# Input Overload - Minimizing Text Input Devices for Small, Touch-Based Systems

Nathan Brown

Sarah Elder

Deepak Nair

Shalki Shrivastava

# Goal

The goal of this work is to help developers build better forms of mobile interfaces by providing information on the usability of keyboards in which a combination of gestures including “swipe” and “tap” are used to overload the keys, enabling an increase in overall target key size while maintaining or minimizing overall keyboard size.

# Introduction

The proliferation of touch-based mobile phones and other touch-input devices have expanded the available options for text input. According to a 2015 study by Pew Research, 64% of adults in the United states own a smartphone, up from 25% in 2011. Over 88% of smartphone owners use their devices to check email, and over 30% of smartphone owners use their devices to access educational content[1]. These and other common applications may require users to perform a significant amount of text entry. Many of the current text input interfaces for these devices, we will call them “keyboards”, remain similar in operation and structure to the traditional keyboard used for desktop computers. Mobile phone interfaces are small, and full keyboards take a significant amount of valuable space on the screen.

We propose leveraging the greater variety of input formats available with touch-screen devices to improve overall usability.  Work comparing gesture-based text input with other keyboards on mobile devices indicates that text input via tap-based keyboards may be slightly more efficient[2] [3]. However, using multiple forms of input enables a developer to overload each key while still enabling users to use a single movement to select each character. Specifically, we propose using “swipe” motions in various directions, possibly in combination with the existing “tap” functionality. Overloading the keys enables a developer to increase key size within the limited space of a touch-phone screen since fewer keys are required. The increased target size may reduce the frequency of errors caused by users hitting the wrong key[4]. Hence the design may trade-off efficiency for effectiveness in hopes of increasing usability.

Our target audience is users of touch-screen devices such as smartphones. While the applications for which our target audience may be inputting text may change based on demographics such as age[5], we think this input method could be used by a broad range of individuals.  We envision that our keyboard could be used with a number of applications on

---

[1] Smith, A. "U.S. Smartphone Use in 2015" 2015. <http://www.pewinternet.org/files/2015/03/PI_Smartphones_0401151.pdf>

[2] Luo, Lu, and Bonnie E John. "Predicting task execution time on handheld devices using the keystroke-level model." *CHI'05 extended abstracts on Human factors in computing systems* 2 Apr. 2005: 1605-1608.

[3] Költringer, Thomas, and Thomas Grechenig. "Comparing the immediate usability of graffiti 2 and virtual keyboard." *CHI'04 Extended Abstracts on Human Factors in Computing Systems* 24 Apr. 2004: 1175-1178.

[4] Page, Tom. "Usability of text input interfaces in smartphones." *Journal of Design Research* 11.1 (2013): 39-56.

[5] Smith, A. "U.S. Smartphone Use in 2015" 2015. <http://www.pewinternet.org/files/2015/03/PI_Smartphones_0401151.pdf>

mobile devices that require alphanumeric input, such as email clients, social-networking sites, and note-taking applications.

One sample scenario in which our device could be used would be for a grocery shopper. Before starting their grocery trip, the user decides to create a shopping list and store it on their smartphone so that it can be conveniently retrieved while they are shopping. The user has a large family, and the list is likely to be long. Hence the user may not be able to see the entire list on a single screen. This may pose a problem since the user would like to avoid duplicates and does not want to always be scrolling up and down on the screen. To maximize the amount that a user can see and minimize the amount of additional scrolling on the page, the size of the keyboard should be minimized. The developer of the application must decide how small they can make the keys of the keyboard for the keyboard to still be effective.

# System Development

The focus of this paper is on a novel method for inputting text for mobile devices. The system was tested in an application designed to measure usability, but could be used in more realistic contexts such as note-taking applications and web forms. The system was inspired in part by the “stick” keyboard which overloaded each key with three characters, each of which could be selected via repeated presses of the key[6]. The original design used in this study had three characters per key, which could be selected by swipe-up, press, and swipe-down movements. Our subsequent investigation indicated that little work had been done in researching the use of swipe gestures across a touch-screen as keyboard input. Hence our variations focus on the usability concerns of this text-input method, rather than optimizing the layout of the characters on the keys, the number and location of the keys, etc. While we do not keep these entirely constant, we have taken the following steps to standardize our keyboards

- Each keyboard has an alphabetical layout, which we predict will minimize the impact of users’ bias towards layouts similar to QWERTY keyboards[7].
- Each keyboard occupies the same percentage of screen space.

An initial prototype was developed within a week using the UNITY framework[8]. One of the advantages to using this framework was the flexibility it provided in specifying and quickly revising the prototypes. The original design was similar to the design described below as the “tri-key” design, which uses a combination of swipe-up, tap, and swipe-down to input characters.

