1

Korean Adapted Sign-language Educator (KASLE) (be a friend with hearing impaired people)

Song Young Iek, Lee Sang Hyun, Lee Sang Hyun, and Yoon Kyoung Taek

Abstract—This electronic document is a description of Educator for Korean Sign-language operated by leap-motion device. In nowadays concern for disabled has been increased. Therefore lots of people wants learn sign language which can be bridge between handicapped and un-handicapped. However, learning sign language can be troublesome. It can cost lots of money and time. To overcome this problem, there are lots of program to help people learn sign language. Problem is that these programs are mostly English, and does not support Korean. Lots of people believe that sign language is same for all around the world, but that is not true. Korean and English in terms of sign language are very different. So, we invented KASLE, which means Korean Adapted Sign-Language Education for beginner. This program uses hardware called Leap-motion which recognizes hand motions. It will compare hand motion and data in the store to guess it is correct. One of the key features of our program is machine learning. By using famous machine learning language SVM, we are going to teach program how to understand this motion. Also, we are going to add some game to help user have fun during study. There are functionality that user can add or delete word in program by using gestures. This function will help user contribute to our program. Our program will help people who use Korean to learn sign language efficiently.

Index Terms—korean sign-language, leap-motion, leap-motion API, google translator API, Java, Eclipse, Educator, Social Network System(SNS)

1 Introduction

The Internet of Things generation has come. And development of IT has tremendous impact on various social area and benefit daily human life. Especially, efforts to break wall between the un-handicapped and the handicapped are continued until these days. As one of that kinds of efforts, the most frequently used way to communicate with deaf is sign language. Some people who are not handicapped also want to learn sign language. However, lots of people who want to study sign language realize that sign language is very difficult. Even if they want to practice sign language there are not enough ways to practice it. Of course, there are some ways to be educated by associated institution or professional educator. However it cost too much in time and money. Also, there are some programs for educating sign language in English, but there are not many programs for Korean. To overcome these problems we made program that people can see, imitate and correcting sign language while learning. This program was made by using LeapMotion which recognize fingers motion.

2 Requirement

2.1 Recognizing hand movement

2.1.1 Get hand data Using LeapMotion hardware

In LeapMotion, there is a function that collects users hand data. So by using this function we can see what kind of data is implemented about movement. We can use this data to use other thing, for example as recognize gesture or determine if gesture is correct or not.
Input Person's Hand gesture.
Implementation process : While user is gesturing, Leap motion captures its shape or movement and recognizes it as corresponding gesture. Leap motion SDK converts data into user understandable form and show it to us.
Output :Hand data

2.1.2 Using data from LeapMotion and machine learning algorithm, program learn about sign-language gestures.

Nowadays machine learning has become most important thing in software. So we are going to use machine learning to improve our accuracy. by using SVM we can teach our program a machine learning and it can improve our programs accuracy.
Input: Person's Hand data
Implementation process: Analyzing given sets of hand data, SVM(machine learning algorithm) learns rules that separate each gesture. So, algorithm itself performs classification on hand data comparing with existing rule based on given past data set. Output: SVM algorithm with gesture classification rule.
Implementation process : While user is gesturing, Leap motion captures its shape or movement and recognizes it as corresponding gesture. Leap motion SDK converts data into user understandable form and show it to us.
Output :Hand data

- .

2.1.3 From the data that program learned, program recognize sign language from current hand movement.

LeapMotion can collect hand data. So, using our LeapMotion SDK we can add gesture. by comparing gesture data and user hand movement, program can determine whether user is doing correct gesture or not. Also, by using machine learning program can evolve our program so it can recognize sign language better.
Input : Person's Hand gesture
Implementation process: For each gesture, Leap motion hardware classifies and recognizes each gesture using SVM algorithm generated by past hand data set. In other words, machine learning algorithm determines what kind of gesture this is. Leap motion SDK receive the data from
Leap motion hardware and processed by SVM algorithm. And it converts the raw data into user-understandable data and shows to us.
Output : Recognized Sign language and its result in user understandable form.

2.2 Functions for sign-language educator.

2.2.1 Recognizing basic vowel for sign-language.

We input a basic vowel for basic sign-language communication. User can use vowel for using unregistered word. It is a kind of finger language. Frankly we have a weak point that can't realize all sign-language because sign-language sometimes use not only all part of arm but also chest so there is hardware-wise problem. For the sake of overcoming this problem, we add this requirement.
Input Person's Hand gesture(vowel)
Implementation process:
(First time) We tell program that current hand data set represents specific vowel. And the program stores that data set into program.
(After specific vowel is stored to program) Program recognizes that users gesture represents vowel.
Output Program data represents vowel.

2.2.2 Recognizing basic consonant for sign-language.

We input a basic consonant for basic sign-language communication. User can use consonant for using unregistered word. it is a kind of finger language. Frankly we have a weak point that can't realize all sign-language because sign-language sometimes use not only all part of arm but also chest so there is hardware-wise problem. For the sake of overcoming this problem, we add this requirement.
Input Person's Hand Gesture(consonant)
Implementation process:
(First time) We tell program that current hand data set represents specific consonant. And the program stores that data set into program.
(After specific consonant is stored to program) Program recognizes that users gesture represents consonant.
Output Program data that represents consonant.

2.2.3 Making a complete character using registered vowel and consonant.

This requirement is associated requirement number 2 and 3. We already explain why we add default vowel and consonant. Consequently requirement number 2 and 3 is for this requirement. If user want to communicate others people by finger language, user must be able to make complete letter so we offer this function.
Input Program's data represents consonant and vowel.
Implementation process:
Program combines vowel and consonant into character and makes word with characters.
(After specific consonant is stored to program) Program recognizes that users gesture represents consonant.
Output Program data(word).

