

BrailleTap: Developing a Calculator Based on Braille Using Tap Gestures

Mrim Alnfai^(✉) and Srinivas Sampalli

Faculty of Computer Science, Halifax, Canada
mrim@dal.ca, srini@cs.dal.ca

Abstract. Touchscreen calculators require a user to visually locate a button on the screen. This poses a challenge for individuals with limited or no vision. To overcome this challenge, prior to use these individuals must invest additional time and energy into memorizing the calculator layout. There is currently no application that simplifies the calculator user-interface for individuals with limited or no vision. To fill this void, this paper proposes BrailleTap. BrailleTap is a button-free calculator application for touchscreen devices that utilizes tap gestures to operate. Using finger tap, the user can input a number in braille and the application will compute the braille code to its corresponding numerical value. Alternatively, using a swipe gesture, the user will be orally prompted to select the desired arithmetic operations. This paper presents our pilot study of BrailleTap that was conducted with two blind users. Results indicate that BrailleTap is faster and more accessible to blind users than the traditional touchscreen calculator. However, in order to increase the efficiency of BrailleTap further development of the underlying algorithm is required.

Keywords: Visually impaired · Braille · Touchscreen calculator · Mobile apps

1 Introduction

There have been significant improvements in the accessibility of touchscreen devices in recent years. One of the most significant improvements is the screen reader. The visually impaired user can touch the screen to identify the screen layout, and the screen reader reads out the item under the user's finger. This includes text, object names, and application labels. In order to activate an object, a user performs a double tap when he/she finds the location of the intended object.

Despite these improvements, locating an object on a touchscreen is a challenging task for people with limited or no vision [1]. For example, it is difficult for blind people to insert text on a touchscreen, even with audio feedback. The user has to find a particular character on the QWERTY keyboard and double tap to insert it. This is time consuming as users have to listen to every character when they explore the screen. The possibility of inserting the wrong character is high because the keys are small. Performing basic arithmetic operations on touchscreen calculator applications present a similar challenge: blind users have to search for their desired character through a time-consuming process that is prone to errors [1]. The calculator is considered to be the most commonly used

application at home, school, and while shopping [5]. As such, there is a need for accessible touchscreen calculator applications for the visually impaired.

To address this issue, we propose the BrailleTap calculator, a novel calculator specifically designed for blind users. The core idea is to permit users to insert numbers using braille, which is the fundamental writing system used by visually impaired people. The BrailleTap calculator relies on tap patterns, as opposed to tap locations, allowing visually impaired users to perform both simple and complex calculations in less time and with fewer errors. Previously, we presented the design of SingleTapBraille, a nonvisual text entry keyboard for smartphone devices [2, 3]. In contrast, the BrailleTap application enables for numerical computation thus allowing users to perform simple calculations using their language. BrailleTap was developed based on a similar technique to that used for SingleTapBraille. We integrated the single tap concept in the BrailleTap calculator application to make it accessible to individuals who are visually impaired. We also improved the algorithm accuracy to insert numbers on touchscreen devices.

The BrailleTap calculator is also a nonvisual number entry method which can be used for basic mathematic calculations and can potentially be used for different tasks, for example, entering phone numbers in a contact list, entering addresses in a map application and entering information in calendar applications. However, our initial design has implemented BrailleTap on a calculator. The BrailleTap calculator allows users to tap braille patterns on the screen to enter a number and uses defined gestures to allow users to select the operations and edit their entries. We provide an initial study to examine the performance of BrailleTap calculator with a blind user. The initial evaluation shows that BrailleTap is faster and more accessible than the TalkBack calculator.

2 Braille Code

Braille is a special code that uses a combination of up to six dots. Each character has a specific combination of these dots that distinguish it from others. The dots are organized in two columns, each column with three spaces [4]. Visually impaired users can read these characters using their fingertips. The codes of numbers in Braille are similar to the first ten characters, as shown in Fig. 1.

