# Senorita: A Chorded Keyboard for Sighted, Low Vision, and Blind Mobile Users


**Gulnar Rakhmetulla**
Human-Computer Interaction Group
University of California, Merced
Merced, CA, United States
grakhmetulla@ucmerced.edu

**Ahmed Sabbir Arif**
Human-Computer Interaction Group
University of California, Merced
Merced, CA, United States
asarif@ucmerced.edu



**Figure 1. With Senorita, the left four keys are tapped with the left thumb and the right four keys are tapped with the right thumb. Tapping a key enters the letter in the top label. Tapping two keys simultaneously (chording) enters the common letter between the keys in the bottom label. The chord keys are arranged on opposite sides. Novices can tap a key to see the chords available for that key. This figure illustrates a novice user typing the word "we". She scans the keyboard from the left and finds 'w' on the 'I' key. She taps on it with her left thumb to see all chords for that key on the other side, from the edge: 'C', 'F', 'W', and 'X' (the same letters in the bottom label of the 'I' key). She taps on 'W' with her right thumb to form a chord to enter the letter. The next letter 'e' is one of the most frequent letters in English, thus has a dedicated key. She taps on it with her left thumb to complete the word.**


## ABSTRACT

Senorita is a novel two-thumb virtual chorded keyboard for mobile devices. It arranges the letters on eight keys in a single row by the bottom edge of the device based on letter frequencies and the anatomy of the thumbs. Unlike most chorded methods, it provides visual cues to perform the chording actions in sequence, instead of simultaneously, when the actions are unknown, facilitating "learning by doing". Its compact design leaves most of the screen available and its position near the edge accommodates eyes-free text entry. In a longitudinal study with a smartphone, Senorita yielded on average 14 wpm. In a short-term study with a tablet, it yielded on average 9.3 wpm. In the final longitudinal study, it yielded 3.7 wpm with blind users, surpassing their Qwerty performance. Low vision users yielded 5.8 wpm. Further, almost all users found Senorita effective, easy to learn, and wanted to keep use it.



### Author Keywords

Text input; chords; accessibility; blind; mobile; tablets.

### CCS Concepts

**•Human-centered computing → Text input; Accessibility technologies; Empirical studies in accessibility; •Social and professional topics →** *Assistive technologies;*




*CHI '20, April 25–30, 2020, Honolulu, HI, USA.*



ACM ISBN 978-1-4503-6708-0/20/04 ...$15.00.
http://dx.doi.org/10.1145/3313831.3376576


## INTRODUCTION

Virtual Qwerty augmented with linguistic and behavioral models has become the dominant input method for mobile devices. It is evident that with enough practice, one can reach a reasonable entry speed with the state-of-the-art virtual Qwerty. However, using this method is difficult in some scenarios as it takes up more than half of the screen real-estate, especially when using a smartphone in landscape position, the keys are too small for precise target selection, causing frequent typing errors (the "fat-finger problem" [49]), and the thumbs do not always reach the keys in the middle of the keyboard on larger devices, such as tablets. Although several alternatives have been proposed, Qwerty continues dominating mobile text entry since most alternatives rely on linguistic models, which make entering out-of-vocabulary words difficult, seldom impossible. Most of these also have a steep learning curve, thus require a substantial amount of time and effort in learning, encouraging users to stick to the method they are already familiar with [31]. Besides, users tend to discard a new solution if they cannot "learn by doing" [13] and are not immediately convinced that the performance and usability gain will worth the effort [14].

These make learning and using virtual Qwerty even more difficult for users with low vision and blindness[1]. Today, mobile devices are not a luxury, but essential for productivity, entertainment, and to keep in touch with loved ones. A large part of these activities requires text entry. Visually impaired users struggle to engage in these due to the absence of effective eyes-free text entry techniques for mobile devices. Most existing so-

[1]This article refers to people with *severe low vision* to *near-total blindness* with a visual acuity between 20/200 and 20/1,200 as "low vision" and people with *no light-perception* as "blind" [50]. Both groups require assistive technology to use mobile devices.

lutions rely on braille, when braille literacy in this population is merely 10% [36]. Other alternatives are time-consuming, error-prone, and have a high learning curve. Speech-to-text is becoming increasingly reliable, but not effective in loud environments [15] and compromises privacy and security [17].

Although the aforementioned scenarios may seem unrelated on the surface, a compact layout that does not occupy most of the screen, has larger keys to accommodate precise target selection, provides comfortable reach to all keys, facilitates learning, and enables the entry of out-of-vocabulary words is desired in all. Senorita is a two-thumb chorded keyboard[2] designed to meet these needs (Figure 1). It enables visually impaired people to enter text on mobile devices, and is effective in circumstances where sighted users are unable to see, such as when wearing surgical eye patches or have lost prescription eyeglasses. It is also a "comfortable" alternative to Qwerty in situations where comfort is more desired than speed, such as while commenting on a video or texting while walking. Its design is motivated by the "design for all" philosophy to accommodate the maximum possible group of users [42, 44].

The remainder of the paper is organized as follows. First, we discuss the design of Senorita and argue the benefits of chording. We summarize related work in the area. We then present results of a longitudinal study evaluating the keyboard on a smartphone. Then, we customize Senorita for larger devices and evaluate it on a tablet. Finally, we discuss the design modifications for low vision and blind users, and present results of a longitudinal study evaluating Senorita with the target user group. We conclude with speculations on future extensions.

## DESIGNING SENORITA

Senorita is a novel two-thumb chorded keyboard with eight keys laid out in a single row (Figure 1). The left four keys are tapped with the left thumb and the right four keys are tapped with the right thumb. It assigns dedicated keys for the most frequent eight letters in the English language: 'E', 'A', 'I', 'S', 'R', 'N', 'O', and 'T' [27]. The name "Senorita" is an anagram of these letters. To enter these letters, users tap once on the respective keys. The remaining eighteen letters are entered by performing chords. Each chord is composed of two keys from the opposite sides of the keyboard. However, the chord keys for the most infrequent letters 'J' and 'Z' are placed on the same side, the furthest from each other to ensure easier reach. Senorita dynamically changes size based on the screen width and the average thumb length (7.65 cm [40]) to ensure that these keys are never out of reach. In a pilot, users were able to enter these two letters on a smartphone in landscape position without any difficulty. The bottom label of each key displays the chords it can construct. Tapping two keys simultaneously enters the common letter between these keys. For example, tapping 'I' and 'N' together enters the letter 'w' (Figure 1).

The layout was designed keeping letter frequencies and the anatomy of the thumbs in mind. First, we used a linguistic table for the relative frequencies of all letter pairs in the English alphabet to sort all letters by frequency [27]. We then placed the most frequent letters closer to the side bezels to make them easier for the thumbs to reach [6, 38] and the infrequent letters further away to reduce thumb movement. The chords are composed of keys from the opposite sides to make them easier to enter with the thumbs. We only considered the horizontal reach of the thumbs to leave the upper area of the screen unoccluded and for the users to use the bottom bezel as a physical reference to facilitate eyes-free text entry. Using the side bezels as physical reference is inconvenient since reaching the topmost and the nethermost keys require extending the thumbs too far [38]. This convention resulted in two possible layouts: Senorita and flipped-Senorita that had the keys switch sides (left keys to the right and vice versa). In a pilot, both keyboards yielded comparable results, yet we selected Senorita since its key arrangement match the most with Qwerty, which may make the transition to Senorita easier for some users. Notice in Figure 1 that the letters in the bottom labels appear in reversed order than the order in which the keys appear on the opposite side. For example, the 'A' key shows the chords as 'LUYB', while the keys on the other side appear in the order of 'BYUL'. This is because we arranged the letters in the labels and the keys from the edges. While it may seem counterintuitive, a pilot revealed that users prefer this design as they naturally scan the letters from the edges, like reading a book. It also facilitates eyes-free text entry since visually impaired users tend to slide their thumbs from the edges. Currently, Senorita does not support the entry of upper-case letters, special symbols, numbers, or languages other than English. But support for these could be easily added by enabling the user to long-press, double-tap, or dwell over a key to switch back and forth between the cases, or change the layout to enter digits and symbols.

