A Minor project report

on

Creating an Android Application for customizable Keyboard

Submitted by

Ohileshwar Itagi
13IT251
Praveen Kotre
13IT229
Shyam BS
13IT144
VI Sem B.Tech (IT)

in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY

In

INFORMATION TECHNOLOGY

At

Department of Information Technology
National Institute of Technology Karnataka, Surathkal.
April 2016

Department of Information Technology
National Institute of Technology Karnataka, Surathkal

HCI Project
End Semester Evaluation (April 2016)

Course Code: IT 351

Course Title: HCI Project

Title of the Project: Creating an Android Application for customizable Keyboard

Details of Project Group

Name Of Student	Register No.	Signature with Date
1. Ohileshwar Itagi	13IT251	
2. Shyam BS	13IT144	
3. Praveen Kotre	13IT229	

Name of Guide: Dr.GRM, Pragnesh T, Deepthi L.

Signature of the Project Guide (with date):

Place:

Date:

DECLARATION

I hereby declare that the project entitled “Creating an Android Application for Customizable Keyboard”, submitted by me for the course IT351 as part of the partial course requirements for the award of the degree of Bachelor of Technology in Information Technology at NITK Surathkal is my original work. I declare that the project has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles elsewhere.

(Signature of the Student)

Place: Mangalore
Date: 22-04-2016

Abstract

Every Desktop application has its own set of keys and controls. It is difficult for users to interact with the vast variety of portable gadgets they carry, especially in the area of text entry. The customizable keyboard application which we have developed provides the user to create a keyboard layout accommodating to his needs. Our project focuses on providing three functionalities namely customizing the keys shape/size, overlapping keys, usage of image keys. Our application also allows the user save his/her own keyboard template in secondary storage so that he can load his keyboard layout whenever he requires it. Upon creation of the application we have tested the application on the professional gaming community and data entry community. We have received approval of our application from the respective committees.

Contents

1 Introduction **1**

1.1 Motivation 2

2 Literature Survey **3**

2.1 Problem Statement 4

2.2 Research Objectives 4

3 System Design and Methodology **5**

3.1 System Architecture 5

3.2 Algorithms and Techniques 6

3.3 Detailed design and Methodologies 7

3.3.1 Design Keyboard Layout 7

3.3.2 Connect to the server 7

3.3.3 Choose between Keyboard and Mouse 7

3.3.4 Send KeyStrokes and Mouse Events to the Desktop 8

4 Work Done **9**

4.1 Experimental Framework 9

4.2 Results 10

4.3 Individual Contribution of Project Members 10

5 Conclusion **11**

6 Future Work **11**

List of Figures

1 System design flow 5
2 Flow Chart 6

List of Tables

1 Introduction

Customizable Keyboard is an on screen keyboard designed to have a flexible key configuration and give functionality which is well beyond that of a regular keyboard. Customizable Keyboard allows the user to create his/her own layout so that the program can conform to the needs and preferences of the user, instead of the user growing more and more accustomed to different programs.

The application is developed for Android Devices with a target API of Android 4.2 Ginger Bread. The App mainly contains two tabs namely "Keys" and "Keyboard". Whenever the user clicks on any button present on the keys tab that particular button pops up in the Keyboard tab. In the Keyboard tab, the user can rearrange the position of the buttons by simply dragging and dropping them.

The user is also presented with two radio buttons. One radio button is to increase/decrease the height of the keys. Another radio button is to increase/decrease the width of the keys. There is a drop down spinner menu which shows the amount by which the size of the keys must increase or decrease.

There is a type/edit toggle switch. Whenever the user is in edit mode he can re-size/reshape the keys as per his requirements. Whenever the user clicks on type mode, the user can start using his keyboard layout to actually start sending keystrokes to the server machine over the network. The keystrokes logs on to the server machine which could be a laptop, Desktop, TV etc.

There is java code running on the server which will be constantly listening to the input provided by the app. The java code contains an object of Robot class which will be used to generate keystrokes regardless of the Operating system and hardware details of the device.

There is another tab, a third tab which is used as a mouse pad for the application. It consists of a rectangular mouse pad and three buttons inserted horizontally at the bottom row. One is for Left Mouse Button, next is for Middle Mouse Button, third is for Right Mouse Button. Whenever the user uses the mouse pad his current touch "x" and "y" co-ordinates are sent over to the system. Also if the user clicks on Left Mouse Button / Middle Mouse Button / Right Mouse Button the corresponding content is sent to the server/device. The Robot class is again used to implement the mouse clicks of left,right,middle mouse buttons.

1.1 Motivation

As the number of desktop applications are increasing everyday, it is becoming more and more difficult for the user to remember each and every configuration of keys in all the apps. Thus we need a system for the user to manage all his keyboard configurations and reload the template of his keys whenever it is required.