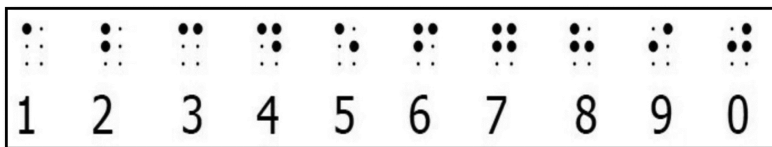


Fig. 1. Braille numbers

3 Related Work

Apple and Android have applied screen readers to smartphone devices to improve accessibility for blind people. Users can move their finger over the screen and the

TalkBack service speaks the object name under the user's finger. Despite this improvement, inserting numbers or arithmetic operations using the standard calculator with the TalkBack service is challenging [2, 3, 5]. The basic TalkBack calculator has one layer, with four operations on the right of side and the numbers 0–9 on the left side. The result of each calculation is displayed on the top of the screen. The user has to move his finger over the calculator screen and TalkBack reads out numbers or operations that are located under the user's finger. Thus, the user has to move his finger over the screen until he finds the desired number. If the user enters a wrong number, the user is required to find a clear button to delete the error. Thus, the main drawback of this method is that it is time consuming as it requires users to listen to all the keys until the desired key is heard [3, 14]. This technique also consumes time when the user makes mistakes since she has to reenter the numbers and operations from the beginning. However, the main strength of the TalkBack calculator is that it provides audio feedback that helps users ensure they enter the correct number.

In addition to the standard keyboard with TalkBack, researchers have proposed several other text entry techniques [6–12], which also predominantly require users to find a specific object location. Several developers have also introduced accessible text entry methods to overcome navigation problems, but they require users to remember particular gestures to enter text or numbers [5, 13].

Limited research has been done in the field of accessible number entry for blind users, and only one method has been integrated with an application for a particular aim, in this case an accessible calculator mainly for this population [5]. Previous work by Hesselman et al. [6] developed a number entry method on a large screen, where a blind user taps on the screen with the number of fingers of the entry number. Thus, in this application, users need to use two hands to interact with a touchscreen. It also allows audio feedback to help users confirm they have entered the intended number. Drawbacks of this technique are that it is limited to entering numbers only from 1 to 10 and requires users to use two hands to interact with a screen, which may be difficult as blind users tend to use one hand to hold a cane or a guide dog [2]. Another input method was proposed by Azenkot [14], which is called DigiTaps. This method was designed mainly for small screens where a user only needs three fingers to interact with the screen, but this application requires several steps to just input one digit. After inserting a digit, the app provides haptic and optional audio feedback.

The H4-writer is a text and number entry method introduced by MacKenzie et al. [6], which uses optimal prefix-free codes based on Huffman codes [15] to insert numbers as well as text. The H4-writer user interface has four buttons and the user need only use one thumb. Thus, it creates navigation issues and requires users to memorize the defined prefix codes. This is similar to Tapulator, a non-visual calculator, which was designed mainly for blind people. It uses prefix-free codes to enter numbers on a touchscreen; users tap on touchscreens using one to three fingers and swipe the screen to insert a number. It requires users to perform specific gestures to perform operations, such as a two consecutive swipes up to down to represent addition. The clearest advantages of this calculator are that it overcomes navigation problems by allowing users to tap anywhere on the screen and it provides audio feedback. Like BrailleTap, there are no numbers represented on the user interface; users can enter numbers on a screen based

on braille patterns. However, the major limitation of this method is that it is not easy to learn because users are required to memorize the defined prefix codes to enter a number or operation. It is also time consuming because users are required to complete more than three steps to enter a number.

4 The Design of the BrailleTap Calculator

Our main goal is to design an accessible calculator on a touchscreen device that allows blind users to tap braille numbers anywhere on a touchscreen using their thumb or finger without the need to find a particular number or operation on a calculator layout. In our proposed calculator, users are not required to navigate the calculator keys.

The calculator is designed based on braille codes. As mentioned above, the code has two columns and three rows. Blind users generally enter braille dots from left to right and from one to six. Our calculator follows the same principle, allowing users to insert braille dots from left to right and from one to six. Users can activate braille dots individually using their thumb or a finger to tap on the screen. In order to activate the first braille dot, users can tap once anywhere on the screen and this will represent the number “1”. In case the user wants to activate more than one dot, he/she can tap the screen quickly to create braille shapes. Once the user stops tapping on the screen for a short time, our algorithm will interpret the braille pattern and the corresponding letter. At the same time, a text-to-speech function will read out the resulting number. Our approach allows users to hold the smartphone device using one hand while they interact with the touchscreen using their thumb to tap braille dots anywhere on the screen. This follows Paisios [16], who found that the most accessible way for blind users to interact with a touchscreen is by using one finger.