Senorita provides visual feedback to facilitate learning. The two sides are distinguished using two different shades of grey. The top and bottom labels in each key represent tap and chord-based keys, respectively. When the user taps a key, Senorita displays only the possible chords on the other side (Figure 1). This enables novices to press two keys in sequence, instead of simultaneously. Hence, unlike most chorded methods, Senorita does not require users to learn the chords before using it, instead allows them to learn the chords as they type.

### Benefits of Chording

Once learned, performing chords is much faster than tap sequence [51]. Stenotype is a chorded method still in use by court reporters as expert stenotyping speed is much faster than expert Qwerty [33]. The main challenge of existing chorded methods is that they require substantial time and effort to master. Thus, most existing chorded keyboards lack in immediate usability and require users to learn the chords before typing. Senorita attempts to mitigate this by letting users enter the most frequent letters with a single tap, using only two keys per chord (most methods use more), and letting user either perform the chording actions simultaneously or in sequences from either side of the keyboard to enter the chorded letters. The goal was to enable users to start typing immediately and learn the chords along the way. Using chords also enabled us to fit the English alphabet in eight large keys, which facilitates precise target selection and saves precious touchscreen real-estate.

[2]With chorded keyboards, users press multiple keys simultaneously to form a chord (like playing a chord on piano) to enter a letter.

It also eliminated the need for using a decoder to disambiguate input, providing the support for out-of-vocabulary words.

## RELATED WORK

This section reviews reduced-size and split virtual keyboards, and text entry solutions for visually impaired people. It does not discuss speech-to-text and physical/non-touchscreen solutions since these are outside the scope of this work.

### Reduced-size Keyboards

While numerous works have focused on developing linguistic and behavioral models [6, 18, 19, 29, 48] for faster and more accurate text entry with virtual Qwerty and designed novel keyboard layouts that are comparable to Qwerty in size [9, 34], not many have focused on reduced-size virtual keyboards. Romano et al. [41] designed a single-row tap-slide hybrid keyboard for mobile devices, but did not evaluate it empirically. Stick Keyboard [20] maps four rows of a standard Qwerty onto the home row. Although designed for smartphones, it was evaluated on a desktop using a physical prototype, where it yielded 10.4 wpm without the support of a linguistic model. 1Line Keyboard [30] is a similar virtual keyboard, designed for tablets. With the support of a statistical decoder, it yielded 30.7 wpm in a longitudinal study. Gueorguieva et al. [21] designed a reduced virtual keyboard to enable text entry through Morse code. In a user study, it reached 7 wpm by the seventh session without the support of a linguistic model. In a different work, Zhu et al. [53] showed that with sufficient practice, expert virtual Qwerty users can reach 37.9 wpm with an invisible Qwerty keyboard augmented with a statistical decoder.

### Split Keyboards

Some have explored split keyboards for tablets, which require both vertical and horizontal movement of the thumbs. Yazdi et al. [1] optimized splitting of a Qwerty keyboard for comfortable thumb movement. In a user study, it yielded 27.6 wpm. Bi et al. [8] enabled gesture typing with both thumbs on a split Qwerty. With the support of a statistical decoder, this method reached 26 wpm. KALQ [38] is a novel split keyboard optimized for thumb movements. It also uses a statistical decoder, and reached 37.1 wpm in a user study.

| Method | Speed |
|---|---|
| BrailleTouch [46] | 9.40–23.20 wpm |
| TypeInBraille [35] | 6.30 wpm |
| Mobile Braille [12] | 5.05 wpm |
| SingleTapBraille [2] | 4.71 wpm |
| BrailleType [37] | 1.49 wpm |

**Table 1. Entry speed of popular braille-based virtual keyboards.**

### Keyboards for Visually Impaired People

Most text entry solutions for visually impaired people rely on the knowledge of braille (Table 1), when braille literacy in this population is only 10% [36]. Alternative solutions include Escape-Keyboard [7] that enables entering letters by pressing the thumb on different areas of the screen, then performing directional strokes. It used a statistical decoder to improve its performance. In a longitudinal study, sighted participants

**Figure 2. Two volunteers participating in the first study with a smartphone in landscape position.**

yielded 14.7 wpm in an eyes-free condition. A similar method, ThumbStroke [26], yielded 10.5 wpm in a longitudinal study with sighted participants. No-Look Notes [11] arranges all letters in an eight-segment pie menu. Users perform a series of taps and stokes with both hands to enter a letter. In a study, it reached 1.67 wpm. EdgeWrite [52] enables entering letters by *"traversing the edges and diagonals of a square hole"* imposed over the screen. Although initially designed for people with motor impairments, it enables eyes-free text entry [25]. In a study, it yielded 6.6 wpm with able-bodied sighted participants. NavTouch [23] lets users navigate the alphabet by performing directional gestures using the vowels as anchors. In a study, blind users reached 1.72 wpm with this method on a smartphone. SpatialTouch [22] is a virtual Qwerty that exploits users' experience with physical Qwerty to enable eyes-free text entry through multi-touch exploration and spatial, simultaneous auditory feedback. In a study, blind participants yielded 2–3 wpm with this method. AGTex [10] is a screen-reader supported virtual Qwerty that enables eyes-free gesture typing by switching between two modes. In a study, blind users reached 5.66 wpm. Some have also evaluated multi-tap with blind participants [37], where it yielded 2 wpm.

## USER STUDY 1: SMARTPHONE

We conducted a longitudinal study to evaluate Senorita on a smartphone in landscape position.

### Apparatus

We used a Motorola Moto $G^5$ Plus smartphone ($150.2 \times 74 \times 7.7$ mm, 155 g) at $1080 \times 1920$ pixels. A custom application was developed using the Android Studio 3.1, SDK 27 to record all performance metrics and interactions with timestamps.

### Participants

Ten sighted volunteers aged from 20 to 31 years (M = 23.8, SD = 3.46) participated in the study (Figure 2). Three of them were female and seven were male. They all were experienced users of virtual Qwerty (M = 8.0 years of experience, SD = 3.44). Six of them rated themselves as native/bilingual-level, one rated herself as advanced-level, and three rated themselves as moderate-level English speakers. None of them had prior experience with chorded keyboards. Eight of them were right-handed and two were left-handed. They all received US $50 for participating in the study.

### Design

The study used a within-subjects design, where the independent variables were the session and method and the dependent variables were the performance metrics. We recorded the commonly used *words per minute (wpm)* and *error rate* [4, 45] metrics to measure speed and accuracy, respectively. We also recorded *chording rate*, which is the average percentage of chords performed per session. If there were 300 chords available in a session and the users performed 90, then the chording rate for the session is 30%. This metric was calculated only for Senorita as Qwerty does not have chords. In summary, the design was as follows.

| **Qwerty** | **Senorita** |
|---|---|
| 10 participants × | 10 participants × |
| 1 session (first day) × | 10 sessions (different days) × |
| 15 random phrases [32] | 15 random phrases [32] |
| = 150 phrases. | = 1,500 phrases. |

## Procedure

The study was conducted in a quiet room. On the first day (Session 1), we explained the research to all participants and collected their consents and demographics. We then started the Qwerty condition, where participants transcribed 15 random English phrases from a set [32] using the default Android Qwerty. All predictive features of the keyboard were disabled to eliminate a potential confound. The purpose of this condition was to record participants' speed and accuracy with Qwerty. We did not include this condition in the following sessions since they all were experienced virtual Qwerty users. Then, we demonstrated Senorita and enabled all participants to practice with it by transcribing two random phrases. These phrases were not repeated in the study. They then completed a System Usability Scale (SUS) [47] inspired pre-study questionnaire that asked them to rate various aspects of Senorita on a 7-point Likert scale. Its purpose was to record their immediate impression of the keyboard. The Senorita condition started after that, where participants transcribed 15 phrases using the keyboard.