2.2.4 Registering the basic words for sign-language.

We offer basic sign-language words. For user convenience, we offer various built in words, user can use it without data entry. Because of our hardware-wise problem we support only basic (not that many) words for user.
Implementation process: Basic words sign
language recognition features is provided by program. It is implemented by data entry by developers during developing process.
Output: Built-in words that are not needed to be entered by user.

2.2.5 Practice game for letters and words.

Humans are interested in a kind of games and have more concentration when they're plying game so we offer a simple game for practicing sign-language it maybe shows some word question by Korean and user makes correct sign-language in response to question then game shows correct sign or incorrect sign.
Input: Person's Hand gesture
lImplementation process : Program recognizes users gesture and compare with current questions sign language gesture in game. It also shows whether users answer is correct or not.
Output: Sign language game and result page with rate of questions to which user answers correctly and questions provided by program

2.2.6 Extracting entered words to text file.

It is important to extracting sentence user want to say because our program maybe offer various function that bases on extracting text file. In other word this function is basic condition for other functions.
Input: Entered character(s) or word(s).
lImplementation process : Program recognizes each word and convert to and save in text file.
Output: Text file

2.2.7 Posting text entered by hand movement to Social Network Service.

In nowadays Social Network Service has become one of most important thing in todays internet. By transferring text data to Social Network Service, we can satisfy demand who wants share their work to Social Network Service.
Input: Entered word
lImplementation process : By selecting SNS to which user want to upload the text and pressing button, the data is transferred to certain SNS site and shared on site. That is,

user can unload text entered as gesture through leap motion hardware to Facebook.
Output: Uploaded text on SNS

2.2.8 Translation to another language using google translation API.

User can obtain text so we offer some function using and processing text file. If you want to translate a Korean to another language, enter Google translator and enter your text. Above stages are too annoying. So we offer a translation feature using Google translator API.
Input: Entered word
lImplementation process : By pressing Translate , translated message is provided to user.
Output: Translated message.

2.2.9 User can register the new words.

We add function that user can add new words by saving gesture to the program. Leap motion can recognize our gesture so user can easily gesture and save gesture to the program.
Input: Users hand gesture.
lImplementation process : While user is gesturing, Leap motion captures its shape or movement and analyzes it and learns what that means by matching the hand data set and users input that tells what certain gesture means to program. Once it is correctly learned by program, it is saved and can be retrieved later.
Output: User defined data (hand data and meaning data corresponds to certain hand data)

2.2.10 User can delete a registered word that not correct.

We added function that user can register word. However this approach can cause incorrect register in program. So, we added function that user can delete and fix registered word to overcome this problem.
Input: Text data.
lImplementation process : By using function that is created based on Leap motion SDK, user can delete words which were existed in program,.
Output: Data deletion.

2.3 Condition for program.

2.3.1 Convenient intuitive user interface.

Users can operate our program easily without complicate explanation. They can recognize what they can do on our program. And they can easily find out how to run all the features. All of features are visualized to be found easily and understood directly.
For implementing this requirement developers will use Java Play Framework. By using it, we can create UI which fits well with program's overall design. As a conclusion, UI should be comfortable for user to use program.

2.3.2 Separating user mode by normal user and manager.

In our program only manager can add the sign language. So we need to prevent users to add sign language because it can create confusion. To separate user mode and manager mode our program has the administrator login so only manager can become administrator. In administrator mode, we can add sign language.
Input: ID and Password.
lImplementation process : User will input id and password in the login section. Program determines whether current user is administrator or not. If the user is administrator, program will grant user more authority to him(or her).
Output: User authority level.

2.3.3 Constructing a simple graphic user interface(GUI).

Lots of current program is using GUI, because it is user friendly. Of course, we can build Also, most of current users are familiar with GUI not with CLI. So building simple GUI is essential. We are going to build GUI by using OpenGL. In this GUI there will be selection of our functions, for example where users can practice sign language and where they can test their ability.

2.3.4 Adding the explanation file (ex-Readme) for unexperienced computing user.

Our program is focusing on the people who are trying to learn sign language, not computer expert. So, there is a high possibility that some people are not good at using computers. Therefore we need explanation for un-experienced users so they can use our program in efficient and effective way. We will put detailed information about program, and also contact information for more questions.

2.3.5 Existing communication channel for repair and additional function between user and administer.

Our program may have some issues or bugs in program. Also user might have some idea that can improve our program. So we are going to notify user by making official SNS or email or contact number. Users can see our contact list in the program. Also if same questions were asked many times, we will add that list to our FAQ (frequently asked question).

2.4 Team Roles

Roles	Name	Task description and ETC
User	Lee Sang Hyun	Inspect convenience of program and tell what is inconvenient point.
Customer	Lee Joo Hyun	Inspect a cost of program for the development and determine whether functions of program can be easily understood and used easily
Software developer	Song Young Iek	Inspect whether program has inefficient design and whether it satisfies requirement of user
Development manager	Yoon Kyoung Taek	Inspect whether overall developing process is properly operated and whether program has reasonable development cost and waste money for useless things

3 Development Environment

3.1 Choice of software development platform

3.1.1 System Platform(e.g., Windows, Linux, Web, or etc.)

We chose Windows platform. All the developers have development environment which runs on Windows system. By choosing Windows system, we can save time and effort to adapt different development platform and focus on development itself.

3.1.2 Programming Language

Leap motion supports several (not that many) programming languages for developer. Among those options, developers chose Java. Because Java is high-level language, so developers dont have to care about annoying physical and low-level issues. Also, it is easy to fine open source code written on Java, because of its popularity.

3.1.3 Cost Estimation

We spent 100 dollars to buy Leap Motion device. Except that, we will use hardware and software that were already equipped. So, our total cost will be 100 dollars.

