The controller has three LED light, and two monochromatic infrared cameras (IR). In the process of capturing the movement, the device generates 3D pattern of IR, then send it to the computer through the USB. Once the data is receiving by the computer, the software called Leap Motion Service processed the raw data using complex mathematical lifting of synthesizing the 3D position data and compare it to the 2D frames that acquired by the cameras [21]. The system application receiving the tracking data from the Application Program Interface (API), and the software development kit (SDK) has two API to retrieve the tracking data, which are Native Interface and WebSocket Interface. Based on our design, native interface is more appropriate for the implementation stage because the leftovers is not building web application in the first place. With native interface, the controller send gesture tracking via USB.

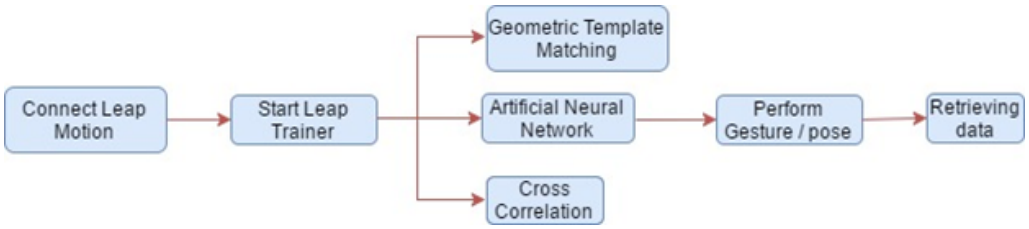


In default, the application can get the data when it is open in the foreground desktop. Therefore, the controller need to be set to be able to retrieve data when the application is running as the background since the main function of our program is to navigate the cursor to operate other programs. In the other word, the program still need to retrieve the data from the controller, even the application is in the background. To enable this feature, Leap Motion provides Controller.set_policy() method and set the BACKGROUND_POLICY_FLAG policy is needed since our program is mouse emulators that need to control the operating system.

```
controller.set_policy(Leap.Controller.POLICY_BACKGROUND_FRAMES)
controller.set_policy(Leap.Controller.POLICY_IMAGES)
controller.set_policy(Leap.Controller.POLICY_OPTIMIZE_HMD)
```

three command above is required for my group project to enable the background data retrieving [22].

After data is retrieving, the system need a platform to recognize what gesture was retrieved. Leap Motion has Leap trainer as a framework gesture recognition. Leap trainer classified gestures into pose for the static hand movement, and gesture for hand movement that has start, end, and to which direction. This far, Leap Motion SDK only recognize four basic gestures, which include circle, swap, screen tap, and key tap. Developers are allowed to create customs gesture, and here is the main function of the Leap Trainer. It helps the system to train for recognizing new custom gestures. How Leap trainer works can be seen in the new gesture creation. The whole process is depicted in the chart below.



Leap Trainer Workflow

In deciding which techniques to use, each technique has their own capabilities. Artificial Neural Network (ANN) promises less technical because it is crude electronic model of the brain. The main goal of ANN is to solve the problems like how the human brains work. The second technique Cross correlation is mathematical approach that measure similarity and relativeness between two functions. It commonly used to search known features. Last, geometric template matching is digital image processing technique to find small parts of an image that match with the template image. Based on [23], template matching required two images, which are template image and source image. This technique is derivation from $P Point-Cloud Recognizer, a 2D gesture recognizer that was built for rapid prototyping many gesture based user interface. Research from source [23] is utilizing all three techniques to create their Sign Language to Text Converter. The application basically translates from twenty-six alphabet gesture into text. From the observations using the three techniques, Geometric Template Matching held the highest accuracy percentage, which is 52.56% from three test over the twenty-six alphabets. Artificial Neural Network come second with 44.87% and Cross Correlation come the third with 35.90% [23]. Based on this finding, the team agree to use Geometric Template Matching technique as the Leap Trainer for the users to input new gestures. The concept is that users need to upload the gesture template they want with the recognized command into our database. Leftovers is going to

provide a platform about our application with this feature, and then if other users like the gestures and its function, they can download the gesture. However, still based on [23], they found out all techniques have difficulties recognizing similar gestures. Therefore, there will be rules and regulations for new gestures uploaded, such as the similarity of the new gestures with the existing on the database.

In conclusion, the team is interested to build a program that work on touchless cursor controller using Leap Motion technology. Leap Motion is a new technology for the team, yet has a lot of part to explore, especially how this tool retrieving and processing the data in general. The program will be a background leap-enable application, it is a little bit tricky and need to follow the policy due to OS exploration. In addition, Leap Motion SDK recognized four gesture, and the team intended to add more and create upload and download platform for the user to have custom gestures and their functions. Geometric Template Matching technique will be implemented to train the program towards the new gestures because it has highest accuracy compare to the other two techniques.

5. Requirements

TigerMotion OS Controller has requirements that users need to make the application run perfectly.

5.1. Hardware Requirements

a. Leap Motion 3D camera.
b. AMD Phenom II or Intel core i3/i5/i7 processor (or better).
c. Minimum Windows 7+ OS or Mac OS X 10.7+
d. 2GB RAM (or more)

5.2. Software Requirements

a. Leap Motion SDK - https://developer.leapmotion.com/get-started/
b. JSON Simple http://www.java2s.com/Code/Jar/j/Down4 loadjsonsimple111jar.htm

5.3. User Requirements

The application is expected to be able to receive gestures and interpret them as commands. Specifically hand movements for actions and cursor movement.

a. User can create new gestures
b. User can map any created gesture to a predefined set of commands
c. User can create new profiles that have different configurations of gestures, action mapping, and settings
d. Support all widely accepted basic user interface actions
    i. Move cursor
    ii. Click
    iii. Double click
    iv. Right click
    v. Scroll

6. Constraints and Applicable Standards

Based on the one semester limit time to develop the project idea, the team put some application constraints and standards to be achieve in the end of semester.

1. Create new profile and its JSON file to store gestures and commands mapping
2. Create new gesture page
3. Recognize gestures that have been created
4. Mapped the gesture to commands provided
5. Enable the gesture tracking to move cursor and operates computers.
6. Application expected to have a high gesture recognition and commands actions accuracy

7. Design

7.1. Leap Motion Overview

Leap Motion Controller

The Leap Motion system is able to track and recognize hands and fingers. The Leap Motion is a small device that connects to the computer via USB. It has a main function to track and capture hand gesture. The controller utilizes optical sensors and infrared light to give high precision and tracking frame rate and report discrete position and motion in frames at the Leap Motion frame rate. It is able to capture hand movement up to 200 frames per second in 150^{o} scope with 8 cubic feet interactive 3D space. The sensors are directed upward (along with the y-axis), field of view about 150 degrees, and work optimum in range 25-600 millimeters (1 inch to 2 feet). The controller has three LED lights and two monochromatic infrared cameras (IR). In the process of capturing movement, the device generates 3D pattern of IR, then sends it to the computer through the USB. Once the data is received by the computer, the software called Leap Motion Service processes the raw data using complex mathematical lifting of synthesizing the 3D position data and compare it to the 2D frames. If the comparison match, gesture is recognized.

Leap Motion data is processed by the drivers and accessible by the API. The main data retrieved from the API is a Frame. One frame consists of fingers, hands, pointables, built in gestures, timestamp, rotation, scaling data, and translation. For this project, the team will create

custom data format that will only contain the necessary data for processing, testing, and training purposes.

7.2. Language Choice

Leap Motion, as per their developer website, is designed to be compatible with several different languages: Javascript, Unity game engine (with UnityScript Similar to Javascript), C#, C++, Java, Python, Objective-C, and Unreal.

Javascript is mainly used in web browsers, while Unity and Unreal are 3d game engines. These properties make those three languages not an ideal choice.

C#, C++, and Objective-C are supersets of C. C# improves on C with object oriented capabilities, garbage collection, array bounds checking, and detection of attempts to use uninitialized variables. Of these three C-like languages, C# fits the needs of the Leftovers the best.

Java is compiled to bytecode that can run on a Java Virtual Machine (JVM) on any system without recompiling, making it very portable. Since Java is widely used, the vast majority of systems already have JVMs installed. Java utilizes an automatic garbage collector, C++ like syntax, and strong typing. The syntax and garbage collection make programming more about the development of the application at a high level instead of dealing with memory and other low level concepts. The typing system helps the programmers in a group setting to understand the overall program and prevents misconceptions. Java is a very strong choice for the application.

In addition, Java has JavaFX, a set of graphics and media packages that enables developers to design, create, test, debug, and deploy rich client applications that operate consistently across diverse platforms. This features will help in the process of building the application's user interface.

Since Java and C# are very similar and Java has extremely good portability, Java will be used to build the application.