Both conditions presented one phrase at a time. Participants were instructed to memorize the phrase, transcribe it *"as fast and accurate as possible"*, then tap the NEXT key to see the next phrase. Error correction was recommended but not forced. All participants held the device in landscape position (Figure 2). They could take short breaks between the phrases, when needed. Logging started from the first keystroke and ended when participants pressed NEXT. The sessions were scheduled on different days, with at most a two-day gap in between. All sessions followed the same procedure, except for the Qwerty condition and the practice block, which were exclusive to Session 1. Upon completion of the study, participants completed a short post-study questionnaire that included the same questions as the pre-study questionnaire. The goal was to find out if practice influenced participants' impression of the keyboard.

## Results

A Shapiro-Wilk test and a Mauchly's test indicated that the response variable (metrics) residuals are normally distributed and the variances of populations are equal, respectively. Hence, we used a repeated-measures ANOVA for all analysis. Only the performance of the last session (Session 10) was considered to compare Senorita with Qwerty.

### *Entry Speed*

An ANOVA identified a significant effect of session on entry speed ($F_{9,9} = 47.9, p < .0001$). A Tukey-Kramer test revealed three distinct groups: 1–4, 5–7, and 8–10. An ANOVA also identified a significant effect of method ($F_{1,9} = 243.02, p < .0001$). Average entry speed with Qwerty and Senorita were 32.7 wpm (SD = 8.9) and 13.99 (SD = 3.3) wpm, respectively. Figure 3 shows average entry speed per session with Senorita.

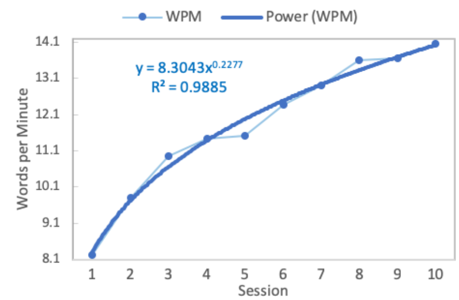


**Figure 3. Average entry speed (wpm) per session with Senorita on a smartphone, fitted to a power trendline.**

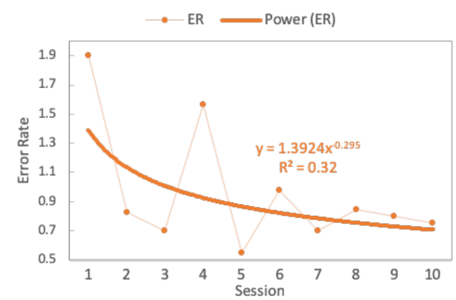


**Figure 4. Average error rate per session with Senorita on a smartphone, fitted to a power trendline.**

### *Error Rate*

An ANOVA failed to identify a significant effect of session on error rate ($F_{9,9} = 0.95, p = .48$). There was also no significant effect of method ($F_{1,9} = 1.40, p = .27$). Average error rate with Qwerty and Senorita were 0.92% (SD = 1.67) and 0.89% (SD = 1.32), respectively. Figure 4 displays average error rate per session with Senorita.

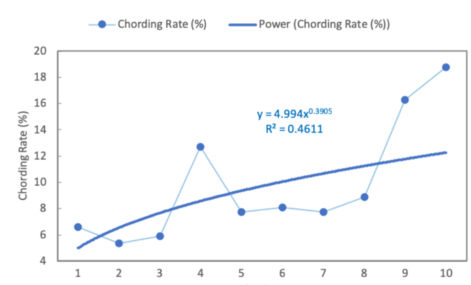


**Figure 5. Average chording rate per session with Senorita on a smartphone, fitted to a power trendline.**

### *Chording Rate*

An ANOVA failed to identify a significant effect of session on chording rate ($F_{9,9} = 1.74, p = .09$). Average chording rate in all sessions was 9.71% (SD = 18.3). The most frequent chords in the study were 'h' (31.1%) and 'l' (28.6%). Figure 5 displays average chording rate per session.

### User Feedback

A Wilcoxon Signed-Rank test revealed that user opinion about Senorita changed significantly in regard to willingness to use ($z = 2.39, p < .05$) and learnability ($z = -2.5, p < .05$) after using it. But no significant effects were identified on ease of use ($z = -6.8, p = .09$), perceived speed ($z = -0.7, p = .4$), and perceived accuracy ($z = -1.6, p = .1$). Figure 6 illustrates median user ratings of all investigated aspects of the keyboard.

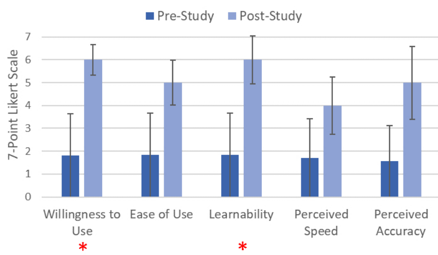


**Figure 6. Median user ratings of willingness to use, ease of use, learnability, perceived speed, and perceived accuracy of Senorita on a 7-point Likert scale, where '1' to '7' signify 'strongly disagree' to 'strongly agree'. The error bars represent $\pm$1 standard deviation (SD) and the red asterisks signify statistical significance.**

### Discussion

Senorita yielded a competitive entry speed. Similar virtual keyboards for smartphones yielded a maximum of 11 wpm with sighted participants [26] compared to Senorita's 14 wpm. Senorita was significantly slower (60%) than Qwerty. But this speed was achieved with only a 19% chording rate, thus likely to improve with sufficient practice. The average entry speed over session correlated well ($R^2$ = 0.9885) with the power law of practice [43]. The fact that learning occurred even in the last session indicates that Senorita did not reach its highest possible speed in the study. Learning of chords occurred at a slower rate, presumably because participants had the choice of performing the chording actions in sequence, instead of simultaneously. Prior studies showed that users learn a new method faster in the absence of a reliable alternative [5]. While forcing users to use chords could have improved entry speed, compromised the keyboard's immediate usability. Senorita's error rate was comparable to Qwerty (< 1%). There was no significant difference in error rate between the sessions, which suggests that Senorita was moderately accurate from the start, and unlikely to get more accurate with practice. User feedback revealed that initially most participants felt that learning and using Senorita will be difficult, slower, and more error-prone, hence did not show much interest in using it on mobile devices. However, their responses were much positive after practice (Figure 6). Particularly, their opinion about the learnability of the keyboard and willingness to use it on mobile devices were significantly more positive. Most of them felt that Senorita could be an effective alternative to Qwerty in special scenarios.

## USER STUDY 2: TABLET

We conducted a study to evaluate Senorita on a tablet. We made two design adjustments for this. The two sides of the keyboard split based on the average thumb length (7.65 cm [40]) and the 'J' and 'Z' keys switch sides when the first key of a chord is tapped. These were to ensure that the thumbs can comfortably reach all keys (Figure 7) and most of the touchscreen is available for the users to engage in other activities.

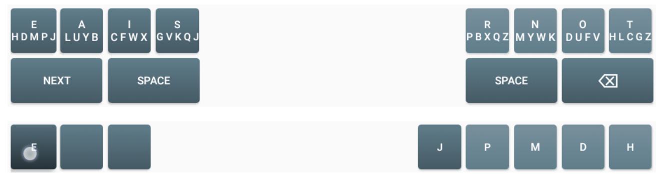


**Figure 7. Senorita dynamically resizes, splits, and the 'J' and 'Z' keys jump sides when chording (bottom) to assure a comfortable reach to all keys on larger devices that are usually held with two hands.**

### Apparatus

We used a Samsung Galaxy Tab A (212.09×124.206×8.89 mm, 690 g) at 1280×800 pixels. A custom application developed with the Android Studio 3.1, SDK 27 recorded all metrics and interactions with timestamps (Figure 8).