3.1.4 Clear Information of Development Environment

Windows 10
Github
Eclipse (Mars.2 Release (4.5.2))
Trello (ver. 3.6.0.1480)
Java Play Framework (2.5.2)

3.2 Software in Use

Open GL JOGL (Java OpenGL) SVM machine learning algorithm Java Play Framework Trello Github

3.3 Task Distribution

Song Young Iek is responsible for SVM machine learning and hand data processing modules. Yoon Kyung Tek is responsible for Login module and all of DB stuffs. Lee Sang Hyun is responsible for GUI module, Quiz module and other applications modules. Lee Joo Hyun is responsible for Quiz module and GUI module.

4 Specifications

4.1 A. Recognizing hand movement

4.1.1 Get hand data Using LeapMotion hardware

In LeapMotion, there is a function that collects users hand data. So by using this function we can see what kind of data is implemented about movement. We can use this data to use other thing, for example as recognize gesture or determine if gesture is correct or not. Sign language has two categories. One is real sign language which contains hand movement and represents one or more word(s) with meaning. Second one is simple finger language without hand movement and represents single vowel or consonant. Each category of sign language has separate way to be represented by LeapMotion.
Input: Person's Hand gesture.
Implementation process: While user is gesturing, Leap motion captures its shape or movement and recognizes it as corresponding gesture. Leap motion SDK converts data into user understandable form and show it to us.
Output: Hand data
Source code 1) Get hand data which represents real sign language (dynamic hand data which contains movement)
Source code 2) Get hand data which represents simple finger language (static hand data without hand movement)

4.1.2 Using data from LeapMotion and machine learning algorithm, program learn about sign-language gestures.

Nowadays machine learning has become most important thing in software. So we are going to use machine learning to improve our accuracy. by using SVM we can teach our program a machine learning and it can improve our programs accuracy.
Input: Person's Hand data
Implementation process: Analyzing given sets of hand data, SVM(machine learning algorithm) learns rules that separate each gesture. So, algorithm itself performs classification on hand data comparing with existing rule based on given past data set.
Output: SVM algorithm with gesture classification rule.

4.1.3 From the data that program learned, program recognize sign language from current hand movement.

LeapMotion can collect hand data. So, using our LeapMotion SDK we can add gesture. By comparing gesture data and user hand movement, program can determine whether user is doing correct gesture or not. Also, by using machine learning program can evolve our program so it can recognize sign language better.
Input:Person's Hand gesture
Implementation process: For each gesture, Leap motion hardware classifies and recognizes each gesture using SVM algorithm generated by past hand data set. In other words, machine learning algorithm determines what kind of gesture this is. Leap motion SDK receive the data from Leap motion hardware and processed by SVM algorithm. And it converts the raw data into user understandable data and shows to us.
Output: Recognized Sign language and its result in user understandable form.

4.2 Functions for sign-language educator.

4.2.1 Recognizing basic vowel for sign-language.

We input a basic vowel for basic sign-language communication. User can use vowel for using unregistered word. it is a kind of finger language. Frankly we have a weak point that can't realize all sign-language because sign-language sometimes use not only all part of arm but also chest so there is hardware-wise problem. For the sake of overcoming this problem, we add this requirement.
Input :Person's Hand gesture(vowel)
Implementation process:
(First time) We tell program that current hand data set represents specific vowel. And the program stores that data set into program.
(After specific vowel is stored to program) Program recognizes that users gesture represents vowel.
Output Program data represents vowel.
// receive current hand data from leap-motion // and compare with initial hand data samples using SVM model // then, SVM returns category number which is most similar to current hand data

4.2.2 Recognizing basic consonant for sign-language.

We input a basic consonant for basic sign-language communication. User can use consonant for using unregistered word. it is a kind of finger language. Frankly we have a weak point that can't realize all sign-language because sign-language sometimes use not only all part of arm but also chest so there is hardware-wise problem. For the sake of overcoming this problem, we add this requirement.
Input: Person's Hand Gesture(consonant)
Implementation process:
(First time) We tell program that current hand data set represents specific consonant. And the program stores that data set into program.
(After specific vowel is stored to program) Program recognizes that users gesture represents consonant.
Output: Program data that represents consonant.

4.2.3 User can practice vowels and consonants of sign language.

User can practice sign-language it maybe shows some vowel or consonants question and corresponding answer User can follow answer of current sign-language question. If users answer is correct then proceed to next question.
Input: Person's Hand gesture
Implementation process: Program recognizes users gesture and compare with current questions sign language gesture. If users answer is correct then proceed to next question.
Output: Sign language practice

4.2.4 User can change mode between vowels-only, consonants-only, vowel s and consonants in practice of vowels and consonants.

There are three modes in vowels and consonants game, which are vowels-only game mode, consonants-only game mode, vowel s and consonants practice mode. User can easily switch between modes using intuitive GUI. User can switch mode at any time. (ex. at the beginning of practice, in the middle of practice, after finishing practice)
Input: Users selection of mode.
Implementation process: recognize users selection and switch mode using event listener and terminate current practice and start new practice of new mode.
Output: new practice of new mode implementation.

4.2.5 User can play game for vowels and consonants of sign language.

Humans are interested in a kind of games and have more concentration when they're playing game so we offer a simple game for practicing sign-language it maybe shows some word question by Korean and user makes correct sign-language in response to question then game shows correct sign or incorrect sign. Score which represents users performance on game is shown in real time.
Input Person's Hand gesture
Implementation process Program recognizes users gesture and compare with current questions sign language gesture in game. It also shows whether users answer is correct or not.
Output Sign language game and result page with rate of questions to which user answers correctly and questions provided by program

4.2.6 User can change mode between vowels-only, consonants-only, vowels and consonants in game of vowels and consonants.

There are three modes in vowels and consonants game, which are vowels-only game mode, consonants-only game mode, vowel s and consonants game mode. User can easily switch between modes using intuitive GUI. User can switch mode at any time. (ex. at the beginning of game, in the middle of playing game, after playing game)
Input : Users selection of mode.
Implementation process: recognize users selection and switch mode using event listener and terminate current game and start new game mode.
Output: new game mode implementation.