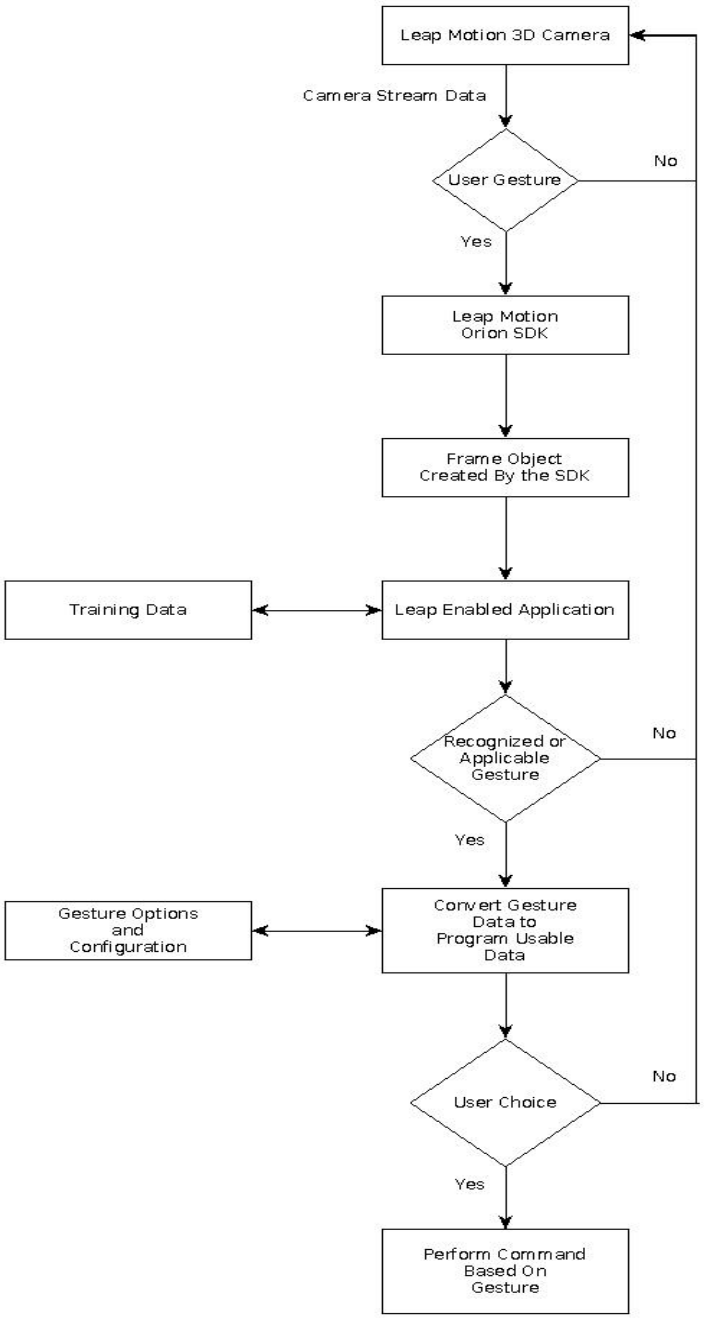
7.3. Gesture Recognition Overview

7.3.1. Sample Gestures and Expected Actions (editable by user)

Example Gesture	Commands
	• Performed by the user when he/she wants to move the cursor
	• Performed by the user when he/she wants to left click
	• Performed by the user when he/she wants to open the start menu
	• Performed by the user when he/she wants to close the current selected window
	• Performed by the user when he/she wants to open program shortcut 1, shortcut 2, or shortcut 3 • Shortcuts would be determined by the user configuration

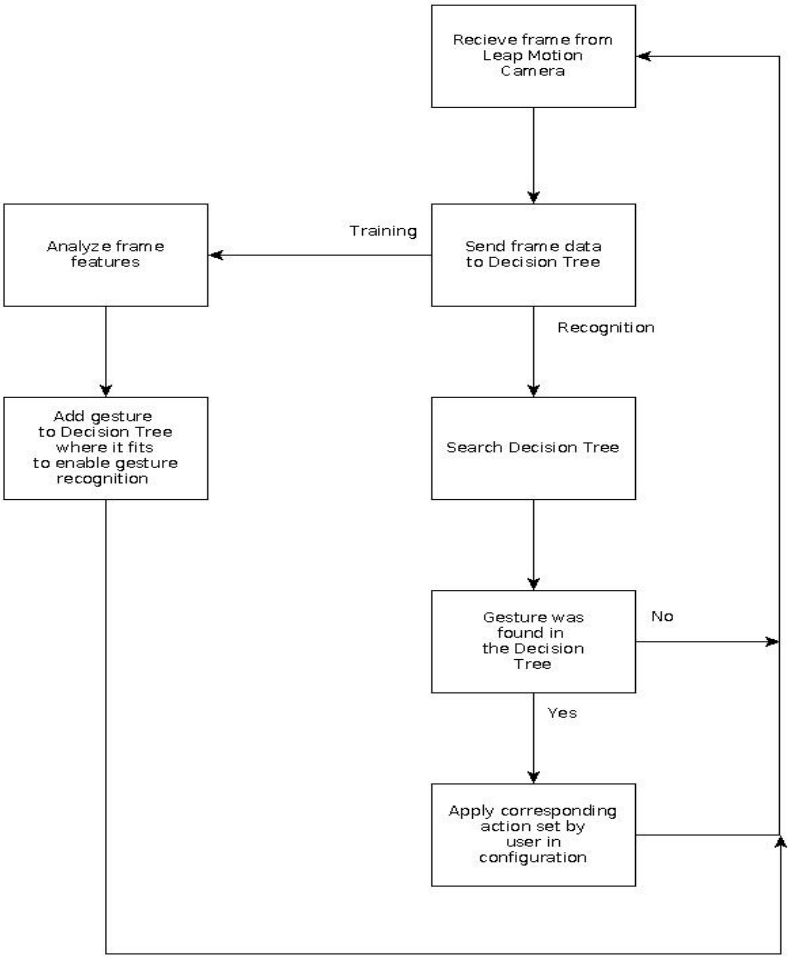
7.3.2. TigerMotion OS Controller Overall Workflow

Leap Motion captures data in sequence of snapshot, called frames. When the leap motion streams the data in each frame, data needed is extracted. Then, the data is sent to the decision tree for recognition. If the gesture is recognized, it will be proceed based on the mapped actions chosen by the user. Below workflow summarize the process.



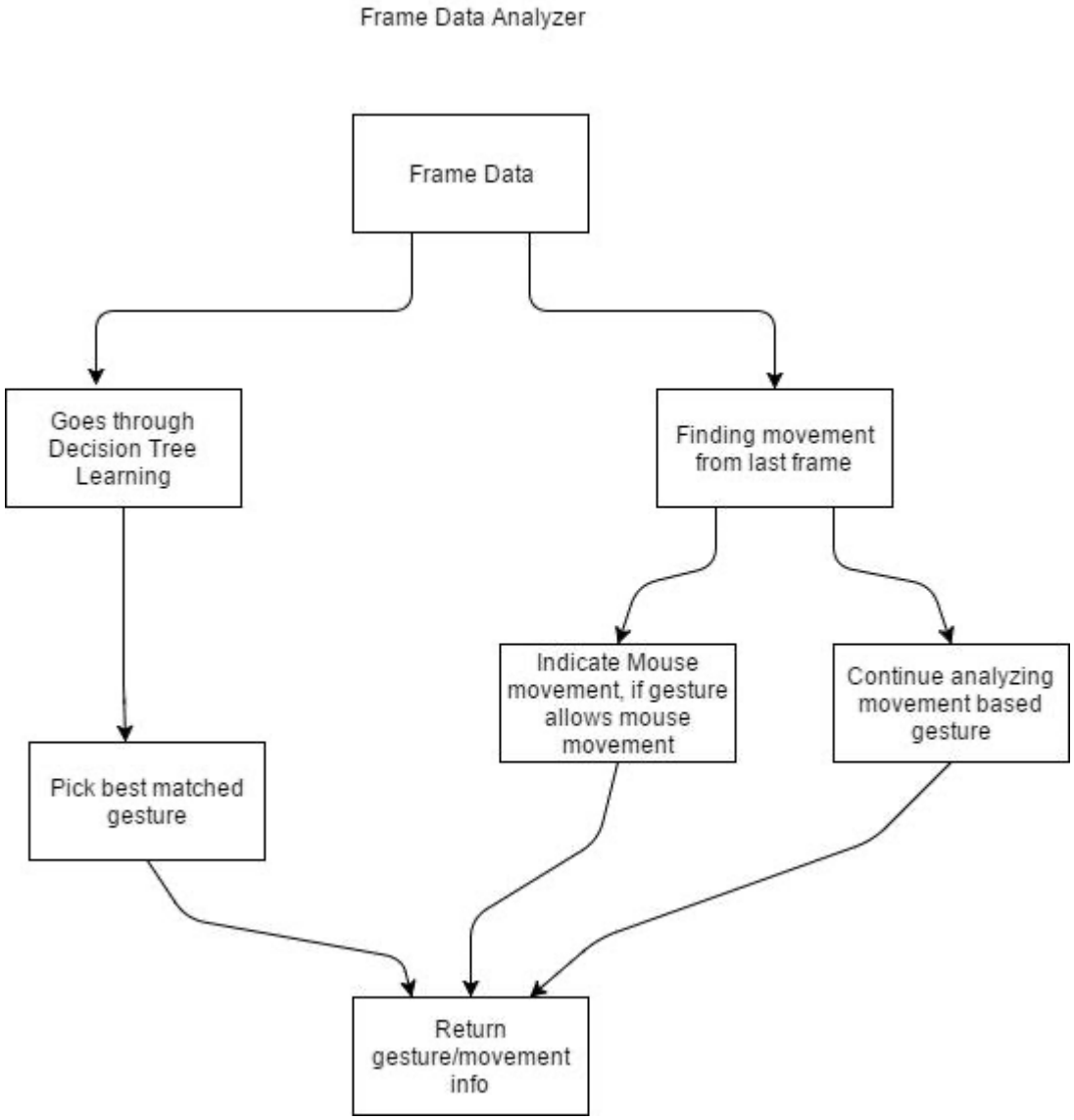
7.3.3. Frames Data processing to Identify the Gestures Performed

For image data processing, TigerMotion is using decision tree for training and recognizing gesture performed by the user. When the data is received, it sends the gesture to the decision tree. In creating new gesture case, the decision tree will train the data received, and add this new gesture features into new nodes in the decision tree. In recognition case, the data received will go through the decision tree nodes. If the features from the data received match with the features in the decision tree, the algorithm will traverse the decision tree until it finds a leaf node. If the leaf is null, it means the gesture has not been registered, otherwise the gesture is recognized. Then, the application will applied the gesture into the corresponding action set in the configuration. Bellow is the workflow of this gesture performed.