In order to determine the corresponding number when users tap braille dots on a screen, our algorithm analyzes the relationship between all activated dots in each braille pattern based on the active dots’ coordinates. For instance, the number two will be represented when a user taps the first dot followed by the second dot on the touchscreen. Our algorithm is mainly based on the clustering of active dots for each number and identifying the relationship between those active dots. The algorithm clusters all the inserted taps on a touchscreen and analyzes all these taps based on the number of taps, the shape of these taps, as well as each tap’s coordinates. When the user stops tapping on the screen for a short interval, the taps are analyzed and the algorithm will compute the corresponding number. Table 1 and Fig. 2 represent the algorithms that have been used to examine the relationship between dots.

Table 1. Categorization of braille numbers based on number of dots

<p>a. Character that has one dot</p> <p>Number 1</p> <p>One tap anywhere in a mobile screen</p>	<p>e. Characters that have four dots</p> <p>Number 7</p> <p>$[X1-X2]<e; [X3-X4]<e; X2<X3; [Y1-Y3]<e; [Y2-Y4]<e; Y3<Y2$</p>
<p>b. Characters that have two dots</p> <p>Number 2</p> <p>$[X1-X2]<e$</p>	<p>Number 3</p> <p>$[Y1-Y2]<e$</p>
<p>Number 5</p> <p>$X1-X2<0; Y1-Y2<0$</p>	<p>Number 9</p> <p>$X1-X2>0; Y1-Y2<0$</p>
<p>c. Characters that have three dots</p> <p>Number 4</p> <p>$Y1-Y2=0; X2-X3=0$</p>	<p>Number</p> <p>$X1-X2=0; Y1-Y3=0$</p>
<p>d. Characters that have three dots</p> <p>Number 8</p> <p>$X1-X2=0; Y2-Y3=0$</p>	<p>Number 0</p> <p>$X1-X2<0; X2-X3=0; Y1-Y3=0$</p>