4.2.7 User can practice for words.

User can practice sign-language it maybe shows some word question and corresponding answer User can follow answer of current sign-language question. If users answer is correct then proceed to next question.
Input: Person's Hand gesture
Implementation process:Program recognizes users gesture and compare with current questions sign language gesture. If users answer is correct then proceed to next question.
Output: Sign language practice

4.2.8 User can play game for words.

We offer a simple game for practicing sign-language it maybe shows some word question by Korean and user makes correct sign-language in response to question then game shows correct sign or incorrect sign. Score which represents users performance on game is shown in real time.
Input : Person's Hand gesture
Implementation process: Program recognizes users gesture and compare with current questions sign language gesture in game. It also shows whether users answer is correct or not.
Output: Sign language game and result page with rate of questions to which user answers correctly and questions provided by program

4.3 Condition for program.

4.3.1 Convenient intuitive user interface.

Users can operate our program easily without complicate explanation. They can recognize what they can do on our program. And they can easily find out how to run all the features. All of features are visualized to be found easily and understood directly. For implementing this requirement developers will use Java Play Framework. By using it, we can create UI which fits well with program's overall design. As a conclusion, UI should be comfortable for user to use program.

4.3.2 User can login.

User can login with id and password which are set during sign in process.
Input: ID and Password
Implementation process: User will input id and password in the login section. Program determines whether current user is valid or not. If the user id and password are valid, program will grant user authority to him(or her).
Output: User mode is applied. (Without login, visitor mode is applied.)

4.3.3 User can easily restart current menu at any time.

There is a button at the top right side of screen of any menu. If user click the button, current menu is re-implemented.
Input:Users click
Implementation process: Program recognize users click on the restart button and restarts current menu.
Output:menu re-implementation.

4.3.4 User can easily get back to main menu at any time.

There is a go to main menu button at the top right side of screen of any menu. If user click the button, current menu is terminated and main page is shown. Input: Users click
Implementation process:Program recognize users click on the go back to main menu button and go to main page.
Output: Main page is shown.

4.3.5 User can see his(or her) accumulated historical information on KASLE program.

User can see his (or her) accumulated historical information on KASLE program. Historical data of current user on KASLE program contains name, gender, age and a few kinds of statistical information like average of game scores until then, how many practices youve ever done, etc.)
Input: ID and Password
Implementation process: User will input id and password in the login section. Program determines whether current user is valid or not. If the user id and password are valid, program will grant user authority to him(or her).
Output: User mode is applied. (Without login, visitor mode is applied.)

4.3.6 Separating user mode by normal user and visitor.

In our program, we has two kind of user mode, which are normal user mode and visitor mode. To apply normal user mode, you need to sign-in first. You can sign-in anytime you want. There is a button on main page and it enables users to access to sign in page.
Input: ID and Password
Implementation process: User will input id and password in the login section. Program determines whether current user is administrator or not. If the user is administrator, program will grant user more authority to him(or her).
Output: User authority level

4.3.7 Constructing a simple graphic user interface(GUI).

Lots of current program is using GUI, because it is user friendly. Of course, we can build Also, most of current users are familiar with GUI not with CLI. So building simple GUI is essential. We are going to build GUI by using Jigloo, SwingWorker and OpenGL. In this GUI there will be selection of our functions, for example where users can practice sign language and where they can test their ability.

4.3.8 Adding the explanation file (ex-Readme) for unexperienced computing user.

Our program is focusing on the people who are trying to learn sign language, not computer expert. So, there is a high possibility that some people are not good at using computers. Therefore we need explanation for un-experienced users so they can use our program in efficient and effective way. We will put detailed information about program, and also contact information for more questions.
This requirement doesnt need source code.

4.3.9 Existing communication channel for repair and additional function between user and administer.

Our program may have some issues or bugs in program. Also user might have some idea that can improve our program. So we are going to notify user by making official SNS or email or contact number. Users can see our contact list in the program. Also if same questions were asked many times, we will add that list to our FAQ (frequently asked question).
This requirement doesnt need source code.

4.4 Architecture Design and Implementation

4.4.1 Overall Architecture

We will show Overall Architecture in Last Page

4.4.2 Directory Organization

Directory	File Names	Module Names in use Etc.
/project/src/gui	mainframe.java mainpage.java settingpage.java	GUI Module
/project/src/login	MemberDAO.java MemberDTO.java MemberPROC.java MemberList.java UserLogin.java	Login Module
/project/src/signlistener	SignListerner.java	Leap Motion Device Control Module
/project/src/svm	asd.model	SVM Machine Learning Module
/project/src/ signlistener	SignListerner.java	Hand Data Processing Module
/project/src/practice	basic_prac.java	Sign Language Vowels and Consonants Practice Module
/project/src/practice	word_prac.java	Sign Language Words Practice Module
/project/src/quiz	basic_quiz.java	Sign Language Vowels and Consonants Quiz Module
/project/src/quiz	word_quiz_easy.java word_quiz_hard.java	Sign Language Words Quiz Module
/project/src/apps	other_apps.java	Other Applications Module

Fig. 1

4.4.3 GUI Module

KASLE has GUI environment. All the GUI stuffs are managed by this module. All other features operated by all other modules are operated based on GUI module. GUI module starts and controls overall process of program.
Class : class mainframe, class mainpage, class basic_prac, class word_prac, class basic_quiz, class word_quizeasy, class word_quizhard, class setting_page