7.3.4. Frame Analysis Workflow

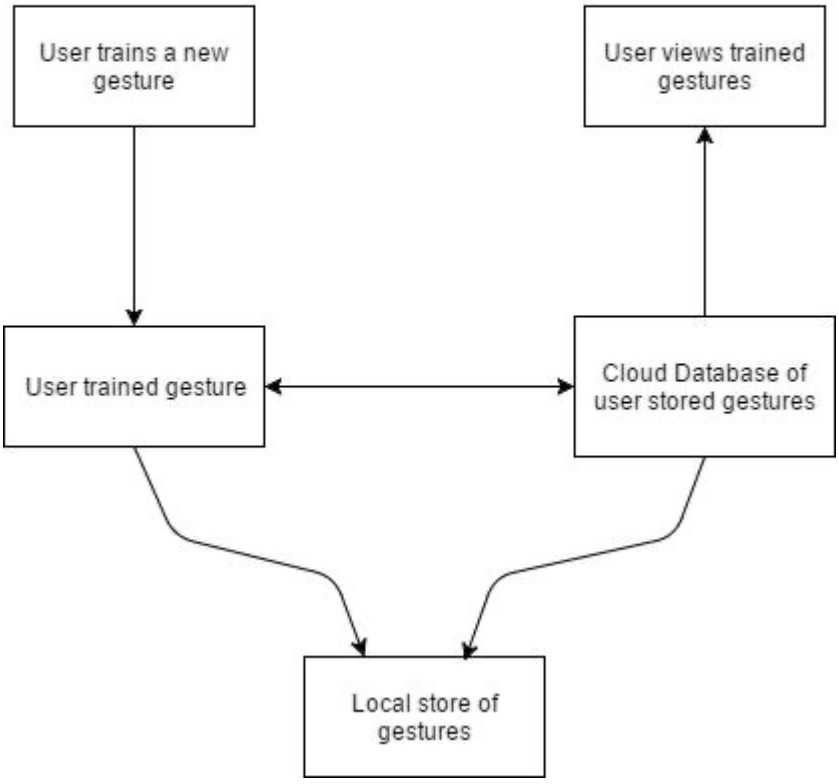
Leap motion controller is able to capture 60 frames per second of reflected gesture data. From frames data, it can characterize the gesture performed as static or moving gestures. For the static gesture, data is sent to decision tree for processing. Otherwise, the application will analyze the movement and associate it with mouse movement.



7.3.5. User Acquisition of New Gesture

In the new gesture acquisition, users will be able to create new gestures by themselves. First, they need to perform the new gesture. Leap Motion will retrieve the data, and the program will train that new gesture. After the gesture is trained, it will automatically be stored in users' local computers. The workflow of new gesture acquisition is provided below. (Cloud database of user stored gestures has been removed.)

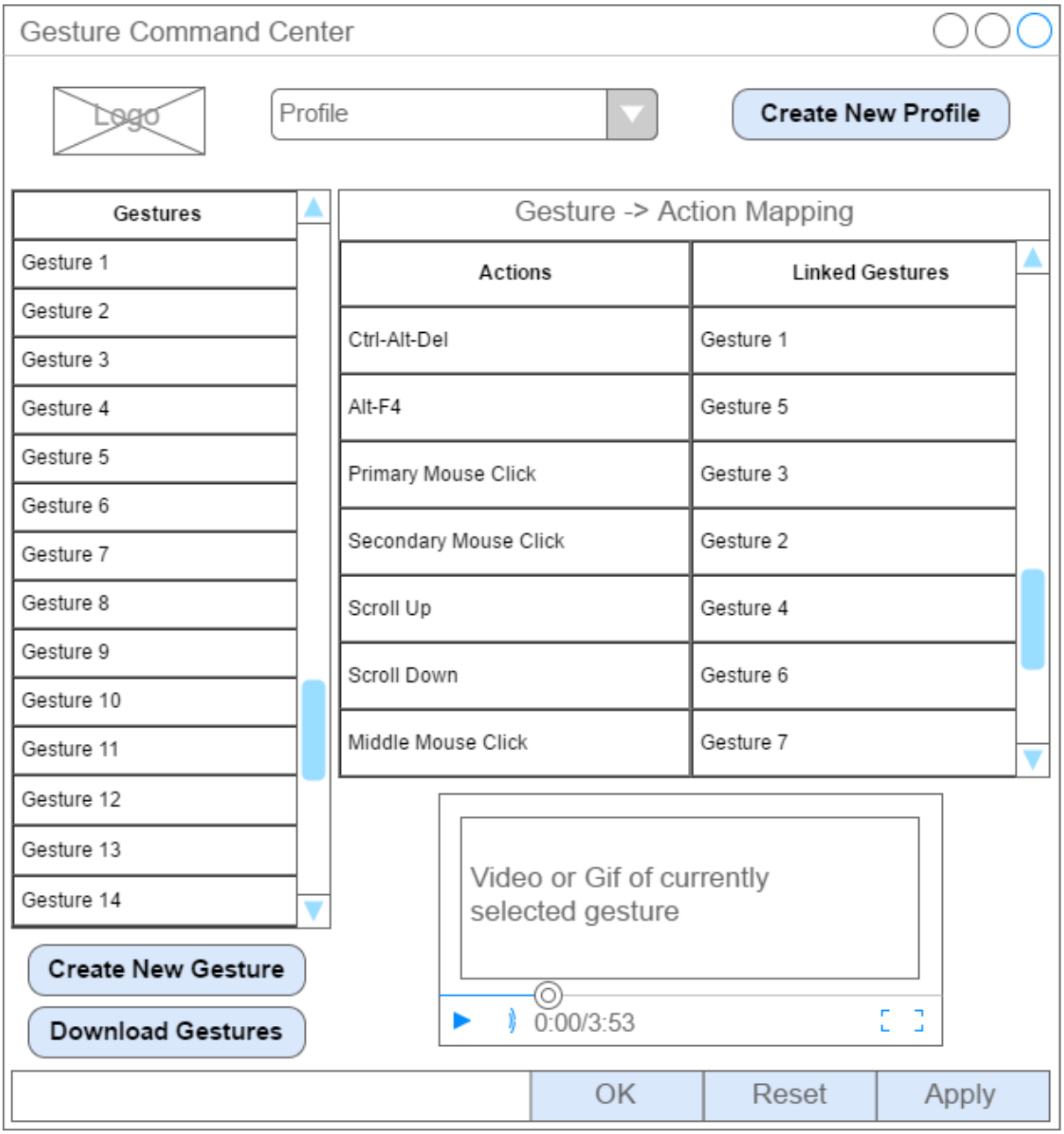
User Acquisition of New Gestures



7.4. User Interface for Gesture Library

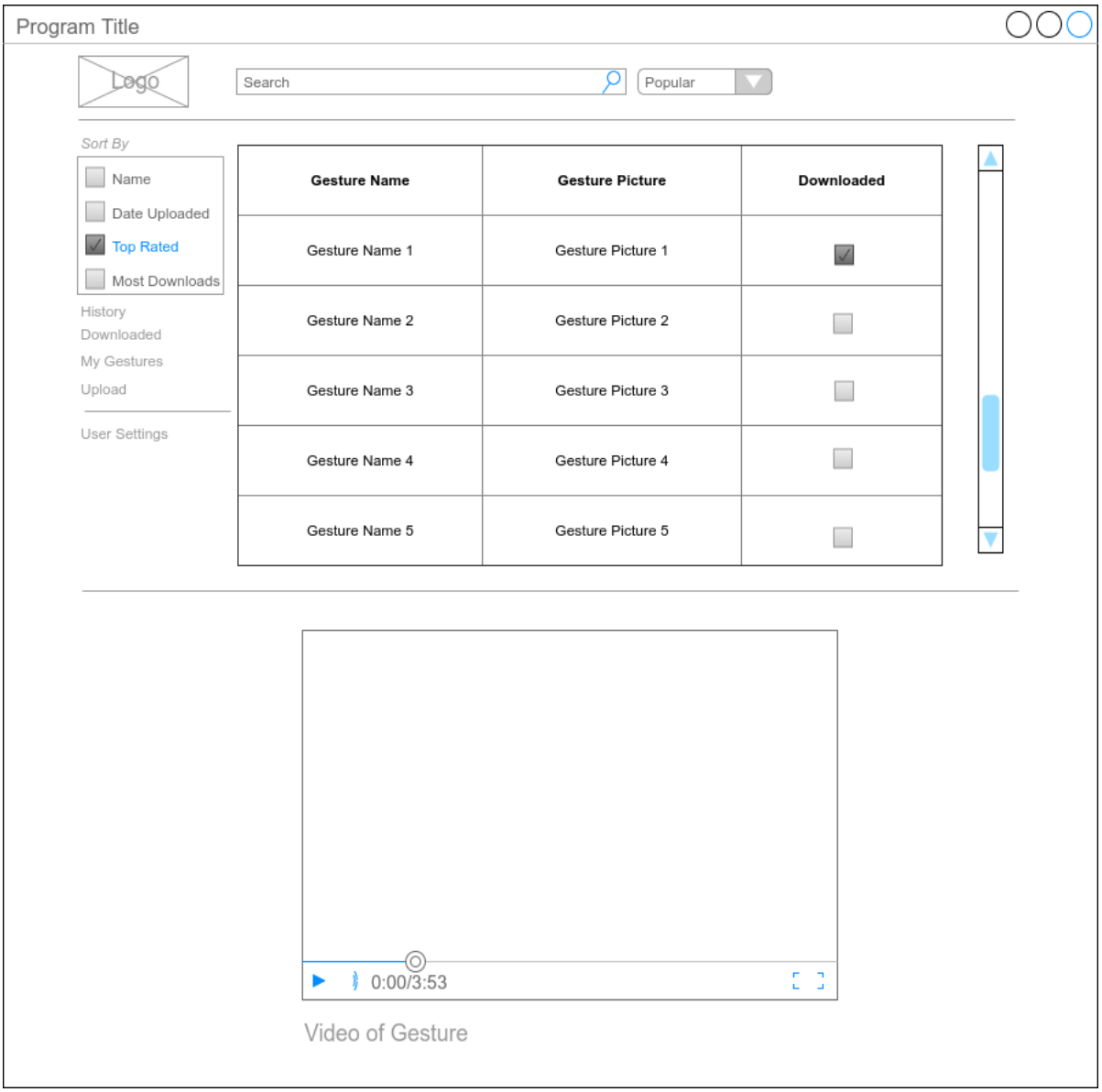
7.4.1. Options Page UI

This page will allow the user to modify what gestures are linked with which actions, view the recognized gestures, and train new gestures. The user will also be able to switch between saved profiles and create new ones. The download gestures button will take the user to the download page to download gestures previously uploaded to the database by other users.