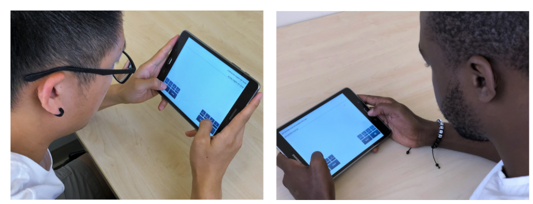

**Figure 8. Two participants taking part in the second study with a tablet.**

### Participants

Ten new sighted participants aged from 24 to 30 years (M = 26.8, SD = 1.99) took part in the study (Figure 8). Two of them were female and eight were male. They all were experienced users of mobile Qwerty (M = 7.5 years of experience, SD = 2.22). Four of them rated themselves as advanced-level and three rated themselves as moderate-level English speakers. None of them had prior experience with chorded keyboards. Nine of them were right-handed and one was left-handed. They all received US $10 for volunteering.

### Design

The study used a within-subjects design, where the independent variable was block and method, and the dependent variables were the performance metrics. We recorded the same metrics as the first user study. In summary, the design was:

| **Qwerty** | **Senorita** |
|---|---|
| 10 participants × | 10 participants × |
| 1 block × | 5 blocks × |
| 10 random phrases [32] | 10 random phrases [32] |
| = 100 phrases. | = 500 phrases. |

## Procedure

This user study was conducted in a quiet room. We started by explaining the procedure to all participants and collecting their consents and demographics. We then started the Qwerty condition, where participants transcribed 10 random English phrases from a set [32] using the default Android Qwerty. The WebTEM [3] application was used to record all performance metrics. All predictive features of the keyboard were disabled to eliminate a potential confound. We then introduced Senorita and enabled all to practice with it by transcribing two random phrases. These phrases were not repeated in the study. Participants then completed a System Usability Scale (SUS) [47] inspired short pre-study questionnaire that asked them to rate various aspects of Senorita on a 7-point Likert scale. Its purpose was to record the participants' initial impression of the keyboard. The Senorita condition started after that, where we instructed participants to transcribe fifty random phrases in five blocks ($10 \times 5$). Both conditions presented one phrase at a time. Participants were asked to memorize the phrase, transcribe it *"as fast and accurate as possible"*, then tap the NEXT key to see the next phrase. Error correction was recommended but not forced. Participants could take short breaks between phrases, when needed. Logging started from the first keystroke and ended when users NEXT. Once done, they all completed a short post-study questionnaire that included the same questions as the pre-study questionnaire to investigate if practice influenced participants' impression of the keyboard.

## Results

A Shapiro-Wilk test and a Mauchly's test indicated that the response variable residuals are normally distributed and the variances of populations are equal, respectively. Hence, we used a repeated-measures ANOVA for all analysis. Only the performance of the last block was used for the statistical tests.

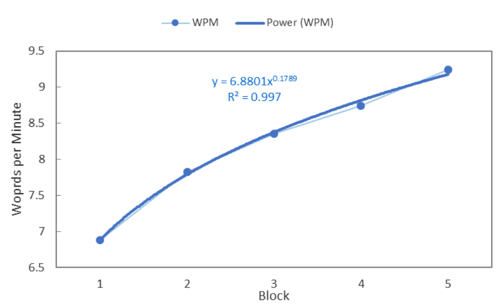


**Figure 9. Average entry speed (wpm) per block with Senorita on a tablet, fitted to a power trendline.**

### *Entry Speed*

An ANOVA identified a significant effect of block on entry speed ($F_{4,9} = 7.57, p < .0005$). There was also a significant effect of method ($F_{1,9} = 16.55, p < .005$). Average speed with Qwerty and Senorita were 27.18 wpm (SD = 8.23) and 9.27 wpm (SD = 2.96), respectively. Figure 9 displays average entry speed per block with Senorita.

### *Error Rate*

An ANOVA failed to identify a significant effect of block on error rate ($F_{4,9} = 0.70, p = .6$). There was also no significant effect of method ($F_{1,9} = 0.04, p = .85$). Average error rate with Qwerty and Senorita were 2.27% (SD = 0.68) and 2.16% (SD = 4.56), respectively. Figure 10 displays average error rate per block with Senorita.

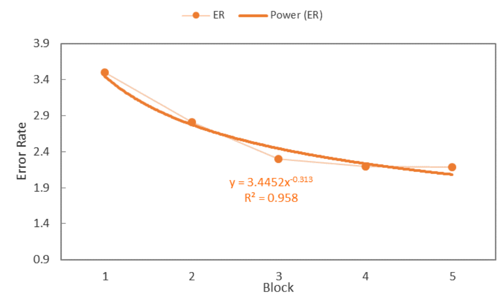


**Figure 10. Average error rate per block with Senorita on a tablet, fitted to a power trendline.**

### *Chording Rate*

An ANOVA failed to identify a significant effect of block on chording rate ($F_{4,9} = 0.43, p = .78$). Average chording rate in all sessions was 9.4% (SD = 16.9). As the previous study, the most frequent chords were 'h' (35.6%) and 'l' (12.4%). Figure 11 displays average chording rate per session.

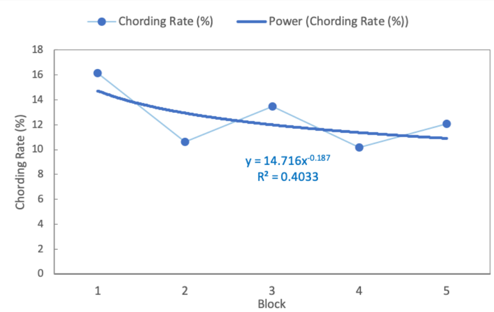


**Figure 11. Average chording rate per block with Senorita on a tablet, fitted to a power trendline. No clear trend is visible here, which is not surprising considering the short duration of the study.**

## User Feedback

A Wilcoxon Signed-Rank test failed to identify any significant change in user opinion about Senorita in regard to willingness to use ($z = 0.877, p = .4$), ease of use ($z = 0.857, p = .4$), learnability ($z = 0.33, p = .7$), perceived speed ($z = 1.56, p = .1$), or perceived accuracy ($z = 0.877, p = .4$). Figure 12 displays median user ratings of all explored aspects of Senorita.

## Discussion

Participants reached a 9.3 wpm entry speed and a 2.2% error rate in a single session (10 phrases), which is comparable to Senorita's 8.23 wpm and 2% error rate in Session 1 of the first study (15 phrases). We find this inspiring since smartphone keyboards tend to reduce in speed and accuracy on tablets [1, 8]. It is also promising that learning occurred in this short-term study. Compared to the first block, entry speed with Senorita increased by 23% in the last block. Like the first study, the

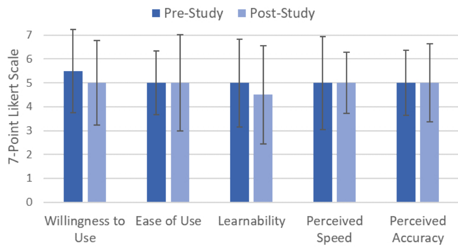


**Figure 12. Median User ratings of Senorita's willingness to use, ease of use, learnability, perceived speed, and perceived accuracy on a 7-point Likert scale, where '1' to '7' signify 'strongly disagree' to 'strongly agree'. The error bars represent ±1 standard deviation (SD).**

average entry speed over block correlated well ($R^2 = 0.997$) with the power law of practice [43]. Learning occurred even in the last block, which suggest that Senorita's entry speed on tablet is likely to increase substantially with practice. As expected, Senorita was significantly slower (66%) than Qwerty. But this speed was achieved with only a 9.4% chording rate. Hence, we speculate that with practice both chording rate and chording time will increase, resulting in a higher entry speed.