Picture of mainpage

4.4.4 Login Module

KASLE has 2 modes, login user mode and visitor mode. If user wants register id, user can register id from program site. User can login to the program, user will get the statistics of quiz data. If current user doesn't log in, visitior mode is applied.
All other modules are operated based on the mode applided by login module
Class : class MemberDAO, class MemberDTO, class MemberPROC, class MemberList, class UserLogin

Picture of login

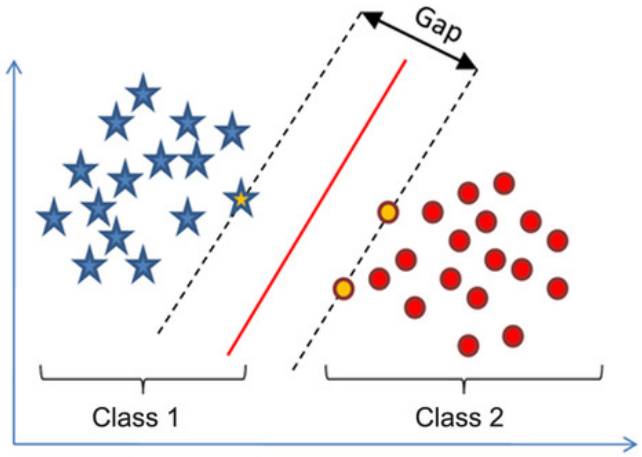
4.4.5 Leap Motion Device Control Module

Leap Motion device shoots a video of hand (30 frames per seconds) and analyze this 2D picture and convert into data set that represents 3D hand status. These detailed analyzing and converting process is handled by Leap Motion device. Current hand status is represented by various variables of various forms such as hand position (based on three dimensions), palm direction(represented in geometrical data), each fingers bending angle, distance from other fingers etc.
And Leap motion device control module delivers converted data set of 3D hand status to SVM machine learning module.
Class : class SignListener

3D model of current hand

4.4.6 SVM machine learning module

This module receives hand data samples from leap motion control module. (Each hand data sample represents hand data status of each moment.) By combining and analyzing plenty of samples (using SVM machine learning algorithm), delicately refined hand data set is generated. Developer can assign this well-defined hand data to corresponding sign language.
(without refining process of SVM algorithm, accuracy of recognition drops sharply.
Class : class SignListener



The graph of SVM

4.4.7 Hand data processing module

This module receives information of current hand status from Leap Motion device control module and compares received data with data samples initially assigned to each sign language.
If there exists data sample corresponds to current received hand status data, this module recognizes input value as valid sign language and returns corresponding assigned sign language.
Class : class SignListener
There is no picture for this module.

4.4.8 Sign language (vowels and consonants) practice module

This module implements sign language vowel and consonant practice. It shows question and answer for each question. They consist of vowels and consonants. User can follow shown sign language. This module compares users hand status with answer for current question. If they are same, it proceeds to next question and repeat above process until all the practice is over.
Class : class basic_prac

The picture of basic practice

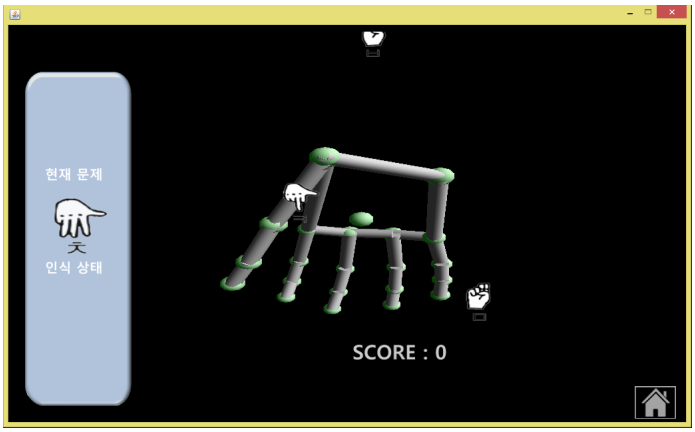
4.4.9 Sign language (words) practice module

This module implements sign language words practice. It shows question and answer for each question. They consist of words. User can follow shown sign language. This module compares users hand status with answer for current question. If they are same, it proceeds to next question and repeat above process until all the practice is over.
Class : class word_prac

The picture of basic practice

4.4.10 Sign language (words) game module

This module implements sign language game. It shows question in forms of rain. Also shows users score on the right. (indicates how well user answers for given questions.) Questions consist of words. This module receive users hand status and compares it with an answer for current question. After 1 minute , this module reflects the result of comparison to score and game is over.
Class : , class rain_game



The picture of rain game

4.5 Use Cases

4.5.1 Sign-in and Login

This use case satisfies user can login and Separating user mode by normal user and visitor. requirements.
Step 1.
Run KASLE program
Step 2.
Click (Sign-in) button.
Step 3.
Enter the basic information of user.
Step 4.
Click (OK) button.
Step 5.

Click (login) button.
Step 6.
Enter ID and password.
Step 7.
Check whether there is error message or not.

4.5.2 User can see his(her) own historical information

This use case satisfies User can see his(or her) accumulated historical information on KASLE program. requirement.
Step 1.
Run KASLE program
Step 2.
Click Login button.
Step 3.
Enter ID and password. (login is necessarily needed.)
Step 4.
After you are logged in, click (personal history) at the main menu.
Step 5.
Check your accumulated information on KASLE program. (average of game scores until then, how many practices youve ever done, etc.)

4.5.3 Properly showing hand data

This use case satisfies Get hand data Using LeapMotion hardware requirement.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click any practice or game menu at main page.
Step 4.
Check the box at the bottom side of screen which represents users current hand status.

4.5.4 Run Vowels and Consonants Practice

This use case satisfies User can practice vowels and consonants of sign language requirement.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click ㅐㄹ button at main menu.
Step 4.
Set the mode (vowels only, consonants only, both)
Step 5.
Check the box at the top of screen which shows current question.
Step 6.
Check the box at the right bottom side of screen which represents answer of current question.
Step 7.
Check the box at the top of screen which shows current question.
Step 8.
Check the box at the left bottom side of screen which represents users current hand status.
Step 9.
Follow the answer and check whether current question is finished and next question is shown.
Step 10.
Check all the questions are shown.