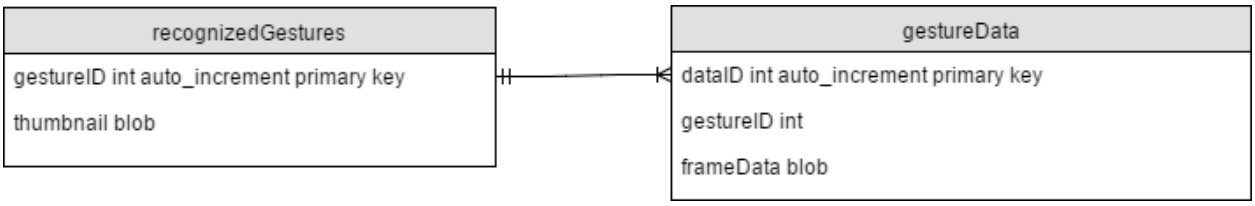
7.4.2. Download Gestures Page

This page will allow the user to view gestures that have been uploaded to the database and download gestures that the user wants.



7.4.3. Database Management Design (Removed from final project)

This project is proposing to use Amazon Web Service for database management system.The database is designed to store the gesture data that is uploaded by the users. Simple ERD is provided below.



The group decided not to use cloud database. The user gestures and configurations can be stored locally in self generated JSON files.

7.5. Data Processing with Decision Tree

This project is proposing to use Decision Tree as the main technique for the data analyzing. Decision tree is chosen because based on [0], the accuracy of decision tree is 95.8%. Besides, with the decision tree it will be easier and simpler to handle a lot of data with efficiency of O(n log n).

7.5.1. Decision Tree overview

Decision tree algorithm uses a decision tree to help it make classifications about objects. The algorithm uses decision trees to map observations about an item to a conclusion about the item's target value. Usually the decision tree model classifies objects by creating a tree with the leaves representing the classification and the branches representing the conjunction of features that lead to that classification. For example if a decision tree is given features of a person it may be able to classify that person into some category such as boy or girl, young or old, etc. There are many different types of decision trees used in machine learning and each has their unique uses. Classification trees are used to predict which class the input data belongs to. Regression trees are used when the outcome predicted is a real number such as a price. The learning part of decision tree learning takes place when a decision tree is constructed from a training set and then that tree is used to help make classifications of objects. Currently the Kinect Development Kit created by Microsoft uses a form of decision tree learning in order to create new gestures recognizable by its Kinect camera. Despite its uses there are also some drawbacks to decisions trees. Decision trees tend to be not as accurate as other approaches. Implementation decision tree algorithms tend to be slow or hard to compute. Some problems are not represented well in terms of a decision tree.

There is research that utilizes a decision tree algorithm for sign language recognition with Leap Motion. In the research, leap motion's features, such as hands or fingers locations, directions, and movements are used to get the data. The result for the experiment shows that the recognition rate reached 95.8%[0].

7.5.2. Decision Tree implementations

The design of the decision tree algorithm based on [0] for this program is as follows:

1. Treat all training data that has provided the best attributes to be the root node of the tree.
2. Each attribute has its own corresponding branch, and each branch can create new nodes. For the selection standard of the best attribute, the choice of testing features of internal nodes shall be conducted on the basis of Mahalanobis distance.
3. Classification based on Mahalanobis distance.

The mahalanobis distance is calculated by this formula: $\Sigma=(\sigma ij)p\times p$

Σ is the covariance array that expresses the gesture data. Where σ is covariance matrix and $\bar{x}$ is mean can be derived from the training data using the formulas

$$\sigma_{ij} = \frac{1}{n-1}\sum_{a=1}^{n}(x_{ai} - \bar{x}_i)(x_{aj} - \bar{x}_j)\mathrm{i,j} = 1, \cdots, \mathrm{p},$$

$$\bar{x}_i = \frac{1}{n}\sum_{a=1}^{n} x_{ai}, \qquad \bar{\mathrm{x}}_j = \frac{1}{n}\sum_{a=1}^{n} x_{aj}$$

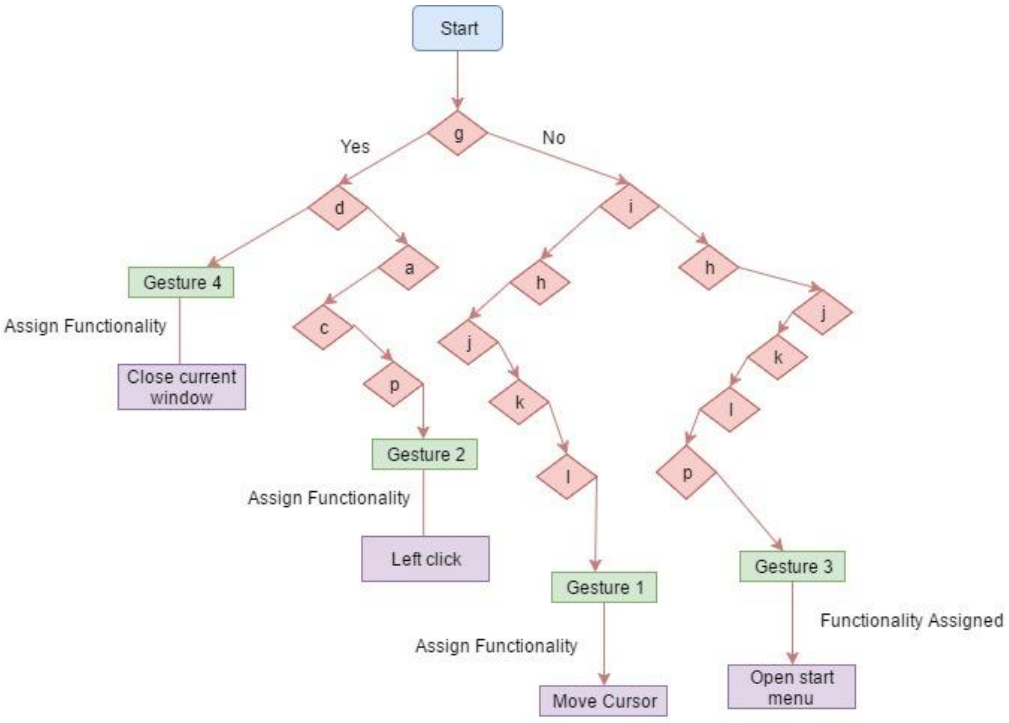
Mahalanobis distance can be calculated from formula

$$d_{ij}^2(M) = (X_t - X_j)'\Sigma^{-1}(X_i - X_j)$$

Xi is the vector composed by p indexes of gesture sample Xi, namely No i vector of the source material array. Sample Xj is similar. From the Mahalanobis distance, the team is planning to divide the classifications into 16 classes. In the decision tree these will be the branch classifications. The following table shows the 16 classes with what they belong to.

	Conditional Branch
a	The palm of hand is facing the opponent.
b	Hand orientation is above.
c	There is a finger that makes a wheel with the first and the second joints
d	The third joints are extended except the thumb.
e	The back of hand is facing the opponent.
f	Hand orientation is below.
g	All of the first and the second joints are bent.
h	The thumb is fully extended.
i	The index finger is fully extended.
j	The middle finger is fully extended.
k	The ring finger is fully extended.
l	The pinky finger is fully extended.
m	The thumb is contact with the middle finger.
n	The ball of the middle finger is on the nail tip of the index finger.
o	The index finger and the middle finger are separated.
p	The thumb and the index finger are open.

The flowchart below is one example of the decision tree implementation in this project. This decision tree decides what gesture the user performs, the tree will help the system to recognize what gesture is performed. Then, the identified gesture will be translated into specific commands.



Based on the tree above, the recognizing scheme is flowing from the start to the assign functionality. When the user performs a gesture, the specification of that gesture will be run through the tree. If the classification is true, it goes left, and if false, it goes right. When it successfully identifies the gesture, the last node is the command to be executed by the OS. There is efficiency challenge in the use of decision tree. From the tree above, there are some redundancy of the classes that are presented. After some research, one solution that can be applied to reduce the redundancy is by putting the best attribute that is most frequently used at the top part of the tree.

7.6. Mouse and Keyboard Click Controller

7.6.1. Possible Actions To Be Controlled By Gestures

MouseButtons = [Left, Right, Middle]
MouseButtons.forEach(){
Mousedown
Mouseup (Triggered when gesture for Mouse Down no longer being performed)
(all mouse down gestures will also allow cursor movement during them)
}
Cursor Movement
Scroll Up
Scroll Down

--- These may be impossible with a Java library due to the nature of them being operating system specific ---

Close window

Windows/Linux: Alt-F4

Apple: Command-Q (close application)

Command-W (close current window)

Task Manager/Process List

Windows: Ctrl-Alt-Del (or similar operating system dependent operation)

Linux: (unknown)

Apple: Cmd-Opt-Esc

Volume Mute/Unmute

Volume Up

Volume Down

Press Windows Key/Start Menu/OS specific special-ness

7.6.2. Java Library For OS Control

Java.awt.Robot

keyPress(int keycode)

Presses a given key.

keyRelease(int keycode)

Releases a given key.

mouseMove(int x, int y)

Moves mouse pointer to given screen coordinates.

mousePress(int buttons)

Presses one or more mouse buttons.

mouseRelease(int buttons)

Releases one or more mouse buttons.

mouseWheel(int wheelAmt)

Rotates the scroll wheel on wheel-equipped mice.