The 1Line Keyboard [30] that also maps all letters to a single row performed much better on a tablet than Senorita. However, its 30.7 wpm entry speed was achieved with the support of a linguistic model. This keyboard relies on a decoder to disambiguate the input, which makes entering out-of-vocabulary words difficult. An almost identical method [20] yielded a 10.4 wpm without the support of a linguistic model, suggesting that the former's performance gain was due to its predictive system. Senorita's performance could also improve substantially when augmented with word prediction and auto-correction.

Interestingly, participants were on board with Senorita from the start. Most felt that Senorita will be easy to learn and use, improve their entry speed and accuracy, thus wanted to use it on tablets even before trying it. Their impression of Senorita did not deviate much after practice. This finding highlights users' frustration with Qwerty on larger devices like tablets. Accordingly, most participants felt that it can be an effective alternative to Qwerty is special circumstances.

## EYES-FREE SENORITA

We used an iterative design process to fine-tune Senorita for visually impaired people. First, we customized the design based on the existing literature and design guidelines (e.g., [24, 28, 39]), then evaluated it in pilot studies involving blindfolded sighted participants. We refined the design based on the findings. This process was repeated until the design reached a satisfactory level. We then tested Senorita with a representative user group, where five low vision and blind participants (2 female, 3 male) used Senorita on a smartphone, then took part in a focus group discussing the challenges in mobile text entry, and the design of the keyboard. All participants were supportive of Senorita from the start, but suggested some design modifications. They all appreciated that Senorita has only eight keys, thus does not require scanning their thumbs through an array of keys. They also found it ergonomic as it does not require stretching their thumbs vertically. They all expressed their frustration with the existing solutions and were willing to invest time and effort in learning a new keyboard if it is effective and user-friendly. We went through another design iteration to address the feedback from the focus group. The final design (Figure 13) made the following modifications.

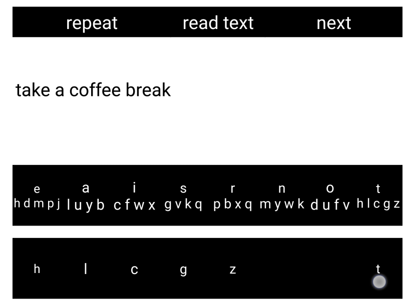


**Figure 13. Senorita dynamically resizes the layout, splits the two sides (when needed), and switches positions of the 'J' and 'Z' keys when the first key of a chord is tapped (bottom) to make sure that the thumbs can comfortably reach all keys when holding a device with two hands.**

### Hybrid and Adaptive Design

We added a high-contrast black-and-white theme to provide visual aids to low vision users with some light-perception [28]. We removed all inactive areas between the keys (known as "gutter") to enable smooth sliding between the keys without lifting the thumbs. We then removed the letters 'J' and 'Z' from the 'S' and 'R' keys, and kept them only on the 'E' and 'T' keys, as visually impaired users found their placement in both the edge and center keys confusing. This is interesting since sighted users did not complain about it. Finally, we replaced the SPACE and BACKSPACE keys with directional strokes, enabling users to enter a space by performing a *right swipe* and delete a letter by performing a *left swipe* anywhere on the screen. While we acknowledge that these gestures could interfere with underlying apps, this can be addressed by using a simple behavioral model. Participants could also *left swipe and hold* for repeated backspaces (like press-holding the BACKSPACE key). Not only this made the design more intuitive but brought the keys closer to the bottom bezel, enabling users to use it as a physical reference (slide their thumbs along the edge).

### Screen Reader

We augmented a screen reader to Senorita to provide auditory feedback on each input and interaction. This feature was implemented using the Android SDK 27's `TextToSpeech` class [16]. It uses a female voice (`Voice I`) from seven available options in default pitch and 2x speed, based on the feedback from the focus group. When a thumb slides over the keys, it reads the letters on the keys as the thumb touches them. For example, when the thumb touches the 'T' key (Figure 13, bottom), it reads 'T', pauses for a brief moment, then reads 'HLCGZ'. Lifting the thumb enters the first letter 'T'. When the other thumb touches the keys from the other side, Senorita reads the chorded letters on the respective keys, specifically 'H', 'L', 'C', 'G', and 'Z'. Lifting the finger on one of these keys enters the respective letter. When the thumb leaves a key, Senorita starts reading the next key immediately, without keeping the user waiting. This enables novices to slowly slide over the keys to find the target letter, while experts could swiftly slide to the intended key. Senorita confirms all input, such as *"A entered"* or *"B deleted"*. When space is entered, it reads the last entered word for the user to verify the input. The user could also press the READ TEXT key or *swipe down* anywhere on the screen to hear what has been typed so far. In pilots, we did not receive any complaints from the participants about not being able to comprehend the spoken letters or an increased cognitive load due to information overload.

## USER STUDY 3: LOW VISION AND BLIND USERS

We evaluated the refined eyes-free design with a representative user group in a longitudinal user study.

### Apparatus

We used a Motorola Moto $G^5$ Plus smartphone ($150.2 \times 74 \times 7.7$ mm, 155 g) at $1080 \times 1920$ pixels. A custom application was developed using the Android Studio 3.1, SDK 27 to record all metrics and interactions with timestamps.

**Figure 14. A low vision person (left) and a blind person (right) participating in the third user study.**

### Participants

Initially, we recruited thirteen participants through the Center of vision Enhancement (COVE) in Merced, CA. But two participants withdrew from the study due to personal reasons. Eleven participants, three blind people (2 female, 1 male, M = 52 years, SD = 7) and eight low vision[1] people (6 female, 2 male, M = 38.13 years, SD = 11.24), took part in the study. Their age ranged from 27 to 62 years (M = 41.91, SD = 11.84). Eight of them were female and three were male. Nine of them rated themselves as native/bilingual and two rated themselves as advanced-level English speakers. They all were frequent users of virtual Qwerty with a screen reader (M = 6.93 years of experience, SD = 3.15). They all expressed their frustration with Qwerty (P5, female, 57 years, *"I hate using Qwerty, it is slow and frustrating"*). Some of them also used Bluetooth keyboards (N = 3) and speech-to-text (N = 5) to enter text on mobile devices. Using these methods, they composed on average 12.9 text messages per day (SD = 7.6). Nine participants were right-handed, one left-handed, and one was ambidextrous. Five of them were Braille literate. COVE arranged transportation to the study for those who needed it. They all received a US $60 gift card for volunteering.