4.5.5 Run Words Practice

This use case satisfies User can practice words of sign language requirement.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click ㄹ button at main menu.
Step 4.
Check the box at the top of screen which shows current question.
Step 5.
Check the box at the right bottom side of screen which represents answer of current question.
Step 6.
Check the box at the top of screen which shows current question.
Step 7.
Check the box at the left bottom side of screen which represents users current hand status.
Step 8.
Follow the answer and check whether current question is finished and next question is shown.
Step 9.
Check all the questions are shown.

4.5.6 Run Words Practice

This use case satisfies User can practice words of sign language requirement.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click ㄹ button at main menu.
Step 4.
Check the box at the top of screen which shows current question.
Step 5.
Check the box at the right bottom side of screen which represents answer of current question.
Step 6.
Check the box at the top of screen which shows current question.
Step 7.
Check the box at the left bottom side of screen which represents users current hand status.
Step 8.
Follow the answer and check whether current question is finished and next question is shown.

Step 9.
Check all the questions are shown.

4.5.7 Run Vowels and Consonants Quiz

This use case satisfies User can play game on vowels and consonants.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click ł button at main menu.
Step 4.
Set the mode (vowels only, consonants only, both)
Step 5.
Check the box at the top of screen which shows current question.
Step 6.
Check the box at the top of screen which shows current question.
Step 7.
Check the box at the bottom side of screen which represents users current hand status.
Step 8.
If answering time (5 seconds per question) is over, program must finishes current question and shows next question.
Step 9.
Check whether the result of questions is properly reflected to score.
Step 10.
Check all the questions are shown.

4.5.8 Run Words Quiz

This use case satisfies User can play game on words.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click button at main menu.
Step 4.
Check the box at the top of screen which shows current question.
Step 5.
Check the box at the top of screen which shows current question.
Step 6.
Check the box at the bottom side of screen which represents users current hand status.
Step 7. I
f answering time (5 seconds per question) is over, program must finishes current question and shows next question.
Step 8.
Check whether the result of questions is properly reflected to score.
Step 9.
Check all the questions are shown.

4.5.9 Share to other applications

This use case satisfies Posting text entered by hand movement to Social Network Service requirement.
Step 1.
Run KASLE program
Step 2.
Login (it is not necessarily needed.)
Step 3.
Click (share) button at main menu.
Step 4.
Check the box at the bottom of screen which shows your hand data.
Step 5.
Express any words in sign language.
Step 6.
Check the box at the top of screen which shows words which is described by sign language.
Step 7.
If you want to express things that cant be described by sign language, you can click button on the screen to help you. Check whether buttons are operating properly.
Step 8.
Choose the application to which you want to share your text written by KASLE.
Step 9.
Click (share) button.
Step 10.
Check whether there is no error message.
Step 11.
Check the external application of your choice to receive text written by KASLE properly.

5 Software Installation Guide

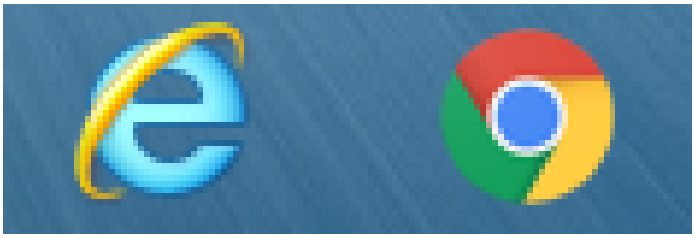
5.1 Installation

5.1.1 System Requirement

The minimum system requirements are:
1. Windows 7+ or Mac OS X 10.7+
2. AMD Phenom II or Intel Core i3/i5/i7 processor
3. 2 GB RAM
4. USB 2.0 port
5. Internet connection
6. Java

5.1.2 Installation for LeapMotion Driver

1. Turn on your Computer. If it doesn't turn on, please contact your adminstrator.
2. Double click on the your favorite browser, as chrome, explorer, or firefox.

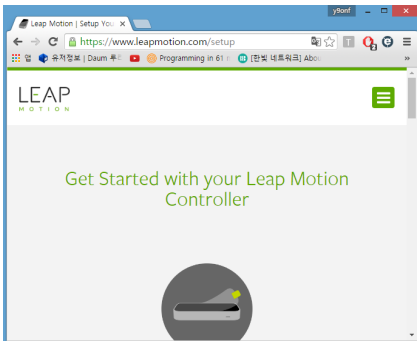


internet browser looks like this

3. If you can't turn on the internet, please check your connections.

If it is red or exclamination mark please contact adminstaror

4. Next, insert https://www.leapmotion.com/setup on the browser

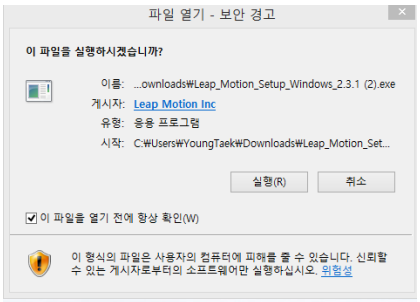


Location address

5. Then, on the website, please click on the 'Windows Download(DESKTOP)'

Please click this button twice

6. When the driver finished download, please click on the driver, and click execute 7.Check your security control and allow program to download

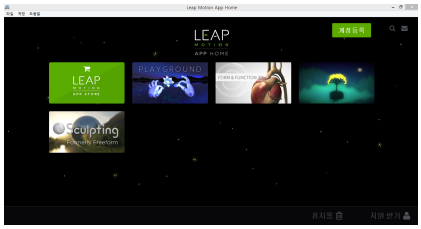


Please click execute

8.If you see download is completed, check 'experience LeapMotion'and click finish

Please click finish

9.You will see main page of LeapMotion applications.

