IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

DESIGNING A MOBILE EYES-FREE SOFT KEYBOARD FOR VISUALLY IMPAIRED IN ANDROID

Bhagyesh Patel*, Sushant Patil, Tanmay Puranik
Department of computer engineering vidyavardhini's college of engineering and technology k. T. Marg, vasai road (w.) Dist-palghar, maharashtra, India

DOI: 10.5281/zenodo.48377

ABSTRACT
Typing is the essence of mobile communication and connectivity. Arguably, today's technologies support mobile text input poorly, though this is the case for everyone, but it is particularly relevant to the visually impaired. The keyboard is specially designed to provide easy typing technology for the visually impaired. There are total 8 logical sensing areas (as Braille language uses cells to denote a character) that are used for acquiring the characters. The whole keyboard works based on Braille system. The words that are typed are recognized by searching in database dictionary and also checked whether the word is valid or not. In this keyboard design, the areas (or cells) are aligned according to the Braille language. The main advantage of this project is that the visually impaired do not need any training since most of them are already specialized in Braille language, barring some initial familiarity with keyboard.

KEYWORDS: Braille, Keyboard, Sensing Areas, Pattern Recognition.

INTRODUCTION
Currently, there are 5.28 billion cellular subscriptions worldwide, according to the International Telecommunication Union, October 2010. Of these, 3.8 billion cell phone subscriptions are in the developing world and 1.4 billion are in developed nations. Taken together with the fact that 90% of the world's visually impaired people live in developing nations (WHO 2010), it is a moral imperative to design an accessible interface for effective mobile eyes-free text input. The project explores the use of Braille for eyes-free text input on touch screen mobile devices. The Braille code consists of a 4 by 2 binary matrix that encodes up to 63 characters, excluding the state with no dots.

MATERIALS AND METHODS
Description of Modules
Calibration
First of all the calibration of user fingers takes place.The user is asked to place his/her fingers on the screen. The fingers can be placed on the screen one by one.

Pattern Matching
When the user starts inputting the patterns into the screen, the patterns is recognized for Braille character.

Word checking
If the pattern inputted by user matches with any of the patterns in the database, the word corresponding to the pattern is given as output to user.

Text-To-Speech
The words given as output to the user are also read out to the user, since the user is visually impaired.

ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

Implementation of Modules

Calibration

Ask user to put fingers on screen one by one from left-hand to right-hand fingers.Get the input coordinates of fingers one by one. Create dynamic custom-button (circular button) and set the coordinates to input coordinates.

Pattern Matching

Ask the user to start typing. User will simultaneously click the buttons to create the pattern. The code will find out which buttons are pressed and will set visibility to “on” or change the button color while the button is pressed. Code will generate letter corresponding to the pattern typed if letter is not found then it will give error through voice. The Matching is being done using the Hash Map. In Hash Map, there are key-value pairs. In our case, the keys are the alphabets and the values are their corresponding patterns. When any of the patterns in the Hash map matches, its corresponding key i.e. the alphabet is given to the user.

Word checking from the Database

There are fixed character patterns corresponding to each and every word. When the user ends a particular Braille sequence, the system will check whether these sequence match with any of the character patterns in the database. If the character pattern matches then the user input word is read out to the user.

Text-To-Speech

Android has had very robust text-to-speech (TTS) functionality. Text-To-Speech instance can only be used to synthesize text once it has completed its initialization. Implement the TextToSpeech.OnInitListener to be notified of the completion of the initialization. Once the initialization is completed, the system can use it for speaking out the words to the user. The typed text is read out to the user on completion of one word.

RESULTS AND DISCUSSION

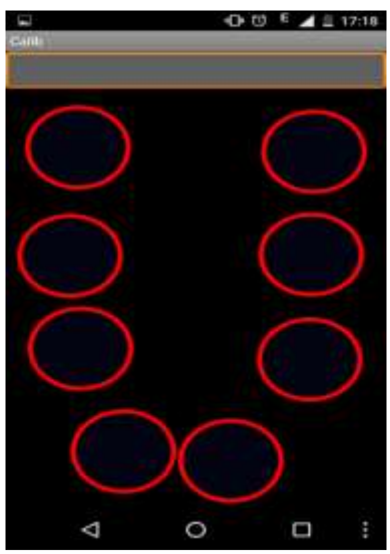

Fig 1: Calibrated Sensing Areas

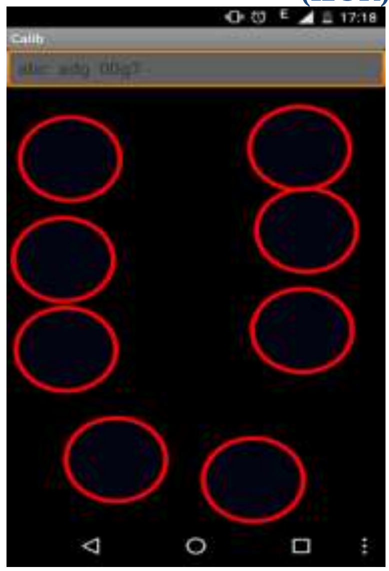

Fig. 2: Input through the calibrated areas

CONCLUSION

The project improves over existing solutions for Braille Keyboard by overcoming the disadvantages of fixed areas and lack of continuity in input process. The project is developed as a prototype which can find use in applications. The project is very much viable due to its dynamic interface and ease of use for the visually impaired people. It finds applications where any soft keyboard does providing a way for the visually impaired people to communicate. The project is implemented and deployed on android platform due to its availability and wide use.

ACKNOWLEDGEMENTS

It gives us a great pleasure to express our deep sense of gratitude to our guide **Assistant Prof. Smita Jawale** for her valuable support and encouraging mentality throughout our ongoing project. We are highly obliged to her for providing us this opportunity to carry out her ideas and work during our project period and helping us to gain the successful progress of our Project. Our special thanks is going to Head of the Department of Computer Engineering of our college, **Dr. Swapna Borde** and to all of the faculties for allowing us to come here and encouraging us constantly to work hard in this project. We are highly grateful to the Honorable Principal of VCET, **Dr. A. V. Bhonsale** for giving us this golden opportunity to be a part of this organization for this period.

REFERENCES

[1] M. Bonner, J. Brudvik, G. Abowd, and W. Edwards. *No-Look Notes: Accessible eyes-free multi-touch text entry*. Pervasive Computing, pages 409–426, 2010.

[2] H. Fard and B. Chuangjun. *Braille-based text input for multi-touch screen mobile phones*. 2011.

[3] B. Frey, C. Southern, and M. Romero.Braille Touch: mobile texting for the visually impaired. Universal Access in Human-Computer Interaction. Context Diversity, pages 19–25, 2011.

[4] T. Guerreiro, P. Lago´a, H. Nicolau, P. Santana, and J. Jorge.*Mobile text-entry models for people with disabilities.*

[5] NektariosPaisios, Alexander Rubinsteyn, Lakshminarayanan Subramanian, *Mobile Brailler: Making Touch-Screen Typing Accessible to Visually Impaired User.*