Demonstration of GestureCalc: An Eyes-Free Calculator for Touch Screens

Leah Perlmutter[1]***, Bindita Chaudhuri**[1]***, Justin Petelka**[2]**, Philip Garrison**[1]
James Fogarty[1]**, Jacob O. Wobbrock**[2]**, Richard E. Ladner**[1]
[1]Paul G. Allen School of Computer Science & Engineering, [2]The Information School
DUB Group | University of Washington, Seattle, WA 98195, USA
[1]{lrperlmu, bindita, philipmg, jfogarty, ladner}@cs.washington.edu, [2]{jpetelka, wobbrock}@uw.edu

ABSTRACT

Keypad-based character input in existing digital calculator applications on touch screen devices requires precise, targeted key presses that are time-consuming and error-prone for many screen reader users. We demonstrate *GestureCalc*, a digital calculator that uses target-free gestures for arithmetic tasks. It allows eyes-free target-less input of digits and operations through taps and directional swipes with one to three fingers, guided by minimal audio feedback. A study of the effectiveness of *GestureCalc* for screen reader users appears in a full paper by the authors at this conference [3].

CCS Concepts

•Human-centered computing → **Accessibility technologies; •Hardware** → *Touch screens;* **•Social and professional topics** → People with disabilities;

Author Keywords

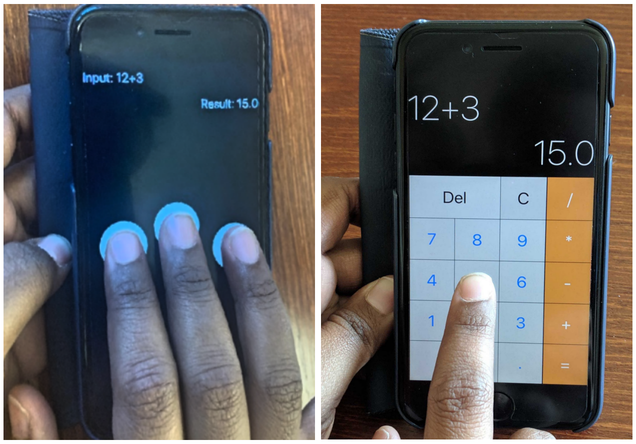
Eyes-free entry; gesture input; digital calculator; touch screen; mobile devices.

INTRODUCTION

The digital calculator is a common application that many people use on touch screen devices. Sighted people visually locate targets in the form of soft buttons and tap them to get a desired result. For people who use screen readers (e.g., VoiceOver for iOS, TalkBack for Android), finding and activating buttons in a spatial layout can be time consuming. Eyes-free solutions for accessible numeric input include Tapulator (gesture-based numeric input) [6], DigiTaps (minimal audio feedback for numeric input) [2], and BrailleTap (Braille-based gesture calculator) [1]. We demonstrate *GestureCalc*, an eyes-free target-less gesture-based touch screen calculator. A study of the effectiveness of *GestureCalc* for screen reader users appears in a full paper by the authors at this conference [3].

ASSETS '19, October 28–30, 2019, Pittsburgh, PA, USA.

ACM ISBN 978-1-4503-6676-2/19/10 ...$15.00.
http://dx.doi.org/10.1145/3308561.3354595



(a) *GestureCalc* (b) Typical touch screen calculator

Figure 1. (a) Performing a three-finger tap on our *GestureCalc* versus (b) Typing "5" on a typical touch screen calculator.

The contributions of *GestureCalc* are:

- *GestureCalc* is a novel eyes-free target-less digital calculator application that uses a minimal number of accessible gestures to enter digits and operations. Our code is available online[1] and we also plan to release the app.
- *GestureCalc* uses intuitive gestures based on conceptual metaphors that are both memorable and easily learnable for visually impaired people as suggested by our study [3].
- In a mixed methods study with eight screen reader users [3], we found that they entered characters with *GestureCalc* an average of 40.5% faster than with a typical touch screen calculator (baseline). They made more mistakes but also corrected more errors with *GestureCalc*, resulting in 52.2% fewer erroneous calculations than the baseline.