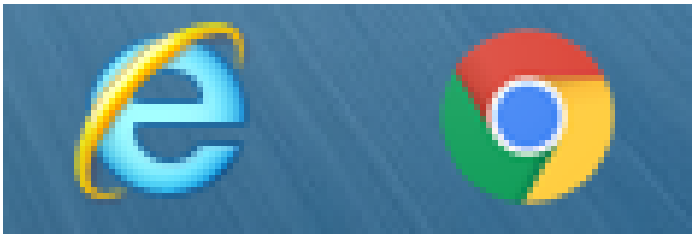


If you see this picture, your download is completed

NOTE: If software doesn't work, or if you have any other questions
Please contact our email, kyoon3@naver.com

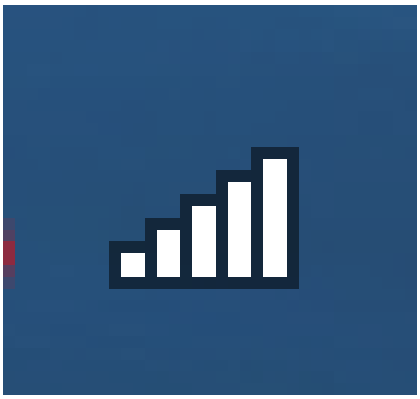
5.1.3 Installation for KASLE

1. Turn on your Computer. If it doesn't turn on, please contact your adminstrator.
2. Double click on the your favorite browser, as chrome, explorer, or firefox.



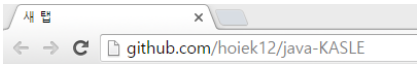
internet browser looks like this

3. If you can't turn on the internet, please check your connections.



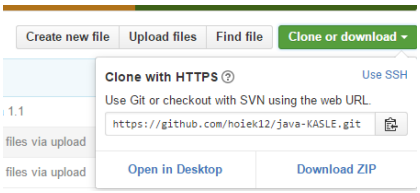
If it is red or exclamination mark please contact adminstaror

4. Next, insert https://github.com/hoiek12/java-KASLE



I

5. Then, on the repository section, please click download ZIP. Download will begin shortly.



Click download ZIP

6. When the KASLE finished download, please click finish
7. Go to the downloaded folder and double click on KASLE.exe.
8. If KASLE.exe works, close your browser and enjoy KASLE.

Enjoy

NOTE: If software doesn't work, or if you have any other questions
Please contact our email, kyoon3@naver.com

6 Discussion

6.1 Discussion

This software development project was first time for all of our team members. So, we expected that our project could encounter a lot of adversities. And it was actually tough more than we expected. And there were so many unexpected challenges weve met. But of course, we have learned a lot of things from that challenges. We could understand how the software development process goes and how many problems that could possibly happen during the process.

At the very beginning of the project, we had problem to decide our program to develop. There were so many ideas. So negotiation was not that easy. After we decide what to develop, there were many problems we had to handle. So, we tried to meet as frequent as possible. However, we could not avoid communication problem. When our team members had meeting, we drew blue print of program and decide who has responsible for what. But we could not decide every implementation details and design. So, we had to contact each other to negotiate and how to achieve each member's job and how to integrate each member's code.

Second hardship was using open source algorithms. At first, we thought that using open source code must be easy. Because we thought that if we need some feature and there is a open source code which implements our feature, we can just copy and paste it to our code. But if we do that so, there were so many errors. We had to modify source code to make it work together smoothly with other features.

Third hardship was related to hardware issue. We used hardware called LeapMotion to get data from hands. However leap motion camera was not good as we thought, so it was very hard to get accurate data. So, we had to apply SVM machine learning algorithm to improve accuracy. Using SVM algorithm was also big challenge for us.

Forth, we had an problem with Java. Not because of Java itself, but the problem was our low experience with java. So, we had to study how JAVA works.

Lastly, we didnt have experiences working together with other people on one project. And making one function was kind of easy but integrating them and make them work together was harder than we thought. Also, there were valuable experiences despite these hardships.

First was we could learn Java. We didnt have any experience with Java before this project. For now, we have learned a lot of things about JAVA and software development. Second, implementing small feature was relatively easy. However integrating them and make it work together without error was extremely hard. It requires not only coding each feature properly but also proper structure of program. Third, we learned how to make GUI. Before this project we thought that making GUI was easy. But in reality, it was not that easy to implement detailed implementation. So, we use a lot of tools to help us like 'Jigloo' and 'SwingWorker'.