Any application installed will not be able to make use of all the keys in the keyboard. Thus there are a l ot if redundant keys in a keyboard based upon the application. Hence we need an application for users to enter only those keys that are required for him in his application. Some keys are more important and frequently pressed than others. Thus the keyboard should provide an option of re-sizing his keys so that the frequently pressed keys is larger in size than non frequently pressed key.

Sometimes the user might want to press multiple keys at the same time to achieve some functionality Eg: Alt+Tab for switching between desktop apps, Ctrl+Alt+Del for entering the task manager quickly. So we need a system which allows combining several keys present in the keyboard tab to become the same key. When the user presses a combined key, the keystroke corresponding to the effect of the combination is executed at the server side using the Robot Class Object in java.

2 Literature Survey

There are a few research oriented works which implement a customizable keyboard. Some of them are:-

- **Customizable Keyboard: Eric S. Missimmer, Samuel Epstein, Dept of Computer Science Boston University**
    - In this paper instead of giving the user a keyboard layout, they allow the user to create a layout that is accommodating to the user's needs.
    - They created a functionality to allow the user to select from a variety of keyboard modes.
    - The keyboard designed by them provides more functionality than a typical On Screen Keyboard including the ability to control infrared devices.
    - Keyboard modes are loaded at run time dynamically to allow faster development and easier deployment of different keyboard modes.
- **Android based Control Interface Solution for Windows Applications: Iliyan Nachev, Stoyan Maleshkov, Technical University of Sofia, Bulgaria**
    - The purpose of this paper was to present an approach for controlling remotely a desktop application under Windows Operating System through Wifi / Bluetooth connection with a device runninng on Android OS.
    - Remote Droid offers simple and convenient remote control over Wi-Fi . It is an open source software and available for free on google play store.
    - The testing of the application was performed on a PC with Windows 7 Operating System, smart phone with Android 2.3, a tablet with Android 4.1 and included work with many common applications like music and media players,text editors, presentation software.
    - That way the solution has been verified to accomplish its objective to provide a convenient way of human-computer interaction from a small distance via wireless communication and to be an equivalent substitute for mouse (touchpad) and keyboard for a large set of application types.

All the above approaches focus mainly on interacting with a laptop or a PC with an Android Application. The main advantages of the above mentioned applications compared to other offerings on the market is the user-friendly control and intuitive experience.

2.1 Problem Statement

Design and Develop a customisable Keyboard Application in Android, which provides these three main functionalities :-

- Customizing the keys shape/size.
- Overlapping of keys
- Usage of image keys

Upon connecting to the Desktop/Laptop the application must be able to send keystrokes and mouse clicks to the Desktop over the shared wifi network.

The desktop must then respond to the inputs sent over the wifi from the android device. It should provide the ability to control the PC with Android Smart Device via touch screen.

2.2 Research Objectives

We wish to assess this new mode of interaction for its usability, responsiveness, recoverability etc. There are many types of keyboards ranging from QWERTY, DVORAK, Colemak, Maltron, Azerty.

Instead of having so many keyboards for different users we propose to use a single keyboard with many layouts, in which the key positions/sizes etc are modifiable for better user ergonomics.

We also plan on testing out the user experience of our keyboard by surveying a gaming and typing community of our college.

3 System Design and Methodology

In this part of the paper, we build an Android Application for a customizable keyboard to automatically send keystrokes and mouse clicks over to the network. We formulate the sending keystrokes and mouse clicks as a server client interaction task over a network.

3.1 System Architecture

Our goal here is given the key strokes and mouse clicks from the android device from various keyboard layouts, we have to transfer the inputs over to the PC.Then we will have to wait for corresponding response. All our implementations have been done using Android Studio, Java and other Java libraries. Our overall approach is as follows:

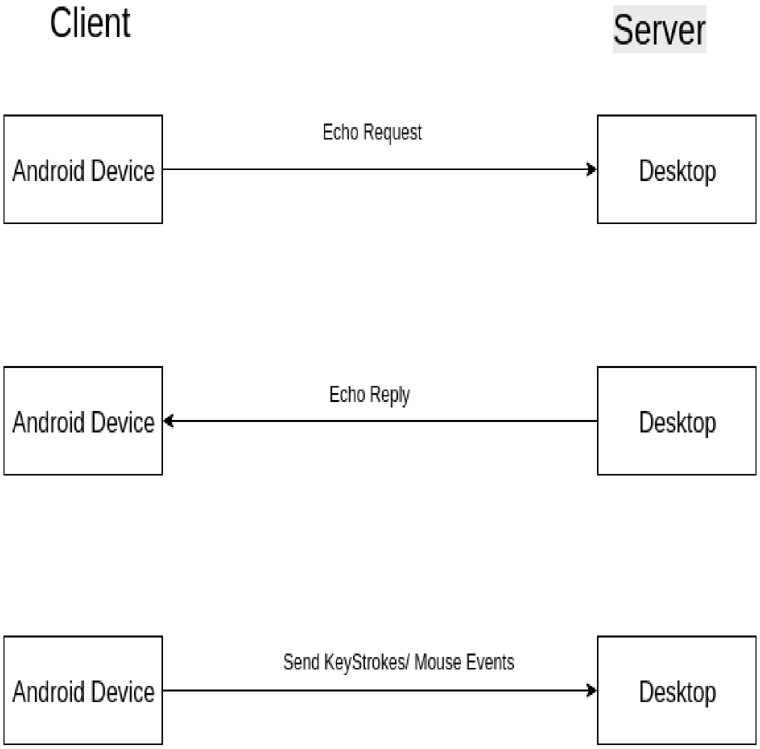

Figure 1: System design flow

3.2 Algorithms and Techniques

- First the user designs his/her own keyboard using the interactive User Interface of our application.
- Then the user uses this android application to connect to the server.
- Once connection is established the user chooses between two tabs.
- One tab is for the keyboard and another tab is for the mouse pad.
- After selection the user sends keystrokes/ mouse strokes over the network.

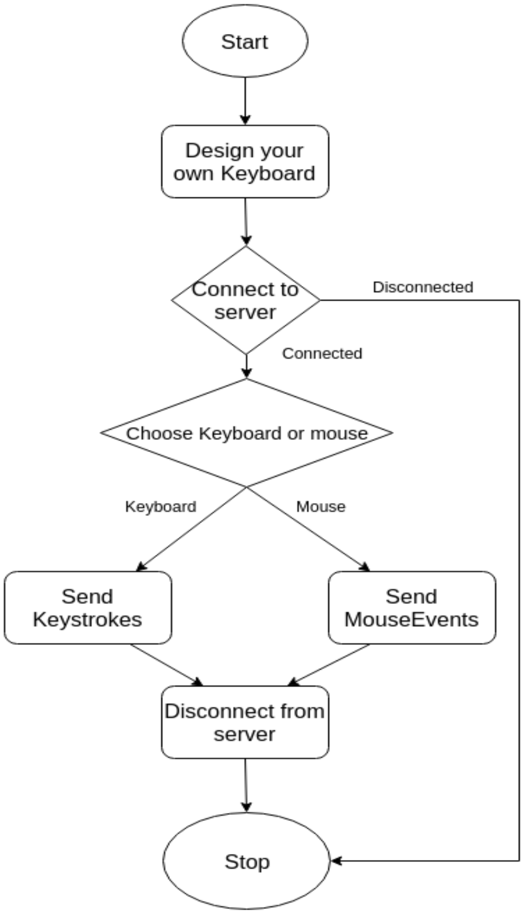

Figure 2: Flow Chart

3.3 Detailed design and Methodologies

As mentioned earlier in the system architecture it contains a total of 4 important parts.which are

- Designing our own Keyboard Layout.
- Connect to the server(Desktop).
- Choose between keyboard and mouse.
- Send KeyStrokes and Mouse events to the Desktop.

Each step is explained below in detail.

3.3.1 Design Keyboard Layout

The user is given two tabs to design his keyboard. In the first tab there is a set of keys A-Z, Spacebar, CTRL, ALT, DEL, Enter, Backspace. Whenever the user clicks on a key in this tab a copy of the key is added to the Custom keyboard tab.

When the user has inserted all the necessary keys that he wishes to have on the keyboard layout, he can modify them in two ways.
1) He can increase the size of the keys both horizontally and vertically.
2) He can place the keys at any position he wants.
3) He can also combine keys by placing one on top of another.

Once the keyboard is designed by the user according to his needs and requirements the process moves on to the next stage.

3.3.2 Connect to the server

The Desktop/PC is running a server creation code in java using Java Socket. Whenever the client sends an echo request to the server(PC) the server responds by replying to the client(Android Device) that it is ready to accept the incoming messages.

Upon successfull connection to the server the client displays to the user that it is succesfully connected to the PC and is now ready to send keystrokes and mouse clicks to the PC.

3.3.3 Choose between Keyboard and Mouse

The user is provided with two tabs. One is for the designed keyboard layout and another is for the mouse. Based on the circumstances the user can switch between mouse and keyboard

The keyboard tab contains the keyboard layout designed by the user. The mouse tab contains a touch pad and three buttons corresponding to Left Mouse, Middle Mouse, Right Mouse Button.

3.3.4 Send KeyStrokes and Mouse Events to the Desktop

The PC code accepts the inputs provided to the server(PC) by the client(Android Device).