### Design

We used a within-subjects design, where the independent variable were the session and method and the dependent variables were the performance metrics. We recorded the same metrics as the previous studies. In summary, the design was as follows.

| **Qwerty** | **Senorita** |
|---|---|
| 11 participants × | 11 participants × |
| 1 session × | 6 sessions × as many random |
| 10 random phrases [32] | phrases [32] as possible in 20m |
| = 100 phrases. | = 887 phrases. |

### Procedure

The study was conducted in a quiet room at COVE. We shared the informed consent form with potential volunteers ahead of time for them to learn about the research. Upon arrival, we explained the procedure again, responded to any questions they had, then asked them to verbally consent to participate in the study. A sighted employee of COVE was present to witness this process and sign the consent form on their behalf. Participants then responded to a demographics and mobile usage questionnaire, where we read the questions and the options to them, and recorded all responses. The first condition started after that, where we asked participants to pick any virtual keyboard of their choice to transcribe 15 random phrases from a set [32]. They all chose Qwerty with a screen reader. We disabled all predictive features of this keyboard to eliminate a potential confound. We used the WebTEM [3] app to record all metrics. We did not include this condition in the following sessions since all participants were experienced in virtual Qwerty. Then, we demonstrated Senorita and enabled participants to practice with it by transcribing two random phrases, which were not repeated in the study. The Senorita condition started after that, where participants were instructed to transcribe as many phrases as possible with Senorita in 20 minutes. Both conditions read one phrase at a time. Participants were instructed to memorize the phrase, transcribe it *"as fast and accurate as possible"*, then tap the NEXT key to see the next phrase. Participants could hear the phrase again by either pressing the REPEAT key (Figure 13) or *swiping up* anywhere on the screen, and SWIPE DOWN to hear what they have entered so far. Error correction was recommended but not forced. Participants held the device in landscape position (Figure 14). They could take short breaks between the phrases, when needed. Logging started from the first keystroke and ended when participants pressed NEXT. The sessions were scheduled from 90 minutes to 24 hours apart, with a maximum of three sessions per day. All sessions followed the same procedure, except for the Qwerty condition and the practice block, which were exclusive to Session 1. All sessions were video-recorded for further analysis. When done, participants took part in a brief interview session where they were asked to comment on the performance and usability of Senorita.

### Results

A Shapiro-Wilk test and a Mauchly's test indicated that the response variable residuals are normally distributed and the variances of populations are equal, respectively. Hence, we used a repeated-measures ANOVA for all analysis. Only the performance of the last session was considered for the statistical tests.

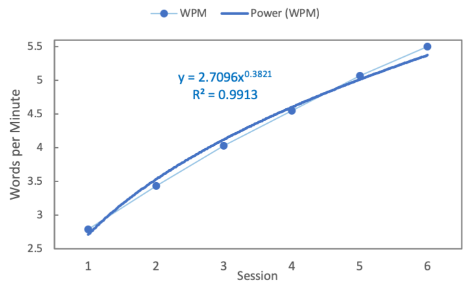


**Figure 15. Average eyes-free text entry speed (wpm) per session with Senorita, fitted to a power trendline.**

*Entry Speed*
An ANOVA identified a significant effect of session on entry speed ($F_{5,10} = 40.79, p < .0001$). A Tukey-Kramer test revealed two distinct groups: 1–4 and 5–6. An ANOVA also identified a significant effect of method ($F_{5,10} = 6.62, p < .05$). Average entry speed with Qwerty and Senorita were 11.4 wpm (SD = 9.1) and 5.5 wpm (SD = 1.64), respectively. Further, a One-way ANOVA identified a significant effect of level-of-sight on entry speed for both Qwerty ($F_{1,108} = 61.06, p < .0001$) and Senorita ($F_{1,885} = 219.2, p < .0001$). Low vision and blind participants yielded on average 14.73 wpm (SD = 8.52) and 2.52 wpm (SD = 0.8) with Qwerty, respectively, and 5.8 wpm (SD = 1.55) and 3.69 wpm (SD = 0.85) with Senorita, respectively. Figure 15 displays average entry speed per session with Senorita.

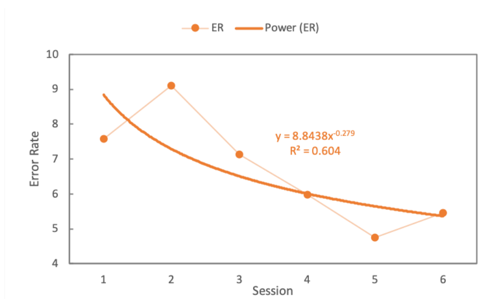


**Figure 16. Average eyes-free error rate per session with Senorita, fitted to a power trendline.**

*Error Rate*
An ANOVA identified a significant effect of session on error rate ($F_{5,10} = 2.58, p < .05$). A Tukey-Kramer revealed two distinct groups: 1–4 and 5–6. An ANOVA also identified a significant effect of method ($F_{1,9} = 243.02, p < .0001$). Average error rate with Qwerty and Senorita were 8.5% (SD = 25.2) and 5.46% (SD = 8.4), respectively. A One-way ANOVA failed to identify a significant effect of level-of-sight on error rate for Qwerty ($F_{1,108} = 0.02, p = .88$), but identified a significant effect for Senorita ($F_{1,885} = 54.35, p < .0001$). Low vision and blind participants yielded on average 8.26% (SD = 28.53) and 9.06% (SD = 12.91) error rate with Qwerty, respectively, and 4.2% (SD = 6.52) and 13.24% (SD = 13.36) error rate with Senorita, respectively. Figure 16 displays average error rate per session with Senorita.

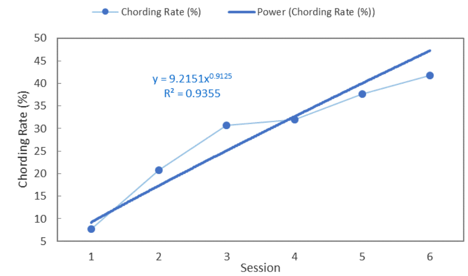


**Figure 17. Average eyes-free chording rate per session with Senorita, fitted to a power trendline.**

*Chording Rate*
An ANOVA identified a significant effect of session on chording rate ($F_{5,10} = 18.30, p < .0001$). Average chording rate in all sessions was 25.01% (SD = 21.45). A One-way ANOVA identified a significant effect of level-of-sight on chording rate ($F_{1,64} = 4.95, p < .05$). Average chording rate for low vision and blind participants were 23.74% (SD = 4.16) and 43.52% (SD = 22.89), respectively. As the previous studies, the most frequent chords were 'h' (82.2%) and 'l' (61.2%). Figure 17 displays average chording rate per session with Senorita.

**Discussion**
Session had a significant effect on entry speed and chording rate. Besides, average entry speed and chording rate over session correlated well ($R^2$ = 0.9913 and 0.9355, respectively) with the power law of practice [43]. These and the fact that learning occurred even in the last session (Figure 15, 17) suggest that Senorita did not reach its highest possible speed in the study, and likely to get much faster with practice. Note that we did not observe a significant effect of session or block on chording rate in the prior studies. Visually impaired users yielded a 62% higher chording rate than the previous studies although they were not forced to use chords. They received auditory feedback for sequential input when they pressed on one key and ran their thumb across the other keys. But this process was time-consuming since as novices they had to hear all letters. Sighted users, in contrast, could tap one key, then visually scan through the other keys for the intended letter, which was relatively faster. Hence, the absence of sight made the sequential approach much slower for the blind users than the sighted users, incentivizing blind users to learn the chords faster since it was the only viable alternative for them to improve entry speed. There was a significant effect of level-of-sight. Blind users performed 45% more chords than low vision users. There was also a significant effect of session on error rate, while there was none with sighted participants. Error rate reduced by 28% in the last session, compared to the first. This is most likely due to the difficulty in locating the letters in the beginning. It is possible that over time visually impaired users will reach an error rate comparable to the sighted users.