7.7. Pseudocode and Object Descriptions

Leap Motion Engine: Orion
Programming Language: Java

7.7.1. Pseudocode

7.7.1.1. Example Leap Motion Program

```
class Sample{
	public static void main(String[] args){

		SampleListener listener = new SampleListerner();
		//The listener for Leap Motion Events.
		//The controller calls the Listener when event occurs


		Controller controller = new Controller();
		//The main interface for the Leap Motion Controller
		//The frame data can be polled using the controller
		//The controller stores up to 60 frames in its frame history


		controller.addListener(listener);
		//adds the listener to the controller

		System.out.println("Press any key to quit");
		//Runs the loop until a key is pressed(just for the sample)
		try{
			System.in.read();
		} catch (IOException e) {
			e.printStackTrace();
		}
	}
}
```

7.7.1.2. Example Leap Motion Listener:

```
class SampleListener extends Listener{

        public void onConnect(Controller controller){
                System.out.println("Connected");
                //The method that is called once the controller has connected successfully
                //Once connected the controller changes its isConnected property to true
                //The controller then calls the onConnect function

        }

        public void onFrame (Controller controller){
                System.out.println("Frame available");
                //The onFrame function is called by the controller each time a frame is available

                Frame frame = controller.frame();
                //The frame is accessed by calling the controller.frame() method
                //The onFrame function makes sure that there is a frame available to access
        }
}
```

7.7.1.3. Controller Class Important Member Functions:

Frame(int history)
- Returns a frame from the leap motion software
- Can return a frame from the frame history kept by the controller
- Up to 60 frames are kept

Frame ()
- Retrieves the most current frame from the leap motion controller

ImageList ()
- A list of the most recent images captured by the leap motion controller

isConnected
- Boolean on whether the controller is connected to the system or not

7.7.1.4. Listener Class Important Member Functions:

onConnect()
- Method called by the controller when the controller connects to the system

onFrame()
- Method called by the controller when a frame is available

7.7.1.5. Hand Class important Member Functions:

direction ()
- A vector containing the direction of the hand

fingers ()
- Returns a list containing the fingers attached to the hand

grabAngle ()
- The angle between the fingers and the hand

isRight ()
-Returns whether hand is the right hand or not

isLeft()

rotationAngle (Frame)
- Angle of rotation since the last frame

Finger Class Important Member Functions:
bone()
- returns the index of the bone that being looked for

Bone Class Important Member Functions:
center()
- Returns the center of the bone

direction()
- Returns the normalized direction of the bone

length()
- returns the length of the bone

Frame Class Important Member Functions:
hands()
- returns the list of hands in the current frame

fingers()

- returns the list of fingers in the current frame

8. Testing / Data Collection and Analysis Plan

Testing is being done in all parts and layers of this application. The test plan is focusing on gesture development, command configurations, GUI, Security, Usability, and performance.

8.1. Development testing of each new gesture that is trained:

a. A gesture is recognized
b. Recognized gesture matches the gesture performed.
c. Gesture performed maps to correct user's command for that gesture.
d. Verify Gesture is added correctly to Decision Tree.

8.2. Testing of user configured commands.

a. Cursor movement
   - Up
   - Down
   - Left
   - Right
b. Mouse events (clicks)
   - Primary mouse button
   - Secondary mouse button
   - Middle mouse button
   - Scroll up
   - Scroll down
c. Window Control
   - Close Window
   - Minimize Window
d. Keyboard commands
   - Windows
     - Alt-F4
     - Ctrl-Alt-Del
     - etc...
   - Mac
     - Option-W

    - Option-Q
    - etc…
- Linux
    - Ctrl-Alt-Del
    - etc...

8.3. Usability and performance testing

a. Is it intuitive? Have others use it.
b. Are gestures recognized?
c. Are gestures recognized quickly?
d. Is it responsive? Do users feel frustrated when using it?
e. Is adding a new gesture intuitive?

8.4. GUI testing

a. Click each button, ensure it does what is expected
b. UI is responsive
c. Profiles can be created and switched to
d. Action to gesture links properly switch when profiles switch
e. Gestures can be mapped to actions
f. User defined keyboard commands can be added
g. Reset function applies default gestures to actions

8.5. Database testing

a. Upload gesture, verify upload
b. Download gesture, verify download
    - Verify downloaded gesture can be correctly inserted into Decision Tree
    - Verify downloaded gesture can be recognized

8.6. Installation Testing

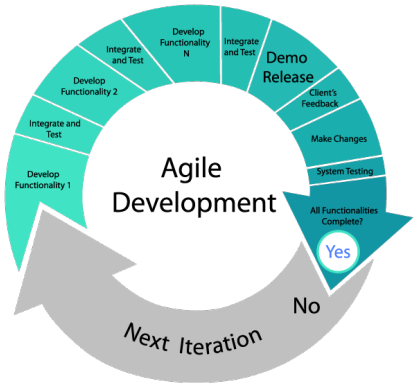
a. Verify that the program and its necessary components have been installed correctly

b. Verify that the project works from a clean install

8.7. Security Testing

a. Test the UI and ensure that it is not susceptible to threats
    - Sanitize user input
    - Ensure user information is private
    - Database is safe from common security threats

9. Project Management Plan



9.1. Development Strategy

Agile development strategy is chosen because it uses iterations or sprint that performed parallel across the entire lifecycle of the project. Agile is more flexible compare to spiral and waterfall. It allows developers to give direct response to any issue, while the sprint is still running. The team members experienced this strategy in software engineering class, and all feel comfortable using this strategy.

9.2. Development Timeline\Schedule

The table below is the rough development timeline. The Gantt chart is attached in the attachment page.

Time	Milestone Target
Summer Break	Recognize Gestures ● Integrate hard coded gestures as well as the commands that the gestures perform. ● Basic demo setup ● Gather gesture data
August-September	Train Gestures ● Implement Decision Tree Learning ● Create at least one gesture that is recognizable by leap motion that has been trained through Decision Tree Learning ● Default gestures implemented in the Decision Tree
September-October	Cursor Control / Input Commands ● Have trained gestures control the cursor / OS
October-November	GUI ● Create only the UI that will be able to do the following functions after the database is set: ○ Create an UI that allow the user to modify what gestures are linked with which actions, view the recognized gestures, and train new gestures. ○ By this UI the user will also be able to switch

	between saved profiles and create new profiles.
November-December	Database Storage / Retrieval ● Set up the database for gesture and command storage ● Connecting the GUI and the database
December	Finalizing the program ● Final Touch ● PRESENTATION!

9.3. Work Delegation

Delegates	Recognize Gestures	Train Gestures	Cursor Control / Input Commands	User Options / Gesture to Command Mapping	GUI
Alec	**X**	**X**		**X**	**X**
Cameron	**X**	**X**	**X**	**X**	
Dewi		**X**		**X**	**X**

10. Development Cost and Resource Needs

10.1. Team Funded / provided

Each group members own a laptop that meet the system and hardware requirements. Each team member designed to have different Operating System brand, which allow application testing on each OS. Also, in final submission, the team funded the flash drive and documentation printing costs.

10.2. School Funded / provided

The main development cost for this application is to get the leap motion controller, which cost $79.99 each. Four controllers are purchased by the University of Missouri on April 2017. Therefore, the total costs are $319.96.

11. Identification of Required Independent Learning

11.1. How to setup leap motion controller and actually make it works

Leap motion and its controller is new to all of team members. All members are hand in hand to do the initial setup, such as installation and netbeans set up. Leap motion installation is basic, similar to installing other hardwares. When the installation process is done, leap motion provides visualizer, where it actually visualize gestures performed above the controller. All the members are amazed and motivated to make something from this incredible technology in front of the team. However, how make the leap motions actually works is challenging. Many developer set up to do in the IDE, in this project the team use NetBeans as the IDE.

11.2. Decision tree Algorithm

All team members took Introduction to Machine Learning course in Spring 2017. That class introduce the team to decision tree algorithm. Based on the class and research, decision tree algorithm being considered because it has a high efficiency, and accuracy. However, the class taught decision tree in mathematical way instead of in computer science way. There was only a minor coding assignment, and was using matlab. Therefore, the team members, especially Cameron independently learn how to implement decision tree in Java.

Decision tree is the essential algorithm that the team use to register and recognize the gesture. During the development process, the team is brain stormed about which attributes values are really necessary to get the most efficient result. In the beginning phase, the tree is not self build, and does not give a great result. Then, the team develop the tree in more advanced way, where it can self build with its optimum height splits.

11.3. Using Github for group work integrations

Using github for group work integration is really effective for the group code exchange. However, during the development time, the team keep facing the same problem when pulling and pushing the code in github. Three members have different Operating System runs in their computers. When a member tries to pull from github, all the system and build preference from the last person who pushed is also got pulled, and it ruins everything. After some research, the team decided to create a new repository with correct git.ignore, which only allow to push and pull source folder.