There is an object of Robot Class in Java which converts the inputs to keystrokes independent of the hardware configurations of the Desktop.

The Robot class object also handles the mouse events. It takes in the x and y coordinates of the mouse pointer from the Android Device and projects the mouse position on the Server(PC).

4 Work Done

4.1 Experimental Framework

The Android Application was built using Android Studio Development Environment. The main application has three tabs in the main activity.

One tab for all available keys.In this tab all types of keys like A-Z, Enter,Backspace, Spacebar, Shift, Arrow Keys etc are present, Whenever the user clicks on any one of these keys that paricular key is replicated in the next tab.

In this tab, there is a layout space available for placement of keys. There is a radio button which decides whether the user is in text editing mode or keyboard designing mode. In the Keyboard Designing mode, there are two switch buttons. One for horizontal increment of size and another for vertical increment of size. Also there is a spinner object which shows by how much amount of size every time an increment or decrement needs to be made.

In this tab, there is a huge rectangle resembling the touch pad of the mouse, followed by three buttons. The buttons in order corresponds to Left Mouse Buttons(LMB), Middle Mouse Buttons(MMB), Right Mouse Button (RMB).

Whenever the user wants to connect to a particular machine the user must start the server code in that machine. The server code is written in java using Java Sockets. During connection the Android Device sends a Hello Request to the server. The server sends back the confirmation to the client.

Once the connection is established, the server accepts the messages from the client. Based on the pre defined messages the client sends and the server receives, the server sends keystrokes and mouse events to the hardware of the PC. This is accomplished using a Robot Class in Java. The robot class's object has two functions for handling keys. One function is called keyPress() and another function is called keyRelease(). Both take in ascii value of the key as the parameter.

The mouse cordinates in the screen are converted to hardware specification by the mouseMove() function of the robot class. It takes in two floating point numbers as x cordinate and y cordinate respectively.

4.2 Results

We tested our application on a large gaming community of our college. We made the gamers play standard games like RoadRash,NFS, GTA Vice City, Mortal Kombat 4, etc on their PC's using our application.

More than 80% of gamers said that their gaming experience improved in games like road rash, Need For Speed etc because of removal of all the redundant keys etc. In racing games only arrow keys,spacebar and one alphabet key is enough. The common key board layout used by gamers for racing games is given below.

For First person shooting games like CounterStrike- Condition Zero etc, the community said unanymously that their experience deteriorated due to constant switching between the mouse pad tab and keyboard layout tab.

For arcade games like Mortal Kombat 4, GTA Vice City etc, lots of users appreciated using our interface mainly because of the feature of combining keys which is necessary in arcade games.

4.3 Individual Contribution of Project Members

Praveen Kotre(13IT229) : The statistical analysis was done using python scripts by him. The data obtained from the website was not in the proper format we expected it to be. Thus preprocessing of rows in the dataset needed to be done. This was done using python code by Praveen. He made most part of the report.

Ohileshwar Itagi (13IT251) : With the data obtained in the proper format required to build the classifier Ohileshwar wrote the code which will build the model for this multi class classification problem. It first creates an Support Vector Machine(SVM) technique to a multidimensional dataset.The output will be the probabilities of each question belonging to each of the 5 classes. He performed most part of Literature survey.

Shyam BS (13IT144) : Our model gives probability of for each classification. Hence using a normal model for calculating efficiency seemed unfit. We had to use a multiclass log loss method to calculate the efficiency. We also wanted a model compatible to the output given by vowpal-wabbit. Shyam wrote the compatible code to compute the score of the model by using the multi log loss classification . It was written in python. He also performed the 10 fold cross validation. He wrote a python script and a shell script for the purpose.

5 Conclusion

As a conclusion the developed application bundle provides a useful ability to control a PC with an Android smart device via touch screen. Every common user action could be executed without hesitation, the actions are seen on the remote PC with minimal (non-perceivable) delay and many applications could benefit from this kind of input device. The main advantage of the proposed solution compared to the other offerings on the market is the user-friendly control and intuitive experience. Furthermore it offers a type of wireless communication and auto-discovery mechanism; the Wifi implementation is stable and compatible with all major Wifi stacks; the delay between the user actions is minimal.

6 Future Work

We plan on resolving the issue of less usability of the application for First Person Shooter Games. We are also looking forward to extending this idea to a bluetooth implementation from our standard wifi implementation. We also plan on providing the ability for the user to save his keyboard layout for future usage.

References

[1] Eric S. Missimer, Samuel Epstein, John J Magee, Margrit Betke : Customizable Keyboard . Department of Computer Science, Boston University.

[2] Illiyan Nachev, Stoyan Maleshov : Android Based Control Interface Solution for Windows Applications. Technical University of Sofia, Bulgaria.