DESIGN

We build on the digit set introduced in DigiTaps [2], adding gestures for basic arithmetic operations. We only use taps and swipes as gestures since they have been found to be easiest to perform and accessible to blind users [4]. Our application: (a) allows users to interact with any part of the screen (i.e., target-less interaction), (b) requires a maximum of two gestures for every input, (c) avoids symbolic gestures

*The first two authors contributed equally to this work.

[1] https://github.com/bindita/GestureCalculator

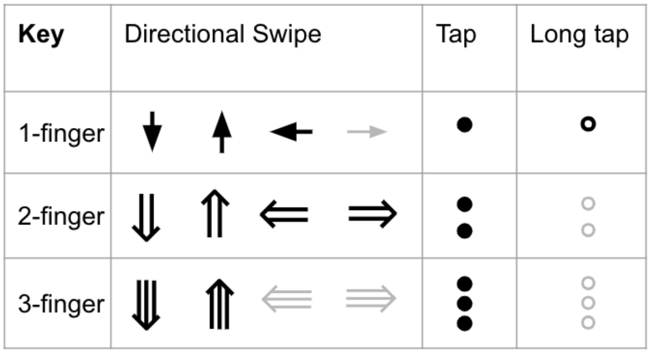


Figure 2. Symbolic representations of a superset of our gestures, with a visual shortcut for each gesture. Gestures used in *GestureCalc* are marked in black, while currently unused gestures are marked in grey.

based on printed characters, (d) avoids complex gestures by using only taps and directional swipes with one, two, or three fingers, and (e) favors intuitive gestures based on conceptual metaphors such as “up” for increasing and “down” for decreasing [5].

Gestures

Figure 2 shows the symbolic representations of the different possible gesture inputs in our design. Long tap is a variation of the tap gesture, in which users press and hold a finger against the touch screen for a short duration. We provide haptic feedback to the user from our app to indicate when the tap is held long enough (0.5 seconds) to be recognized as a long tap gesture.

Character Codes

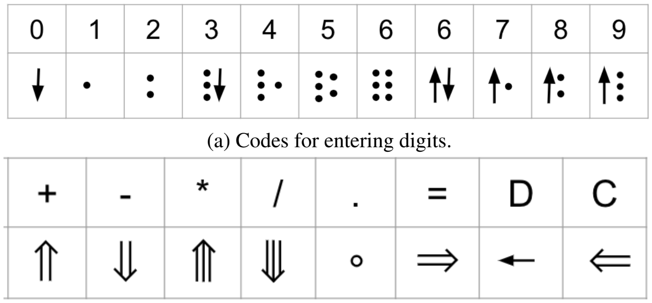
We define the term ‘characters’ as the digits (0-9) and operations that *GestureCalc* accepts as input. Our character codes are prefix-free, i.e., no character’s code is a prefix of another code. This property is important for unambiguous parsing of the input. In addition, our codes are based on conceptual metaphors which helps in remembering them.

Digits

We represent digit 0 by a one-finger downward swipe, digit 1 by a one-finger tap, and digit 2 by a two-finger tap. The digits 3, 4, 5, and 6 (i.e., the 3 block) can be represented as (3 + 0), (3 + 1), (3 + 2) and (3 + 3) respectively, hence we encode them with a three-finger tap followed by a one-finger downward swipe, a one-finger tap, two-finger tap and three-finger tap respectively. The delimiter of 0 at the end of digit 3 ensures prefix-free property. Similarly, the digits 6, 7, 8 and 9 (i.e., the 6 block) can be represented as (6 + 0), (6 + 1), (6 + 2), and (6 + 3) respectively, where the prefix 6 for these digits is represented by a one-finger upward swipe. Note that the digit 6 has two different representations. Figure 3(a) visualizes the codes for entering digits. We use an average of 1.7 gestures per digit, which is fewer than 1.8 gestures for Digitaps$^{1.8}$ and 2.5 gestures for BrailleTap.