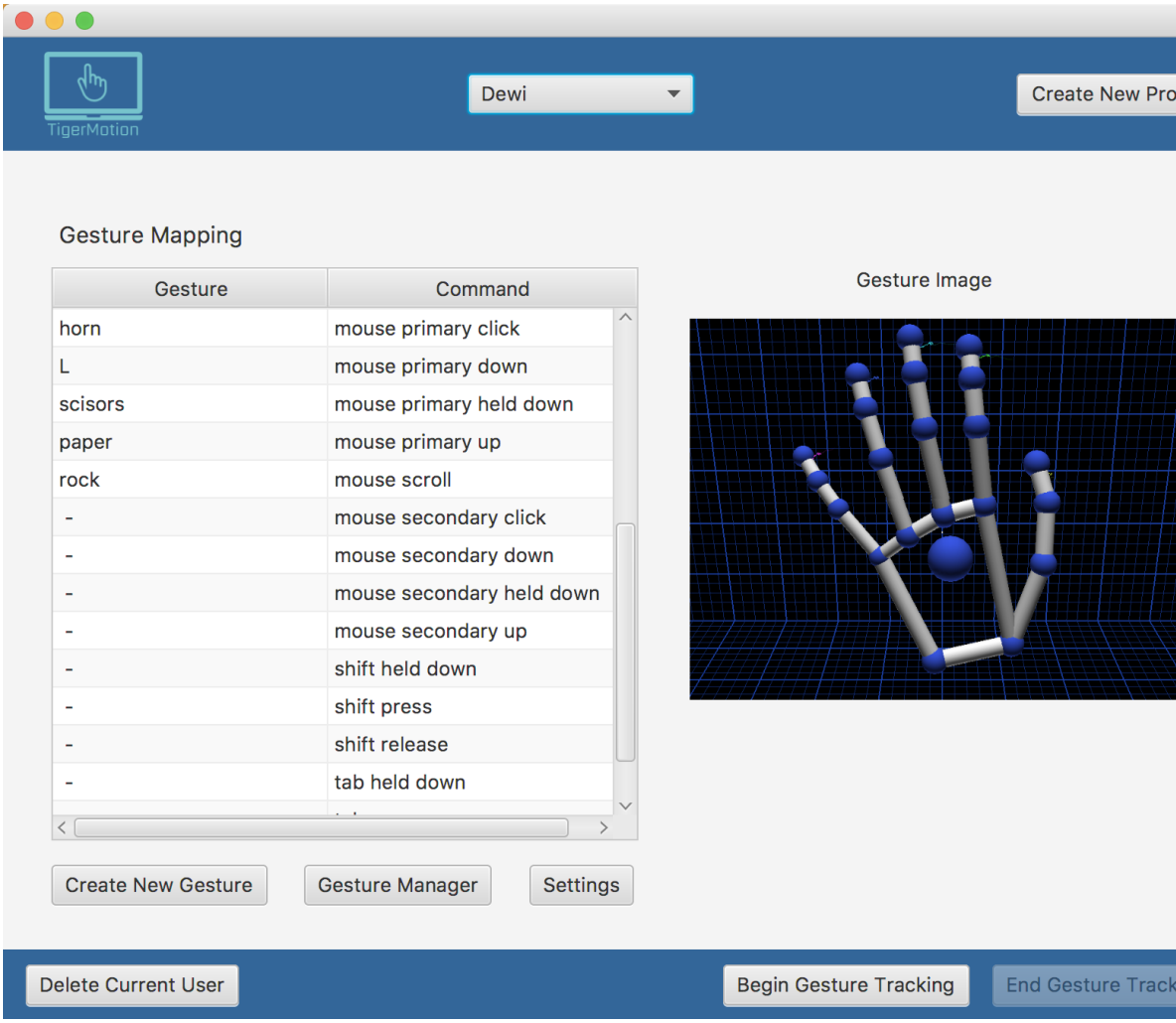
12. Result / Project Outcomes

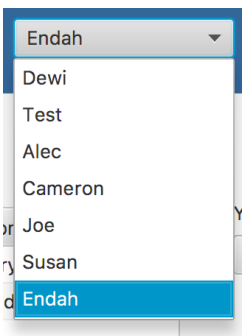
The final result of TigerMotion OS Controller application are explained in the following section

12.1. TigerMotion User Interface

12.1.1. *Main Page*

Main page is the navigation center of this application. Main page displayed group's logo, create new profile button, table view of gesture and commands mapping, create new gesture button, gesture manager button, settings, delete user, begin gesture tracking, and end gesture tracking. Each feature has an important role.

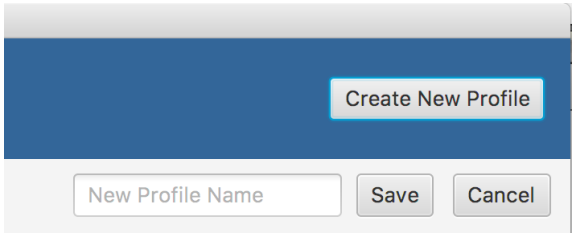




a. Profile list combobox

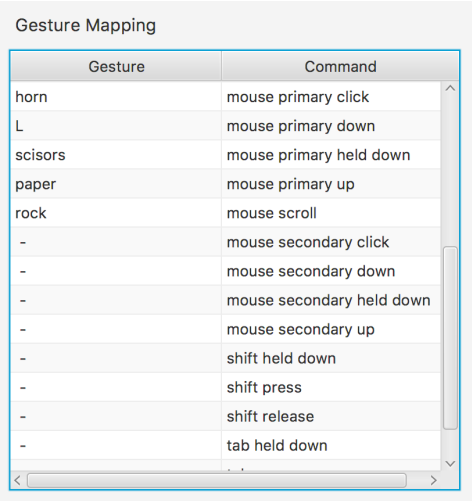
Profile combobox holds all profiles that has been created by the user. Each Profile has its own gesture saved and mapped command. This feature allows users to have liberty to choose which profile to select based on gesture and command mapping. When the user selected a profile, the table will be populated based on the information saved corresponding to that profile.

b. Create new profile button



When create new profile button is clicked, new profile attributes will appeared below the button. Text field to hold new profile name, save, and cancel button is shown. When the user clicked save, new profile will be created with empty gesture registered and list of all commands. This all information is saved in user's local computer in JSON file. This new profile will automatically become the selected user.

c. Gesture and Command Mapping Table View



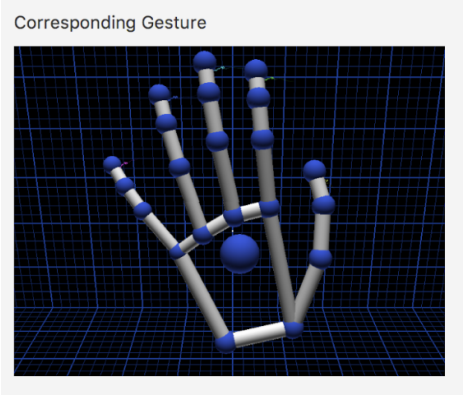
In the first column, all gestures created are listed. In this column, users have liberty to map each gesture to their desire commands. This mapping can be done by clicking a row of targeted command, then the drop down list will expands. Users can click to the selected gesture to assign the selected gesture. One gesture only can be mapped to one command. If the user try to assign the gesture into another command, the latest action will override the one.

In the second column, all the commands available are listed. They are static, and users cannot change or add the list. Following are all the list of commands and their functions.

- Backspace held down
- Backspace press
- Backspace release
- Caps Lock press
- Enter held down
- Enter release
- Enter press
- F10 press
- F11 press
- F12 press
- Mouse primary down,
- Mouse primary held down
- Mouse primary up
- Mouse primary click
- Mouse secondary down
- Mouse secondary held down
- Mouse secondary up
- Mouse secondary click
- Mouse scroll
- Shift held down
- Shift release
- Shift press
- Tab held down

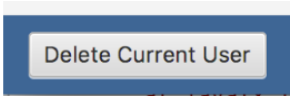
- Tab release
- Tab press

d. Image View for Selected Gesture on Tableview



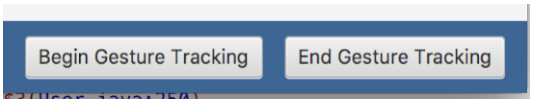
This section is designed to hold corresponding image captured from selected gesture. Therefore, the application can give the user information about which gesture is corresponding to the gesture name, in case the user forgot. For now, this feature has not been implemented yet.

e. Delete Current User Button



This button is for deleting the current selected user. User is important navigator in this program. All features have different behavior based on the selected user profile. When the user wants to delete a profile, the confirmation alert will pop up, to make sure that the action is not accidental. When the user click ok in the confirmation alert, the user will be deleted and current profile will goes to first profile on list.

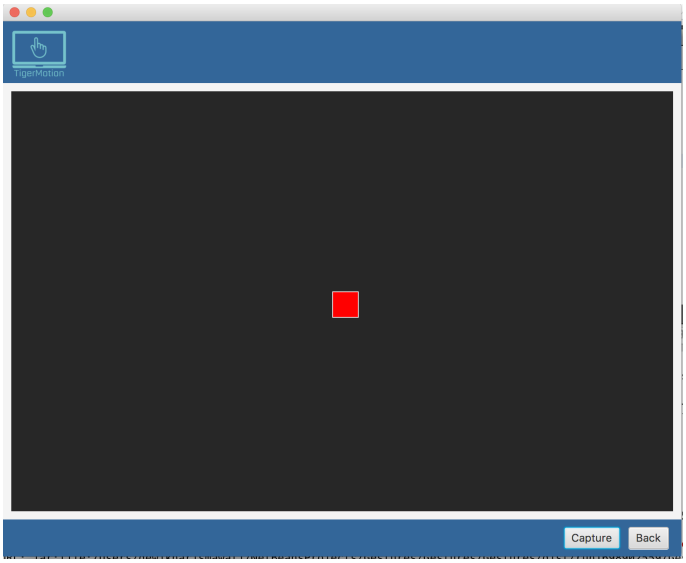
f. Navigation Buttons



This application has two navigation button. They are located in the bottom right of the main page. Begin gesture tracking to start the tracking. When it is clicked, it enables the gestures to perform it corresponding commands that has been set up in the tableview. When users want to stop the service, they can click end gesture tracking.