Entry speed with Qwerty was significantly different for low vision and blind users, 14.73 wpm and 2.52 wpm, respectively. We speculate, this is because low vision users had some light-perception that aided them in typing to some extent. In Figure 18, one can see that low vision participants yielded a wide

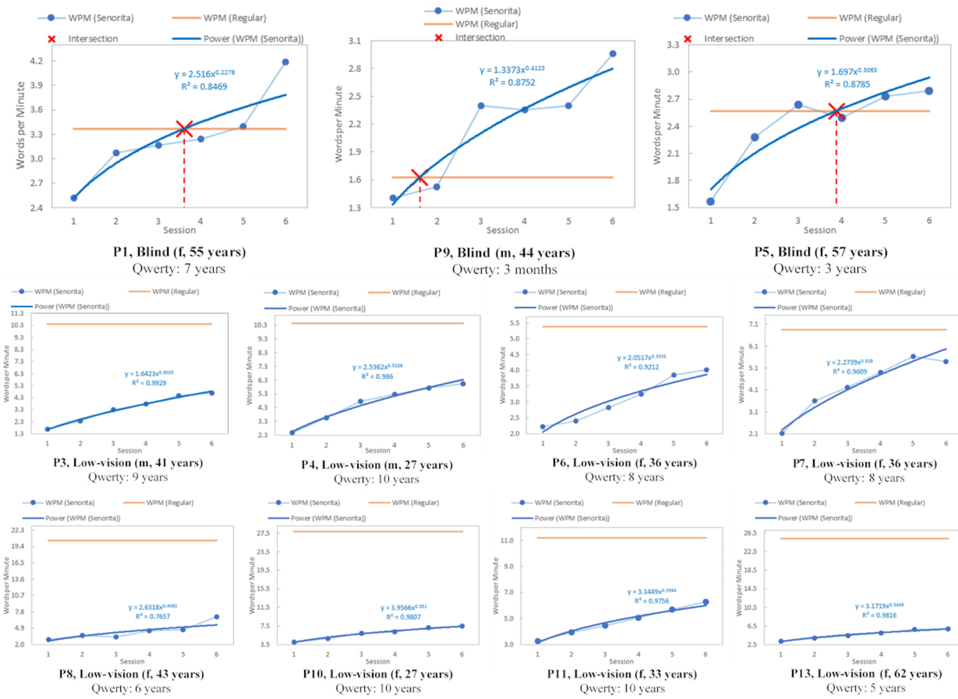


**Figure 18. Average entry speed (wpm) with Senorita for all participants in all sessions compared to Qwerty.**

range of speed: 5.4–27.8 wpm. We tried to find relationships between age, expertise, and speed but failed to find a clear pattern. It is possible that low vision users had different levels of light-perception, which influenced speed. However, we could not explore this as participants were unable to articulate their levels of light-perception. It is troubling that most yielded <10 wpm with Qwerty after years of practice. The three blind users struggled even more: they yielded <3 wpm, two had years of practice and the other was a novice (Figure 18). This implies, it is unlikely that their Qwerty performance will improve over time. The fact that they all surpassed their Qwerty speed with Senorita (32% faster) suggests that it can be an effective method for them to input text on mobile devices.

Senorita performed well compared to most braille-based methods (Table 1). BrailleTouch [46] reported a 9.4–23.2 wpm, but with users who were already familiar with its physical counterpart. It is unknown how it will perform with users who had never used the layout before. TypeInBraille[35] reported a 6.3 wpm, but is difficult to learn. A user who practiced it almost daily for two months reached only 10 wpm [35]. One benefit of Senorita is that it does not rely on the knowledge of braille, thus accessible to a larger audience. Other methods reported less than 2 wpm [23, 25, 37]. Methods that reported 10-15 wpm [7, 26, 25, 52] were evaluated with sighted people, thus not reliable, and/or rely on linguistic models to decode input, making the entry out-of-vocabulary words difficult. Senorita's speed could also be improved by using a predictive system.

Participant responses were very positive in the post-study interview. They all found Senorita easy to learn and use, and most (N = 10) wanted to use it on mobile devices frequently. Blind users praised the faster speed of Senorita. P1: *"Speed is faster especially when I know where the letter."* P5: *"It's better [than Qwerty], over time I will be more accurate and faster."* P5: *"With that keyboard I will text all the time, but with iPhone's keyboard I don't like texting."* Although, low vision users did not perform as well, they were thrilled by the speed and accuracy achieved in such a short period. P11: *"I really liked it, it's innovative. I look forward to use it every day."* P3: *"It's very accurate, because of the fat finger it is difficult with regular keyboard, usually I have to delete a lot."* Some even felt that they were faster with Senorita, while in reality they were not. P11: *"I am faster with chorded keyboard than with Qwerty. Accuracy is the same."* P13: *"I like chorded keyboard better in terms of speed."* Some found the keyboard playful and fun. P6: *"It's fun, like a memory game!"* P13: *"It's playful."* P5, P6: *"It was fun!"* All participants said they would recommend Senorita to their friends and family, particularly to those who are visually impaired. There were also some suggestions for improving Senorita, such as using a predictive system (P11) and providing haptic feedback (P13).

## CONCLUSION

We presented Senorita, a novel two-thumb chorded keyboard aimed at both sighted and visually impaired mobile users. It yielded a 14 wpm on a smartphone by the tenth session and 9.3 wpm on a tablet in a single session. In the final longitudinal study, blind users surpassed their Qwerty performance with Senorita (32% faster), while low vision participants yielded a 5.8 wpm. Besides, visually impaired users found it effective, playful, and wanted to keep using it on mobile devices. In the future, we will address the feedback from the study.

## REFERENCES

[1] Mohammad Ali Alamdar Yazdi, Ashkan Negahban, Lora Cavuoto, and Fadel M Megahed. 2019. Optimization of split keyboard design for touchscreen devices. *International Journal of Human–Computer Interaction* 35, 6 (2019), 468–477.

[2] Maraim Alnfiai and Srinivas Sampalli. 2016. An evaluation of SingleTapBraille keyboard: a text entry method that utilizes braille patterns on touchscreen devices. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. ACM, New York, NY, USA, 161–169. DOI:http://dx.doi.org/10.1145/2982142.2982161

[3] Ahmed Sabbir Arif and Ali Mazalek. 2016. WebTEM: a web application to record text entry metrics. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 415–420. DOI: http://dx.doi.org/10.1145/2992154.2996791

[4] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. DOI: http://dx.doi.org/10.1109/TIC-STH.2009.5444533

[5] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2014. User adaptation to a faulty unistroke-based text entry technique by switching to an alternative gesture set. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 183–192. http://dl.acm.org/citation.cfm?id=2619648.2619679

[6] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 251–260. DOI: http://dx.doi.org/10.1145/2371574.2371612

[7] Nikola Banovic, Koji Yatani, and Khai N. Truong. 2013. Escape-keyboard: a sight-free one-handed text entry method for mobile touch-screen devices. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 5, 3 (2013), 42–61.

[8] Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual gesture keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 137–146. DOI: http://dx.doi.org/10.1145/2380116.2380136

[9] Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: what difference does one key change make? Gesture typing keyboard optimization bounded by one key position change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 49–58. DOI: http://dx.doi.org/10.1145/2858036.2858421

[10] Syed Masum Billah, Yu-Jung Ko, Vikas Ashok, Xiaojun Bi, and IV Ramakrishnan. 2019. Accessible gesture typing for non-visual text entry on smartphones. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 376:1–376:12. DOI: http://dx.doi.org/10.1145/3290605.3300606

[11] Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, and W. Keith Edwards. 2010. No-look notes: accessible eyes-free multi-touch text entry. In *International Conference on Pervasive Computing*. Springer, 409–426.

[12] Rony Aldhea Dwi Cahya, Anik Nur Handayani, and Aji Prasetya Wibawa. 2018. Mobile Braille Touch Application for Visually Impaired People using Double Diamond Approach. In *MATEC Web of Conferences*, Vol. 197. EDP Sciences, 15007.

[13] John M. Carroll and Caroline Carrithers. 1984. Training wheels in a user interface. *Commun. ACM* 27, 8 (Aug. 1984), 800–806. DOI: http://dx.doi.org/10.1145/358198.358218

[14] John M. Carroll and Mary Beth Rosson. 1986. Paradox of the active user. In *Interfacing Thought: Cognitive Aspects of Human-Computer interaction* (1 ed.), John M. Carroll (Ed.). MIT Press, Cambridge, MA, USA, Chapter 5, 80–111.

[15] Li Deng and Xuedong Huang. 2004. Challenges in adopting speech recognition. *Commun. ACM* 47, 1 (2004), 69–75.