Discussion of the initial prototype gave rise to the “penta-key” and “bi-key” designs. In order to aid users in their initial learning of the keyboards, visual affordances were added for swipe gestures. Lastly, the baseline variation keyboard was added to allow comparison of our novel designs against the currently prevalent design applied in a smaller space.

[6] Green, Nathan et al. "A reduced QWERTY keyboard for mobile text entry." *CHI'04 extended abstracts on Human factors in computing systems* 24 Apr. 2004: 1429-1432.
[7] Norman, Donald A, and Diane Fisher. "Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 24.5 (1982): 509-519.
[8] "Unity - Game Engine." 2008. 24 Nov. 2015 <https://unity3d.com/>

# System Design

To test the usability of combining input methods we developed three prototype keyboards and compared them against a baseline keyboard in which all characters were input via tapping the keys. The prototype keyboards included two visual affordances to aid users: arrows to indicate when swipe gestures could be used and the highlighting of selected characters. It should be noted that the exact position on an individual key where a user presses is not relevant to our keyboards, only the relative motion of the user’s press is used in determining what letter a key produces.

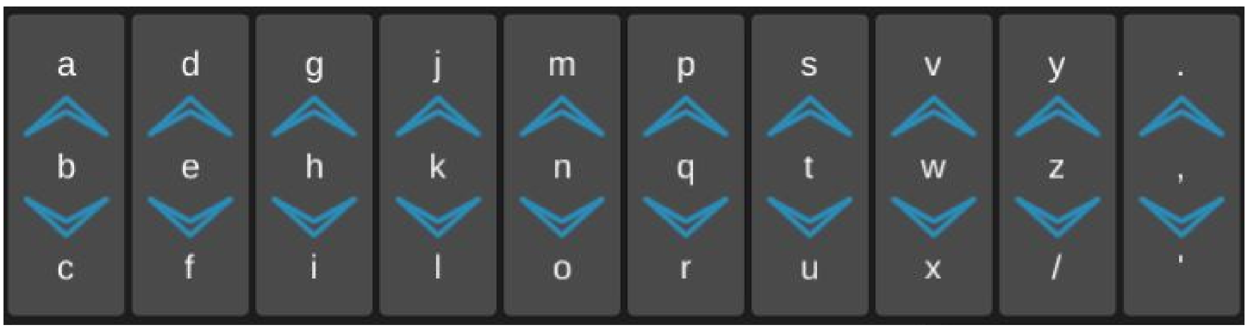


**Figure 1: Tri-key Keyboard**

The tri-key keyboard design is shown in figure 1. Each tri-key contains three letters, each one mapped to either upward swiping, downward swiping, or tapping. An example input from this keyboard would be inputting the word “late” by swiping down on the fourth key from the left, swiping up on the first key on the left, tapping the fourth key from the right, and tapping the second key from the left.

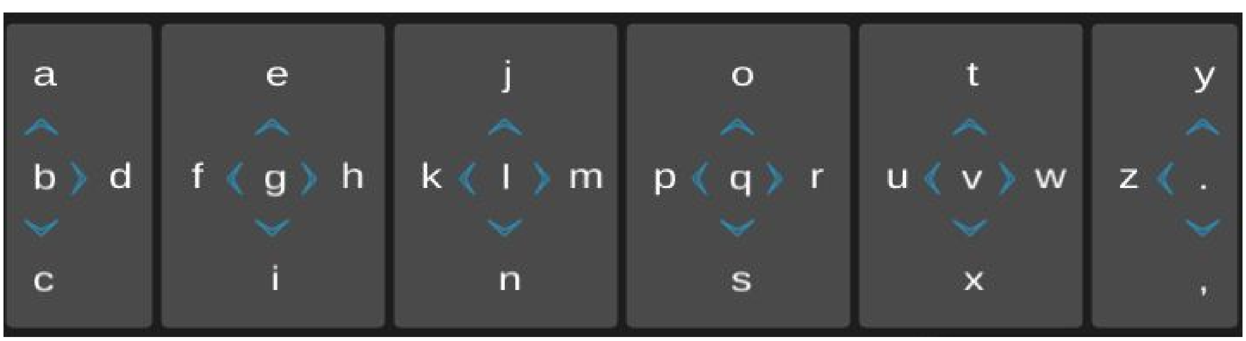


**Figure 2: Penta-key Keyboard**

The penta-key design, shown in figure 2, overloads each key with 5 characters. One character was specified by swiping up across the key, the second by swiping right to left across the key, the third by tapping the key, the fourth by swiping left to right across the key, and the fifth by swiping down across the key. Since the numbers of letters in the alphabet is not a multiple of 5, partial keys were used as the left-most and right-most keys in this design. A screenshot of the prototype is shown below. An example input for this keyboard would be to input “school” by swiping down across the third key from the right, swiping down across the first key on the left, swiping left to right across the second key from the left, swiping up across the third key from the right, swiping up across the third key from the right a second time, and finally tapping the third key from the left.