12.1.2. Create New Gesture Page

When create new gesture button is clicked, it will redirect the user to the create new gesture page as shown below.



Procedure to create a new gesture:

a. The user need to put her/his desire gesture above the leap motion controller

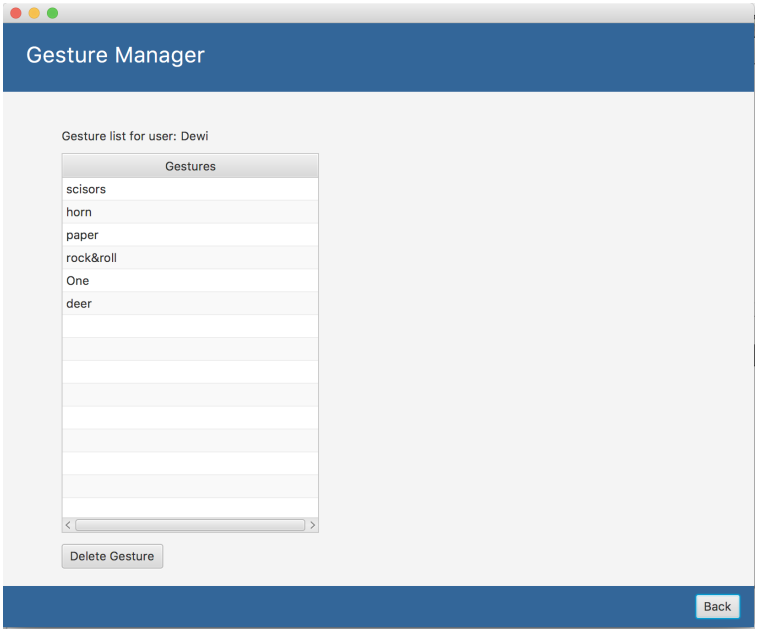
b. Then, click capture button to capture the gesture

c. Text field for the gesture name will pop up.

d. Click ok to save the new gesture.

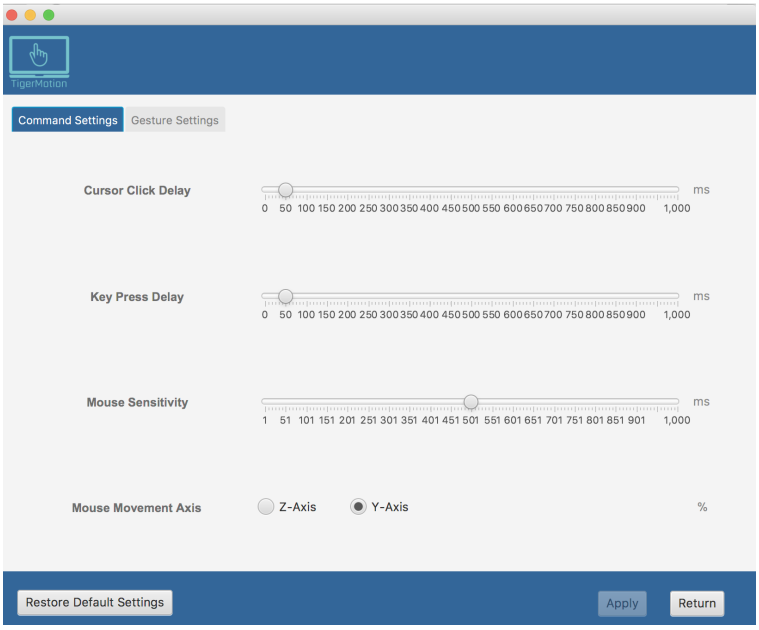
After a new gesture has been created, it will be visible in the gesture drop down list on the main page table view, and it is ready to use.

12.1.3. Gesture Manager Page



When gesture manager from main page is clicked, it will redirect to gesture manager page. In this page, users are able to see all the gesture created, and able to delete undesired gestures. To delete, select the gesture in the tableview and then click delete gesture button. After the action, gesture is deleted from the JSON file and from the tableview. Finally, back button is to navigate back to the main page.

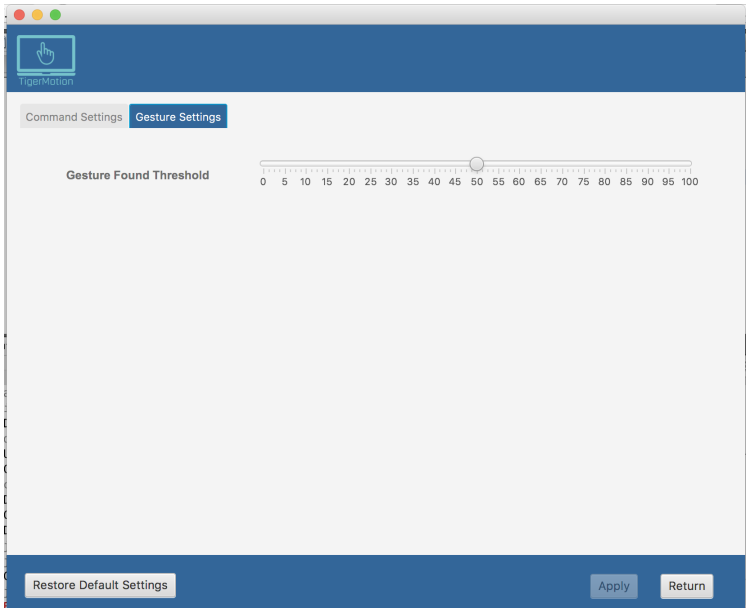
12.1.4. Settings Page



When user clicked settings button in the main page, it will redirect to the settings page, where the user can set customs delay and sensitivity of the commands. There are two tabs in settings page, first tab is to set the cursor delay, key press delay, and mouse sensitivity. Users can set their desire amount of delay and sensitivity

using the slider values. In addition, user also can set the mouse movement axis. Either in z-axis like what usual mousepad does, or in Y-axis parallel with the screen orientations.

In the second tabs, user can set the gesture found threshold. This setting is for gesture recognition accuracy. When gesture attribute is runs at the decision tree, there is possibility that the gesture performed does not match with all of the gesture list 100%. The application provides this feature because in test case, there are times where the gesture has been matches 90% to one of the gesture listed on the decision tree, but the remaining 10% does not match, and it turns out to gesture not found. Therefore, this feature allows a little flexibility on the process finding the gesture. But, the application recommends to have above 75% threshold to avoid decision tree confusion or duplication on recognizing gestures.



12.2. TigerMotion OS Controller Tutorial

The following steps are tutorials to use TigerMotion OS Controller with gestures.

1. First of all, make sure to have all the leap motions controller and the application installed properly.
2. Open the application. Initially, there is no profile and gestures registered. So, before enable the gesture controller, profile and gestures need to be created first.
3. Create new profile by clicking the “Create New Profile” button on top right of the page (it will show create new profile sections).
    a. Type the desired profile name
    b. Click save. This new profile will automatically become the selected profile
4. Next step, create new gestures by clicking “Create New Gesture” button on bottom left of the page (it will redirected into create new gesture page).
    a. Put desired hand and gesture above the controller. The performed gesture can be seen in the visualizer provided
    b. Click capture if ready
    c. The application will ask for the gesture name. Type the desired name, make sure the name has an association with the gesture registered. Or else, there will be a confusion.
    d. Click ok for naming.
    e. Repeat a-d to register several gestures
    f. Then click back to go to the main page

5. Now, it is time to maps the gesture into the commands.

a. To assigned a gesture to a desired commands. Click the gesture row for that desired commands.
b. Click it again to show the drop down choices.
c. Click the desired gesture name on the drop down. The gesture will automatically mapped to the commands
d. If the users change her/his mind, just do step a-c into the new desired commands. It will automatically override the old mapping.

6. Click “Settings” to set up preference delay, mouse movement axis, and gesture threshold. If setting is skipped, the application has a default values for each setting.
7. To enable the application, click “Begin Gesture Tracking” button. It will automatically enable the gesture tracking and commands the computer based on the gesture performed.
8. To delete user, click “Delete Selected User” button.
9. To delete undesired gesture, click on “Gesture Manager” button. It will redirect to the gesture manager page.
    a. To delete undesired gesture, click a gesture in the table view.
    b. Hit the “Delete Gesture” Button.
    c. Press “Back” to go back to the main page

13. Discussion

The project is implemented using intel i5 and i7 processors, with three different operating systems, which are Windows, UNIX, and MacOS. All the code is written using Java programming language, and the UI is written with JavaFX with scene builder. Testing data is conducted by 4 different people: Cameron, Alec, Dewi, Rifki. Each person tests and runs the application in different OS, and it is works normal in all of OSs.

After running the application on each person's computer. First step to do is creating a new user profile. Then, creating five same new gestures and give each gesture unique name. After some gestures created, subjects are back to main page and mapping the saved gesture into commands available as follow.

1. Scissors: mouse primary down
2. Rock: mouse primary up
3. Paper: mouse primary press
4. L shape: scroll
5. Horn: backspace

Then, all subjects set the setting in all same numbers. Finally, all subjects hits the Begin Gesture Tracking button, begin to perform the gesture above the controller, and try to close an application window and clicking the End Gesture Tracking button to end the tracking.

The overall experiments gives expected result with minor inaccuracy. Create new gesture works, it able to write the gestures nodes attributes on the decision tree and save it to desire profile's JSON file. The gesture performed is also recognized well by the decision tree. For developer, this tracking can be monitored with decision tree visualizer. This visualizer does not available for users because it does not necessary for the application to run. When users

performed the gesture, the visualizer will shows which nodes are match with the attribute from the streamed frame. Also, recognizing status can be monitored in the Netbeans output windows. The application is printing the gesture name if it recognizes a specific gesture. Otherwise, it does nothing. The application can print up to 60 frames in one seconds. That is why the application has delay settings to prevent too rapid actions in a period when the gesture is performed.