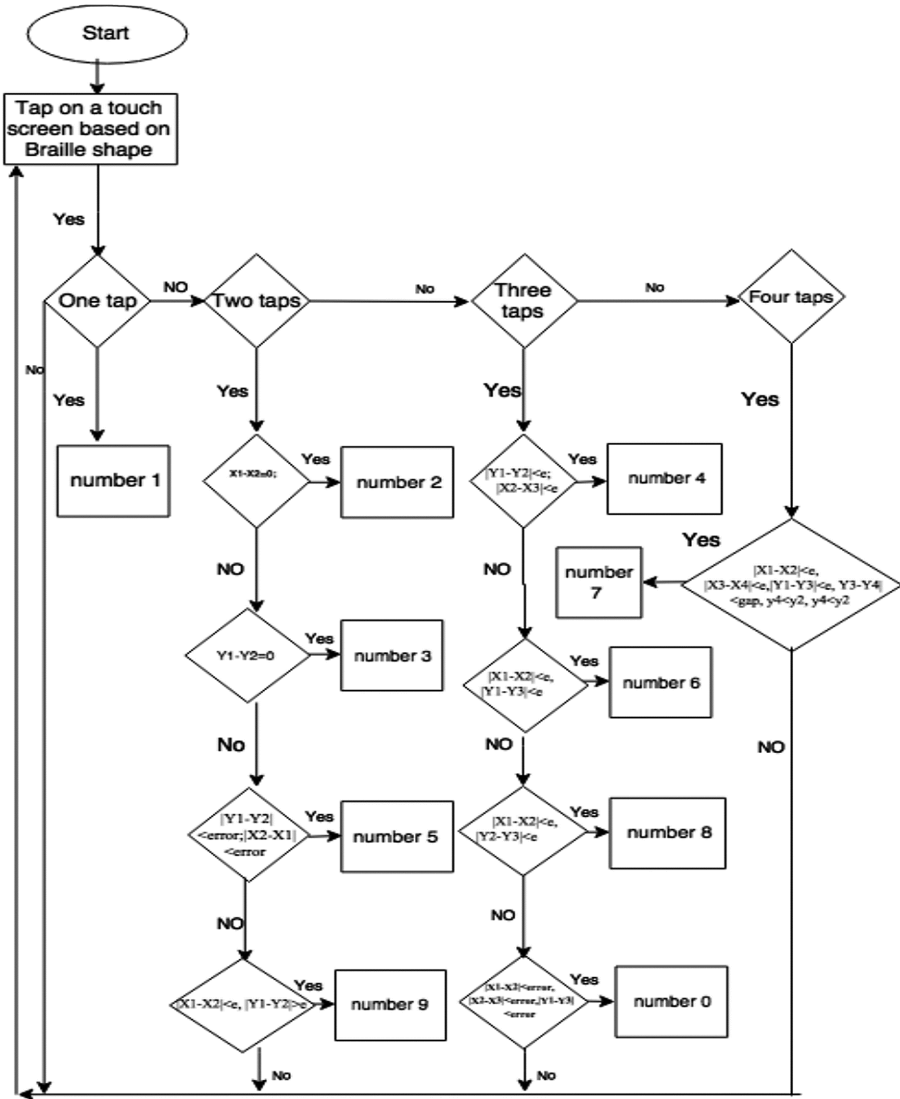


Fig. 2. Algorithm flowchart

The factors that indicate the relationship between braille dots are:

1. The value of coordinates (X, Y) for each tap on the screen
2. The number of dots in each number in braille

As seen in table above, each braille dot's position is reflected by both an x and y coordinate. These coordinates are used by the algorithm to create an equation that reflects the relationship between the position of the dots in each braille code. These equations

are represented below each braille code in the table above. Once the pattern of braille code is recognized the algorithm will present the corresponding number.

The error variable “e” is a defined value that is used as a threshold for whether two floating numbers are in a vertical or horizontal line. If the difference between the x or y values of two taps is smaller than the error value, the algorithm will accept it because it is difficult for a user to insert two taps in an exact vertical or horizontal line.

5 Representing Braille Numbers Using BrailleTap

BrailleTap allows operations to be entered in a way that is accessible and effective using swiping gestures with a single thumb. This is in accordance with Paisios and Richard et al. [16, 17], who found that using swiping and tapping to interact with a touchscreen is the easiest gesture and most accessible methods for blind users. In order to insert an operation, users can swipe vertically in a loop to activate a particular operation. The first swipe will activate the first operation and TalkBack service will speak the operation name. Then, if the users swipe vertically twice, they will obtain the second operation. For example, when the user performs the first vertically swipe from bottom to top, the app will say “addition”. In case the user does not want the addition operation, he/she can swipe again from bottom to top to select subtraction instead of addition. Performing a vertical swipe from top to bottom will insert division operation. A second swipe from top to bottom will insert multiplication instead of division. They can also swipe horizontally to edit their typing (see Table 2). After the users enter the mathematics equation, they can perform a long press gesture to obtain the result.

Table 2. Braille keyboard interaction techniques and their purposes.

Interaction techniques	Purposes
Swipe from bottom to top	Addition, subtraction
Swipe from top to bottom	Division, multiplication operations operations operations
Swipe from left to right	Clear operation
Swipe from right to left	Backspace
Long press	equal

The BrailleTap interface is based mainly on audio feedback and does not use any buttons on the screen. Users can tap and perform gestures anywhere on the screen (as shown in Fig. 3). The TalkBack accessibility service in Android platforms reads out the written text on the screen.