[16] Android Developers. 2017. TextToSpeech. (2017). https://developer.android.com/reference/android/speech/tts/TextToSpeech.

[17] Wenrui Diao, Xiangyu Liu, Zhe Zhou, and Kehuan Zhang. 2014. Your voice assistant is mine: how to abuse speakers to steal information and control your phone. In *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM '14)*. ACM, New York, NY, USA, 63–74. DOI: http://dx.doi.org/10.1145/2666620.2666623

[18] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 649–658. DOI: http://dx.doi.org/10.1145/2702123.2702503

[19] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. 2013. ContextType: using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2795–2798. DOI:http://dx.doi.org/10.1145/2470654.2481386

[20] Nathan Green, Jan Kruger, Chirag Faldu, and Robert St. Amant. 2004. A reduced Qwerty keyboard for mobile text entry. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems (CHI EA '04)*. ACM, New York, NY, USA, 1429–1432. DOI: http://dx.doi.org/10.1145/985921.986082

[21] Anna-Maria Gueorguieva, Gulnar Rakhmetulla, and Ahmed Sabbir Arif. 2018. Enabling input on tiny/headless systems using morse code. (Oct. 2018). https://www.asarif.com/pub/Gueorguieva_CCBM2018_Morse_Code.pdf

[22] João Guerreiro, André Rodrigues, Kyle Montague, Tiago Guerreiro, Hugo Nicolau, and Daniel Gonçalves. 2015. TabLETS get physical: non-visual text entry on tablet devices. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI '15*. ACM Press, Seoul, Republic of Korea, 39–42. DOI: http://dx.doi.org/10.1145/2702123.2702373

[23] Tiago Guerreiro, Paulo Lagoá, Hugo Nicolau, Daniel Gonçalves, and Joaquim A Jorge. 2008. From tapping to touching: making touch screens accessible to blind users. *IEEE MultiMedia* 15, 4 (2008), 48–50.

[24] Lilit Hakobyan, Jo Lumsden, Dympna OâĂŹSullivan, and Hannah Bartlett. 2013. Mobile assistive technologies for the visually impaired. *Survey of ophthalmology* 58, 6 (2013), 513–528.

[25] Jianwei Lai, Dongsong Zhang, Sen Wang, Isil Doga Yakut Kilic, and Lina Zhou. 2019. ThumbStroke: a virtual keyboard in support of sight-free and one-handed text entry on touchscreen mobile devices. *ACM Transactions on Management Information Systems (TMIS)* 10, 3 (2019), 11.

[26] Jianwei Lai, Dongsong Zhang, Sen Wang, Isil Yakut Kilic, and Lina Zhou. 2018. A thumb stroke-based virtual keyboard for sight-free text entry on touch-screen mobile phones. In *Proceedings of the 51st Hawaii International Conference on System Sciences*.

[27] Jeffrey S. Leon. 2008. Frequency of Character Pairs in English Language Text, Codes and Cryptography. (2008). http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/char_freq2.pdf.

[28] Stefan Leuthold, Javier A Bargas-Avila, and Klaus Opwis. 2008. Beyond web content accessibility guidelines: Design of enhanced text user interfaces for blind internet users. *International Journal of Human-Computer Studies* 66, 4 (2008), 257–270.

[29] Frank Chun Yat Li, Leah Findlater, and Khai N. Truong. 2013. Effects of hand drift while typing on touchscreens. In *Proceedings of Graphics Interface 2013 (GI '13)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 95–98. http://dl.acm.org/citation.cfm?id=2532129.2532146

[30] Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1Line keyboard: a Qwerty layout in a single line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 461–470. DOI: http://dx.doi.org/10.1145/2047196.2047257

[31] Stan J. Liebowitz and Stephen E. Margolis. 1990. The fable of the keys. *The Journal of Law and Economics* 33, 1 (1990), 1–25.

[32] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. DOI: http://dx.doi.org/10.1145/765891.765971

[33] I. Scott MacKenzie and Kumiko Tanaka-Ishii. 2007. *Text Entry Systems: Mobility, Accessibility, Universality*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[34] I. Scott MacKenzie and Shawn X. Zhang. 1999. The Design and Evaluation of a High-performance Soft Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 25–31. DOI: http://dx.doi.org/10.1145/302979.302983

[35] Sergio Mascetti, Cristian Bernareggi, and Matteo Belotti. 2012. TypeInBraille: quick eyes-free typing on smartphones. In *International Conference on Computers for Handicapped Persons*. Springer, 615–622.

[36] National Federation of the Blind (NFB). 2009. *The Braille literacy crisis in America*. Technical Report. National Federation of the Blind (NFB), Baltimore, MD, USA. https://nfb.org/images/nfb/documents/pdf/braille_literacy_report_web.pdf.

[37] João Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2011. Blind people and mobile touch-based text-entry: acknowledging the need for different flavors. In *The Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '11)*. ACM, New York, NY, USA, 179–186. DOI: http://dx.doi.org/10.1145/2049536.2049569

[38] Antti Oulasvirta, Anna Reichel, Wenbin Li, Yan Zhang, Myroslav Bachynskyi, Keith Vertanen, and Per Ola Kristensson. 2013. Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2765–2774. DOI: http://dx.doi.org/10.1145/2470654.2481383

[39] Kyudong Park, Taedong Goh, and Hyo-Jeong So. 2014. Toward accessible mobile application design: developing mobile application accessibility guidelines for people with visual impairment. In *Proceedings of HCI Korea (HCIK '15)*. Hanbit Media, Inc., South Korea, 31–38. http://dl.acm.org/citation.cfm?id=2729485.2729491

[40] Keith B. Perry and Juan Pablo Hourcade. 2008. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics Interface 2008 (GI '08)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 57–64. http://dl.acm.org/citation.cfm?id=1375714.1375725

[41] Marco Romano, Luca Paolino, Genoveffa Tortora, and Giuliana Vitiello. 2014. The tap and slide keyboard: A new interaction method for mobile device text entry. *International Journal of Human-Computer Interaction* 30, 12 (2014), 935–945.

[42] Anthony Savidis and Constantine Stephanidis. 2004. Unified user interface development: the software engineering of universally accessible interactions. *Univers. Access Inf. Soc.* 3, 3-4 (Oct. 2004), 165–193. DOI:http://dx.doi.org/10.1007/s10209-004-0096-8

[43] George S Snoddy. 1926. Learning and stability: a psychophysiological analysis of a case of motor learning with clinical applications. *Journal of Applied Psychology* 10, 1 (1926), 1.

[44] Mads Soegaard. 2019. Accessibility: Usability for all. Interaction Design Foundation. (2019). https://www.interaction-design.org/literature/article/accessibility-usability-for-all.

[45] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 113–120. DOI: http://dx.doi.org/10.1145/642611.642632

[46] Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. 2012. An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM, 317–326.

[47] Thomas S Tullis and Jacqueline N Stetson. 2004. A comparison of questionnaires for assessing website usability. In *Usability professional association conference*, Vol. 1. Minneapolis, USA.

[48] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 659–668. DOI: http://dx.doi.org/10.1145/2702123.2702135

[49] Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 657–666. DOI: http://dx.doi.org/10.1145/1240624.1240727

[50] World Health Organization (WHO). 2008. Change the definition of blindness. (2008), 5 pages. http://www.who.int/blindness/Change%20the%20Definition%20of%20Blindness.pdf.

[51] Daniel Wigdor and Ravin Balakrishnan. 2004. A comparison of consecutive and concurrent input text entry techniques for mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 81–88. DOI: http://dx.doi.org/10.1145/985692.985703

[52] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 61–70. DOI: http://dx.doi.org/10.1145/964696.964703

[53] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 439, 13 pages. DOI: http://dx.doi.org/10.1145/3173574.3174013