The team learns that each data attribute extracted from frames has an important part of gestures recognition. One time, palm_normal attribute is accidentally commented in an important place. The application could not recognize gestures at all.

In experiments, inaccuracy is still happening in some cases. It identifies that the major cause of inaccuracy in gesture recognition is mislabeled data from the leap motion. During the test, the team provides a decision tree visualizer, which enable the developers to know which gesture is performed, and whether gestures is recognized or not. The observation shows many instances in which the data provided does not mimic the gesture performed. This data collection evaluation is a little hard to do since the raw data from the controller are not publicly accessible. Therefore, the application has gesture recognition threshold, which it can tolerate inaccuracy of gestures recognition.

In conclusion, TigerMotion OS controller with gesture is ready to deploy. The expected functions and features on the application have been implemented. Based on the testing, application provides all service as expected, such as intuitive, gestures are recognized quickly, responsive, and does not make users feel frustrated when using it. However, there are some parts that needs to be improved, such as in recognition accuracy, more commands available, and cursor smoothness.

14. Future Work

14.1. Add more available commands to control computers

For now, the application provides 34 commands that can control computers. Those 34 commands including both keyboard and mouse commands. The team hopes to provide more commands, especially the most common commands. For instance copy, cut, close window, paste, delete, undo, and redo.

14.2. Provide gesture visuals

In the future, when the application has more commands available, there is a big chance of user confusion about which gesture mapped to which command. Therefore, the group hopes to have a visualizer saved as an image when the user is capturing her/his gesture when registering a new gesture. The idea actually has been presented in the main page. There is a section to placed corresponding gesture image. For now, this feature does not work yet.

The plan is that the application can give the user information about which gesture is corresponding to the gesture name. This will be an interactive section, where when the user is clicking gesture name on the table view, it will display the corresponding gesture image captured by the visualizer when it is created.

14.3. Provide more settings for cursors smoothness

The performance of this whole application can be evaluate from how smooth the gesture can control the computer cursor. For now, the cursors smoothness in the application is still average.

15. References

0. J. S. Artal-Sevil, J. L. Montañés, (2016). Development of a robotic arm and implementation of a control strategy for gesture recognition through Leap Motion device, *Technologies Applied to Electronics Teaching (TAEE) 2016*, pp. 1-9. Retrieved from http://ieeexplore.ieee.org/document/7015190/citations

1. C Sharp (programming language), https://en.wikipedia.org/wiki/C_Sharp_(programming_language)
2. JavaScript, https://en.wikipedia.org/wiki/JavaScript
3. Unreal Engine, https://en.wikipedia.org/wiki/Unreal_Engine
4. Objective C, https://en.wikipedia.org/wiki/Objective-C
5. C++, https://en.wikipedia.org/wiki/C%2B%2B
6. Java (programming language), https://en.wikipedia.org/wiki/Java_(programming_language)
7. Python, https://en.wikipedia.org/wiki/Python_(programming_language)
8. Unity (game engine), https://en.wikipedia.org/wiki/Unity_(game_engine)
9. Decision Tree, https://en.wikipedia.org/wiki/Decision_tree
10. Quadratic Classifier, https://en.wikipedia.org/wiki/Quadratic_classifier
11. Decision Tree Learning, https://en.wikipedia.org/wiki/Decision_tree_learning
12. Linear Discriminant, https://en.wikipedia.org/wiki/Linear_discriminant_analysis
13. Reinforcement Learning, https://en.wikipedia.org/wiki/Reinforcement_learning
14. Unsupervised Learning, https://en.wikipedia.org/wiki/Unsupervised_learning
15. Supervised Learning, https://en.wikipedia.org/wiki/Supervised_learning
16. Deep Learning, https://en.wikipedia.org/wiki/Deep_learning
17. Artificial Neural Network, https://en.wikipedia.org/wiki/Artificial_neural_network

18. Pattern Recognition, https://en.wikipedia.org/wiki/Pattern_recognition

19. itersnews. (2014). 3D gesture-sensing technologies in smart devices poised for double-digit worth. Retrieved from http://itersnews.com/?p=64443

20. Weichert, F., Bachmann, D., Rudak, B., & Fisseler, D. (2013). Analysis of the accuracy and robustness of the leap motion controller. Sensors, 13(5), 6380-6393. Retrieved from http://www.mdpi.com/1424-8220/13/5/6380/htm

21. Leap Motion Developer. (2015). System Architecture. Retrieved from https://developer.leapmotion.com/documentation/csharp/devguide/Leap_Architecture.html

22. Leap Motion Developer. (2015). Controller Policies. https://developer.leapmotion.com/documentation/cpp/devguide/Leap_Policies.html

23. Khan, F. R., Ong, H. F., & Bahar, N. (2016). A Sign Language to Text Converter Using Leap Motion. International Journal on Advanced Science, Engineering and Information Technology, 6(6), 1089-1095. Retrieved from http://insightsociety.org/ojaseit/index.php/ijaseit/article/view/1252

16. Appendices

16.1. Team member Biographies

Cameron Barker

Cameron Barker was born and raised in California. He excelled in school and played high school tennis, making it to the number two singles spot on the team. Cameron graduated from high school in 2006. After working for his dad's roofing company that summer, Cameron attended Brigham Young University. Having freedom away from home and being introduced to World of Warcraft caused him to withdraw from BYU during the first semester. Cameron half-heartedly searched for jobs while attending a local community college and taking easy classes like volleyball, tennis, piano, and accounting. His brother convinced him to join the military. Cameron and Trevor entered the Navy in 2009. In 2014, after five years of service, Cameron earned his G.I. Bill through the VA allowing him to attend school full time. He moved to Missouri to be with his now wife and enrolled at Mizzou.

While attending Mizzou Cameron has learned much and grown in leaps and bounds, inside and outside of class. Deciding to take advantage of the VA's payment methods, he took on a full course load of eighteen credits each semester. The first year consisted of many prerequisites and general education classes. The next two years were nearly entirely computer science related. During this time Cameron had Lasik eye surgery, ran a half marathon, bought two houses, married his wife, and had a daughter. He applied for an internship at Socket, a local telecom company, in January 2016 and earned the position. Cameron worked there for a year and a half, learning how to use PHP, MySQL, JavaScript, HTML, and CSS in a production environment. In October 2017 he applied for a full-time position at Quarkworks, a mobile application development studio in Columbia. He got the job and has been working part-time for

a little over a month. When the semester is over and Cameron graduates he will work for Quarkworks full-time.

Alec Knight

My name is Alec Knight and I am currently a senior at the University of Missouri Columbia. I am a computer science major and have been going to Mizzou for the past four years. I am currently living in Columbia, Missouri but I am originally from a town about an hour away from St.Louis called O'Fallon, Missouri. I am a high school graduate and attended a Catholic high school in O'Fallon called St.Dominic's High School. Before that I attended a Catholic school named St.Joseph's in Cottleville, Missouri.

I began attending St.Dominic's High School in 2009 and went their for four years. I graduated in May of 2013 and was in a graduating class of about two hundred people. During high school I was on the track and field team and played football as well. I played football for three years at St.Dominic's and was only on the track team for one year. During my senior year I stopped playing football for the most part because I had found a job and started working. I decided that I wanted to go to the University of Columbia Missouri my junior year and started to take tours of the campus in the summer. During my senior year I applied and was accepted into the University of Missouri Columbia. My freshmen year I lived in the college avenue dorms on campus. The next year I moved into a house in Columbia.

Before I attended Mizzou I decided that I wanted to major in electrical engineering and so I started taking classes for electrical engineering my freshmen year. During the second semester of my freshmen year I decided that electrical engineering was not the career I wanted and so I switched to computer science the next semester.

Dewi Endah Kharismawati

Dewi was born and raised in Indonesia. She travelled 13,000 miles away from home to study computer science in a better school environment. Studying computer science is tough for her. However, she tried her best during her study at Mizzou, and finally will be graduated on Fall 2017. Dewi is graduated from Sampoerna Academy Boarding School, and continued her study in Sampoerna Community College before transferring to Mizzou.

During her study at Mizzou, she learn several programming language, such as Java and C. She also learn how to use PHP, MySQL, JavaScript, HTML, and CSS in internet class. After developing this capstone project, Dewi has an interest to take CS concentration in Digital Image Processing. Dewi is an active student at Mizzou. In her first semester, she joined as the board member of Missouri International Student Council. 2 years serving she was promoted to be the publicity team Manager, the Sponsorship Coordinator, and the Director of the Publicity team. She is happy to be part of international community and introducing Indonesian cultures to Mizzou Community. In addition, she also a part of Upsilon Pi Epsilon since Fall 2016. In Fall 2017, she served as the Historian of the organization.

Dewi loves travels. During her 2.5 years study at Mizzou, she has been travelled to 20 states and explored each state's attractions. When she was young, she has a dream to visit the statue of liberty, and she visited the statue in the beginning of 2017.

16.2. Failure and Solutions

- Recognition system does not adequately distinguish between individual gestures.
  - Only allow significantly different gestures
  - Use a different method to differentiate gestures
- System performs too slowly to allow the user a responsive experience.
  - Optimize and manage code better
- Getting Leap Motion to behave with each different Operating System.
  - Download different leap motion driver on different leap motion
    - V2 for UNIX
    - Orion for Windows and MacOS
  - Research more how to make it works on all OS
  - Setting the VM option in Netbeans
- Pushing and Pulling in github gave the same problem over and over again. It is ruins the settings on each computer because it is pulling the setting from the last person pushed to the repository
  - Create a new repository with a correct settings on git.ignore file
- Machine Learning Method too strict for gesture recognition
  - Create a threshold for gesture recognition for more flexibility
- Using database is not really necessary for this application
  - use JSON file to store user gestures and configurations