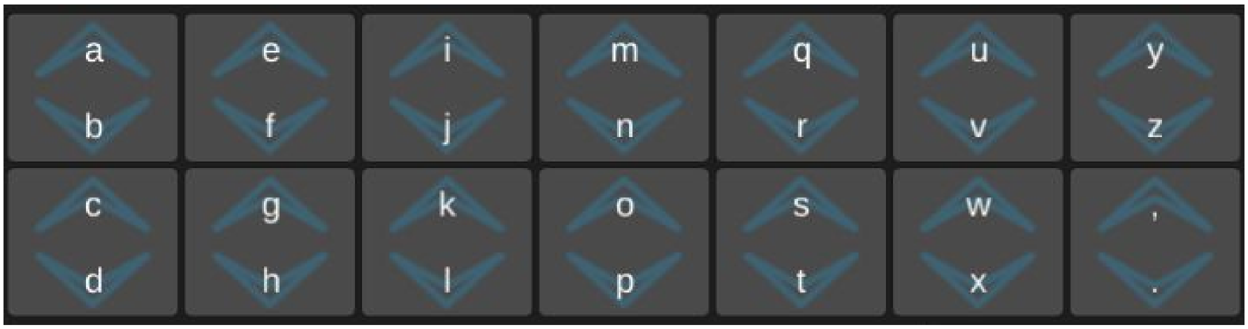


**Figure 3: Bi-key Keyboard**

The bi-key design, shown in figure 3, used exclusively swipe gestures for input. Each key mapped to two characters, which could be output by swiping either up or down on the key. An example input of this keyboard would be to input “their” by swiping down on the third key from the right on the bottom row, swiping down on the second key from the left on the bottom row, swiping up on the second key from the left on the top row, swiping up on the third key from the third key from the left on the top row, and swiping down on the third key from the right on the top row.


**Figure 4: Baseline Keyboard**

Finally, our baseline keyboard, shown in figure 4, was designed to resemble existing keyboards found on touch devices. Each character is mapped to pressing a single key. An example input from this keyboard would be to input “must” by tapping the keys labeled with each letter, in order.

# Evaluation

## Metrics

We evaluated the keyboard designs across the three attributes described by ISO 9241: Effectiveness, Efficiency, and Satisfaction[9].

### Effectiveness

We assumed that, given infinite time to complete the task and the inclusion of a "backspace" key, a literate user would be able to use a keyboard based on our prototypes to input text as needed. For some applications additional keys may need to be added or the keyboard itself could be overloaded as is done in many current applications, i.e. when a modifier key is selected the entire keyboard changes from alpha-numeric input to punctuation or other necessary characters. Hence our main metric for effectiveness is error rate rather than completion. We anticipated that the error rate would be lower for keyboards with larger keys, as discussed by Tom Page in his analysis of the "Thick Buttons" keyboard[10]. Key size corresponded with the number of movements a key was overloaded with, i.e. the penta-key design had significantly larger keys than the baseline keyboard. For consistency, we did not include a "backspace" key and instructed users to simply ignore the errors they made as noted in the script seen in Appendix B. This allowed us to compare the characters entered with the intended characters. We measured error-rate at the word level.Our hypothesis was that the error rate would have an inverse correlation with key size. The penta-key design would perform best in this regard with the lowest error rate, followed by the tri-key design, followed by the bi-key design. The baseline design would have the highest error rate.

### Efficiency

Efficiency was measured as the average amount of time, in seconds, it took for users to input a word. To form our hypothesis on efficiency, we modeled each keyboard in CogTool, an application developed by researchers at Carnegie Mellon University that can be used to predict how users will interact with a system prior to implementation. One of the greatest advantages of using CogTool came from our challenge to find a model that could reasonably applied to the "swipe" movements we were implementing. Models based on Fitt's law are difficult to apply when the endpoint of the movement is not defined[11], which is the case for "swipe" motions. While there have been attempts to extend the Keystroke-Level-Model, another well-known

---

[9] Jokela, Timo et al. "The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11." *Proceedings of the Latin American conference on Human-computer interaction* 17 Aug. 2003: 53-60.
[10] Page, Tom. "Usability of text input interfaces in smartphones." *Journal of Design Research* 11.1 (2013): 39-56.
[11] Carroll, John M. *HCI models, theories, and frameworks: Toward a multidisciplinary science*. John M Carroll. Morgan Kaufmann, 2003.