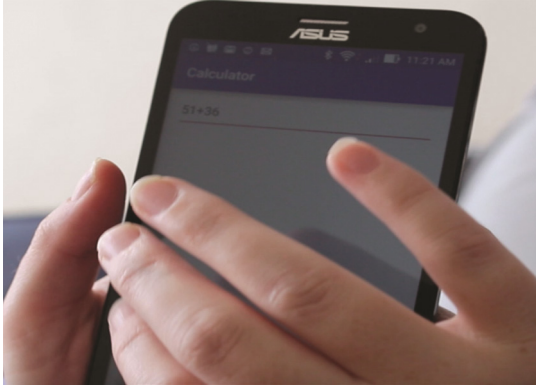


Fig. 3. BrailleTap user interface

6 Pilot Study

We evaluated the BrailleTap calculator to examine the performance of the calculator as well as to compare it with the TalkBack calculator. Other touch-based methods that aim to increase the accessibility of calculators to visually impaired individuals include Tapulator and DigiTaps. However, in contrast to TalkBack these methods are not available on smartphone devices. In addition, DigiTaps does not enable numerical computation. As a result, the researchers only compared BrailleTap to the TalkBack calculator. This evaluation was conducted with two participants who are completely blind and who are fluent in braille. The participants were involved from the initial phase of the calculator's design and provided some suggestions about choosing the appropriate touchscreen gestures for operations. After we designed the application, they were asked to perform eight mathematics equations using both the BrailleTap calculator and the TalkBack calculator.

During a brief training period of approximately five minutes, participants learned how to use the BrailleTap calculator. This pilot study utilized the counterbalance methodology meaning that participant 1 used the BrailleTap application prior to the button calculator and vice versa. Participants stated that they liked the application because they can use braille code, the language they are most familiar with, to input their commands.

We asked the participants to perform several mathematical equations using both the TalkBack calculator as well as the BrailleTap calculator. Table 2 indicates the amount of time that each participant spent to compute each mathematical equation on both calculators.

Table 3 shows that BrailleTap was faster than the button keyboard because it overcomes the navigation problem and the user can tap anywhere on the screen. BrailleTap is also faster because it eliminates noisy audio feedback that is used to help the user identify where he is on the calculator interface or identify the button name. In the interview, she stated that the BrailleTap calculator is a promising tool that allows users to enter numbers based on braille and perform operations using accessible gestures. She

also likes being able to perform several tasks on the touchscreen without having to identify an object location. She also found it helpful to delete everything at once using a simple gesture.

Table 3. The time spent to use both the button calculator and the BrailleTap calculator

Mathematical equation	Average time spent using button calculator	Average time spent using BrailleTap
$51 + 36$	28 s	23 s
$54 + 34 - 15$	71 s	27 s
$50 * 200$	96 s	48 s
$450 / 5$	108 s	15 s
$50 + 25 / 4$	87 s	35 s
$600 / 5 - 50$	119 s	39 s
$200 + 40 * 2$	109 s	64 s

Overall, the initial result indicates that the proposed calculator is easy to learn, more accessible than the TalkBack calculator, and it overcomes most of the limitations that are associated with the previously discussed calculators. In the BrailleTap calculator, users are not required to learn specific techniques to enter a number. Unlike Tapulator, where users have to memorize the prefix codes to be able to interact with the screen, BrailleTap allows them to simply use their experience with braille. Based on our observation of the participants' interaction with both calculators, we noticed that there are noticeable improvements in both accessibility and speed of the BrailleTap calculator over the button calculator. However, the most noticeable error that occurred during the study is that the users forgot where they had first tapped the screen when the braille code had more than three dots, like in the number 7. This type of error leads to typing the wrong number. When their attention was focused on where they tapped the screen, they correctly typed the number 7. Although it is common for these types of studies to use one or a few participants, the main limitation of the current study was being unable to recruit a large number of blind participants.

7 Conclusion

We have presented BrailleTap, a non-visual number entry application with audio feedback based on braille coding. The proposed calculator allows visually impaired people to perform basic operations on touchscreen devices more easily than presently available calculators. It does so by overcoming their main limitation, which is requiring visually impaired individuals to locate an object on a touchscreen.

Based on the initial study, there are certainly opportunities for improvement to the BrailleTap calculator. Firstly, we need to test the calculator with larger groups of people via both qualitative and quantitative methods in order to better understand its strengths and weaknesses. We will also measure total typing time, digits per minute, as well as identifying types of errors. We will improve the calculator based on the participant's

suggestions, for example, by adding other mathematics operations using accessible touchscreen gestures for blind users.