Operations

We oriented directional swipes used in *GestureCalc* operations according to a conceptual metaphor that “more is higher” [5]. Addition increases value whereas Subtraction decreases it, so we represent ‘+’ and ‘-’ operations with a two-finger upward swipe and a two-finger downward swipe respectively. The gesture for subtraction can be used as either an operator between two operands or to negate a single operand. Multiplication implies multiple additions, and division implies multiple subtractions, so ‘*’ and ‘/’ operations are represented by a three-finger upward swipe and a three-finger downward swipe respectively. Finally, the ‘.’ (decimal point) operation is represented with a long tap.



(a) Codes for entering digits.

(b) Codes for entering operations.

Figure 3. Character codes for *GestureCalc*.

The ‘=’ (equals) operation metaphorically moves the expression forward by generating a result. Hence, it is represented by a two-finger horizontal swipe from left to right. Incidentally, this also resembles the shape of the equals symbol. When the user enters the equals operation, the application displays and speaks the result of the computation and clears the input for the next computation. The ‘D’ (delete) operation deletes one character at a time and speaks the character being deleted. In left-to-right writing systems such as Braille [7] or written English, backspace conventionally deletes a character to the left of the cursor. We therefore chose to represent this with a one-finger left swipe. The ‘C’ (clear) operation deletes all characters in the input, equivalent to multiple deletions, so it is represented by a two-finger left swipe. Figure 3(b) visualizes the codes for entering operations.

Additional Features

GestureCalc provides audio feedback (i.e., speaks the entered character) after every digit or operator is entered, similar to Digitaps. Our implementation currently supports mathematical calculations involving just two operands. A readback feature enables users to shake the device to trigger audio feedback, reading aloud the expression that has been entered. This feature is designed to be helpful for feedback while entering expressions involving long operands.

CONCLUSION

GestureCalc is a basic touch screen calculator that is based on target-free metaphoric gestures instead of buttons designed for eyes-free use. An open opportunity is how to expand the calculator to be a scientific calculator with many more operations as discussed in the associated paper [3]. Another opportunity is to explore the combination of button-based interfaces that other applications currently have and our gesture-based interface. We look forward to discussing *GestureCalc* and such additional opportunities in a poster and demo session.

REFERENCES

[1] Mrim Alnfiai and Srinivas Sampalli. 2017. BrailleTap: Developing a Calculator Based on Braille Using Tap Gestures. In *Universal Access in Human–Computer Interaction. Designing Novel Interactions*. Springer International Publishing, Cham, 213–223. DOI: http://dx.doi.org/10.1007/978-3-319-58703-5_16

[2] Shiri Azenkot, Cynthia L. Bennett, and Richard E. Ladner. 2013. 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 (UIST '13)*. 85–90. DOI:http://dx.doi.org/10.1145/2501988.2502056

[3] Bindita Chaudhuri, Leah Perlmutter, Justin Petelka, Philip Garrison, James Fogarty, Jacob O. Wobbrock, and Richard E. Ladner. 2019. GestureCalc: An Eyes-Free Calculator for Touch Screens. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '19)*. To Appear (preprint: https://bit.ly/30cWxSh).

[4] Shaun K. Kane, Jacob O. Wobbrock, and Richard E. Ladner. 2011. Usable Gestures for Blind People: Understanding Preference and Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. 413–422. DOI: http://dx.doi.org/10.1145/1978942.1979001

[5] George Lakoff and Mark Johnson. 1980. The Metaphorical Structure of the Human Conceptual System. *Cognitive science* 4, 2 (1980), 195–208. DOI: http://dx.doi.org/10.1207/s15516709cog0402_4

[6] Vaspol Ruamviboonsuk, Shiri Azenkot, and Richard E. Ladner. 2012. Tapulator: A Non-visual Calculator Using Natural Prefix-free Codes. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '12)*. 221–222. DOI:http://dx.doi.org/10.1145/2384916.2384963

[7] Wikipedia. 2018. Braille. https://en.wikipedia.org/wiki/Braille. (2018).