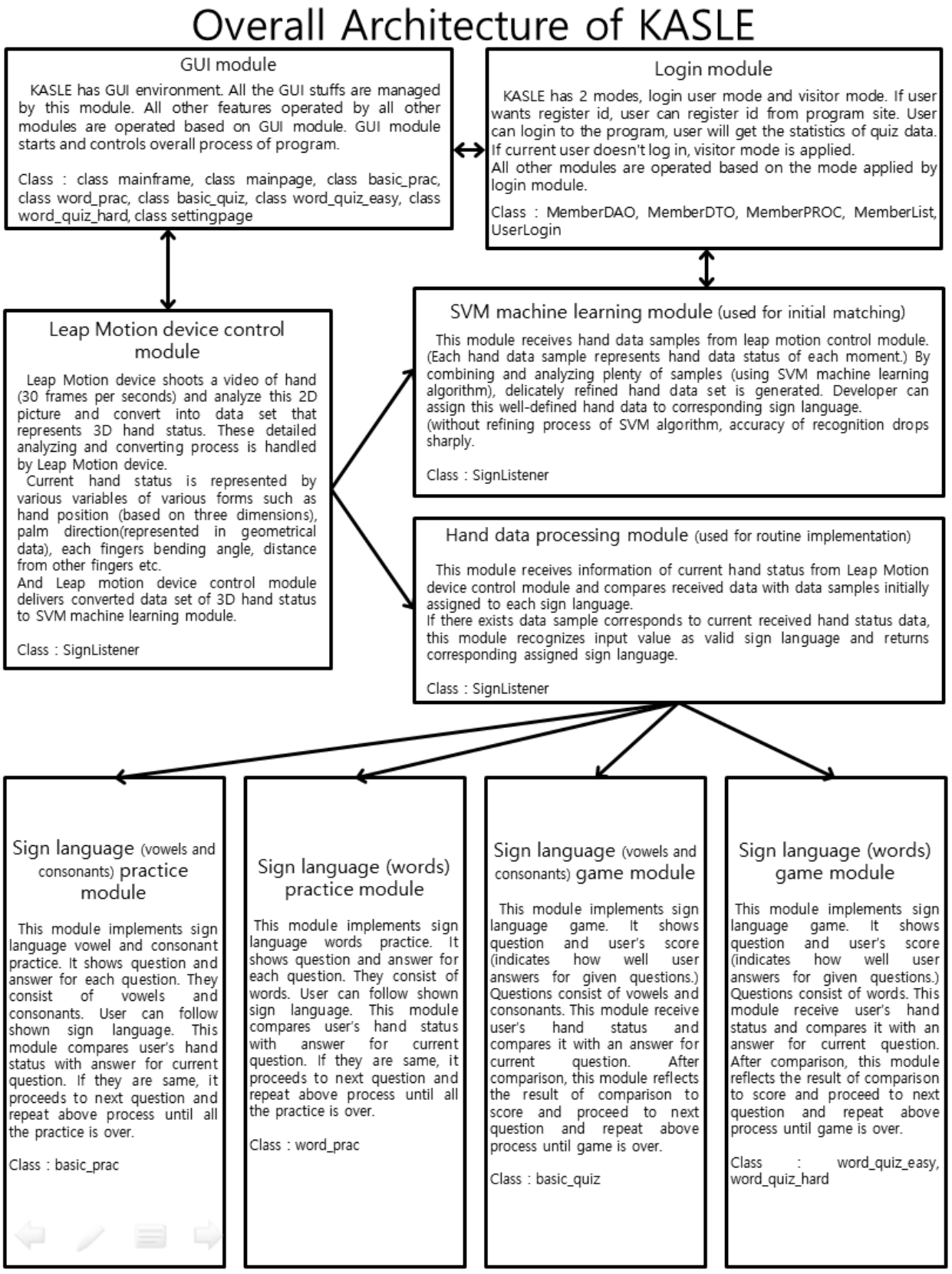

Overall Architecture of KASLE
GUI module
KASLE has GUI environment. All the GUI stuffs are managed by this module. All other features operated by all other modules are operated based on GUI module. GUI module starts and controls overall process of program.
Class : class mainframe, class mainpage, class basic_prac, class word_prac, class basic_quiz, class word_quiz_easy, class word_quiz_hard, class settingpage
Login module
KASLE has 2 modes, login user mode and visitor mode. If user wants register id, user can register id from program site. User can login to the program, user will get the statistics of quiz data. If current user doesn't log in, visitor mode is applied.
All other modules are operated based on the mode applied by login module.
Class : MemberDAO, MemberDTO, MemberPROC, MemberList, UserLogin
Leap Motion device control module
Leap Motion device shoots a video of hand (30 frames per seconds) and analyze this 2D picture and convert into data set that represents 3D hand status. These detailed analyzing and converting process is handled by Leap Motion device.
Current hand status is represented by various variables of various forms such as hand position (based on three dimensions), palm direction(represented in geometrical data), each fingers bending angle, distance from other fingers etc.
And Leap motion device control module delivers converted data set of 3D hand status to SVM machine learning module.
Class : SignListener
SVM machine learning module (used for initial matching)
This module receives hand data samples from leap motion control module. (Each hand data sample represents hand data status of each moment.) By combining and analyzing plenty of samples (using SVM machine learning algorithm), delicately refined hand data set is generated. Developer can assign this well-defined hand data to corresponding sign language.
(without refining process of SVM algorithm, accuracy of recognition drops sharply.
Class : SignListener
Hand data processing module (used for routine implementation)
This module receives information of current hand status from Leap Motion device control module and compares received data with data samples initially assigned to each sign language.
If there exists data sample corresponds to current received hand status data, this module recognizes input value as valid sign language and returns corresponding assigned sign language.
Class : SignListener
Sign language (vowels and consonants) practice module
This module implements sign language vowel and consonant practice. It shows question and answer for each question. They consist of vowels and consonants. User can follow shown sign language. This module compares user's hand status with answer for current question. If they are same, it proceeds to next question and repeat above process until all the practice is over.
Class : basic_prac
Sign language (words) practice module
This module implements sign language words practice. It shows question and answer for each question. They consist of words. User can follow shown sign language. This module compares user's hand status with answer for current question. If they are same, it proceeds to next question and repeat above process until all the practice is over.
Class : word_prac
Sign language (vowels and consonants) game module
This module implements sign language game. It shows question and user's score (indicates how well user answers for given questions.) Questions consist of vowels and consonants. This module receive user's hand status and compares it with an answer for current question. After comparison, this module reflects the result of comparison to score and proceed to next question and repeat above process until game is over.
Class : basic_quiz
Sign language (words) game module
This module implements sign language game. It shows question and user's score (indicates how well user answers for given questions.) Questions consist of words. This module receive user's hand status and compares it with an answer for current question. After comparison, this module reflects the result of comparison to score and proceed to next question and repeat above process until game is over.
Class : word_quiz_easy, word_quiz_hard

Fig. 2