Acknowledgements. We thank the Canadian National Institute for Blind, and especially the study volunteers. We also gratefully acknowledge support from Taif University and the Saudi Arabian Cultural Bureau in Canada.

References

1. Nicolau, H., Guerreiro, T., Jorge, J., Gonçalves, D.: Proficient blind users and mobile text-entry. In: Proceedings of the 28th Annual European Conference on Cognitive Ergonomics (ECCE 2010), pp. 19–22, New York, NY, USA. ACM (2010)
2. Alnfai, M., Sampalli, S.: SingleTapBraille: developing a text entry method based on braille patterns using a single tap. In: The 11th International Conference on Future Networks and Communications (FNC 2016) (2016)
3. Alnfai, M., Sampalli, S.: An evaluation of SingleTapBraille keyboard: a text entry method that utilizes braille patterns on touchscreen devices. In: Proceedings of ASSETS 2016, 24–26 October 2016, Atlantis Casino Resort Spa - Reno, Nevada, US. ACM (2016)
4. what is braille. The Tennessee Council of the Blind (TCB) (2008). <http://www.acb.org/tennessee/braille.html>
5. Ruamviboonsuk, V., Azenkot, S., Ladner, R.E.: Tapulator: a non-visual calculator using natural prefix-free codes. In: Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2012), pp. 221–222. ACM, New York (2012). DOI=<http://dx.doi.org/10.1145/2384916.2384963>
6. Hesselmann, T., Heuten, W., Boll, S.: Tap2Count. In: Proceedings of ITS 2011, pp. 256–257. ACM Press, New York (2011)
7. Mattheiss, E., Georg, R., Johann, S., Markus, G., Schrammel, J., Garschall, M., Tscheligi, M.: EdgeBraille: braille-based text input for touch devices. *J. Assist. Technol.* **9**(3), 147–158 (2015)
8. Mascetti, S., Bernareggi, C., Belotti, M.: TypeInBraille: a braille-based typing application for touchscreen devices. In: Proceedings of ASSETS 2011, pp. 295–296 (2011)
9. MacKenzie, S., Soukoreff, R.W., Helga, J.: 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST 2011), pp. 471–480. ACM, New York (2011). doi:<http://dx.doi.org/10.1145/2047196.2047258>
10. Frey, B., Southern, C., Romero, M.: BrailleTouch: mobile texting for the visually impaired. In: Stephanidis, C. (ed.) UAHCI 2011. LNCS, vol. 6767, pp. 19–25. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-21666-4_3](https://doi.org/10.1007/978-3-642-21666-4_3)
11. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: BrailleType: unleashing braille over touch screen mobile phones. In: Campos, P., Graham, N., Jorge, J., Nunes, N., Palanque, P., Winckler, M. (eds.) INTERACT 2011. LNCS, vol. 6946, pp. 100–107. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-23774-4_10](https://doi.org/10.1007/978-3-642-23774-4_10)
12. Azenkot, S., Ladner, R., Wobbrock, J., Borning, A., Landay, J., Levow, G.-A.: Eyes-free input on mobile devices. ProQuest Dissertations and Theses (2014)
13. Shabnam, M., Govindarajan, S.: Braille-coded gesture patterns for touch-screens: a character input method for differently enabled persons using mobile devices. In: International Conference on Communication, Computing and Information Technology (ICCCMIT-2014) (2014)

14. Azenkot, S., Bennett, C.L., Ladner, R.E.: DigiTaps: eyes-free number entry on touchscreens with minimal audio feedback. In: Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology. ACM (2013)
15. Huffman, D.A.: A method for the construction of minimum-redundancy codes. Proc. IRE **40**(9), 1098–1102 (1952)
16. Paisios, N.: Mobile accessibility tools for the visually impaired. Ph.D. Thesis (2012). <http://cs.nyu.edu/web/Research/Theses/nektariosp.pdf>. Accessed 19 Sep 2012
17. Ladner, R.E., Kane, S.K., Wobbrock, J.O.: Usable gestures for blind people: understanding preference and performance. In: Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems. ACM (2011)