model in Human-Computer Interaction, to mobile devices[12], these also do not yet include the "swipe" motion. Some of the closest research to this area involves text-entry methods that do not involve a traditional "keyboard" where each letter is traced on the touch-screen[13][14]. Fortunately, CogTool already includes predictive analysis of one of these text-entry methods, Graffiti[15], which was originally developed for Palm OS[16]. Although it is not an exact match, the movements for inputting "!" and "1" in Graffiti were judged to be similar enough to our "up" and "down" swipes for a valid comparison. The predictive analysis for the "-" character from Graffiti was used to model both left to right and right to left swipes since we could not find an exact match for both movements. Using the same key for both horizontal gestures would have less impact since the difficulty of executing horizontal swipes would be based on which hand was used.

| **Word** | **Baseline Prediction (in seconds)** | **Bi-Key Prediction (in seconds)** | **Tri-Key Prediction (in seconds)** | **Penta-Key Prediction (in seconds)** |
|---|---|---|---|---|
| That | 8.5 | 11 | 9.1 | 11.1 |
| Different | 14.6 | 15.7 | 13.9 | 19.9 |
| Course | 11.3 | 15.6 | 14.8 | 15.1 |
| From | 8.3 | 10.9 | 10.6 | 10.5 |
| Industry | 14.4 | 19.8 | 18.3 | 19.8 |
| Event | 11.3 | 13.3 | 10.7 | 12.5 |
| Book | 6.8 | 9.4 | 8.0 | 8.7 |
| Billion | 11.1 | 15.9 | 14.5 | 13.9 |
| Love | 8.3 | 11.1 | 10.4 | 9.5 |
| Would | 9.8 | 13.3 | 12.6 | 12.4 |
| **AVERAGE** | 10.44 | 13.6 | 12.29 | 13.34 |

**Table 1: Predicted results for efficiency**

12 Holleis, Paul et al. "Keystroke-level model for advanced mobile phone interaction." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* 29 Apr. 2007: 1505-1514.
13 Wobbrock, Jacob O, Brad Myers, and Scott E Hudson. "Exploring edge-based input techniques for handheld text entry." *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on* 19 May. 2003: 280-282.
14 Költringer, Thomas, and Thomas Grechenig. "Comparing the immediate usability of graffiti 2 and virtual keyboard." *CHI'04 Extended Abstracts on Human Factors in Computing Systems* 24 Apr. 2004: 1175-1178.
15 John, BE. "CogTool user guide version 1.2." (2012).
16 Költringer, Thomas, and Thomas Grechenig. "Comparing the immediate usability of graffiti 2 and virtual keyboard." *CHI'04 Extended Abstracts on Human Factors in Computing Systems* 24 Apr. 2004: 1175-1178.

To predict how these models would apply to the tasks we are testing, we chose ten random words from the 500 potential candidates user would be presented with. We then used CogTool to model the tasks using each keyboard, and ran the analysis tool. The resulting analysis is shown in table 1 above. The values listed are the approximate amount of time it should take to complete the task, in seconds. Some of the assumptions made when creating and applying the models were:

- Users would only need to look at the prompt one time, and would remember the word that they were attempting to input for the remainder of the time spent inputting that word
- Users would need to look at each letter prior to entering it EXCEPT when the same letter was entered twice in succession, e.g. the second "l" in billion. This assumption is based on the fact that users would not be familiar with they keyboards and format beforehand. Although some recall would be anticipated, its full impact was difficult to determine prior to testing.

These assumptions were based on our experience with the initial prototypes.

As can be seen in table 1 above, on-average the baseline model was predicted to be most efficient; the tri-key design for which approximately one-third of the character inputs would be taps, ignoring differences in character frequency in the English languages, was predicted to be second-most efficient; the bi-key design, for which all character inputs would be swipes, was predicted to be least efficient. Interestingly, there was some discrepancy as the length of the word increased. For the largest two words the bi-design was predicted to be equally or more efficient than the penta-key design as shown in table 1 above. However, given the small sample size this could have been influenced by a variety of factors. Overall our hypothesis was that the average number of seconds necessary to complete a word would decrease as as the number of swipes necessary to input the text decreased in relation to the number of taps necessary to input the text.

### Satisfaction

To measure satisfaction as part of our testing, we used a set of surveys based on the System Usability Scale (SUS) as described by John Brooke[17]. We performed a short survey after a user completed the five-minute test for each keyboard to evaluate the usability of that keyboard, and performed a final survey once the user had tested all four keyboards to determine how users thought the keyboards compared against each other.

Since satisfaction is a very subjective measurement, our hypothesis on this measure was less confident. Our team felt that the tri-key and penta-key models were more user-friendly, which is why we developed the prototypes. Satisfaction is further complicated as we compare across keyboards. As mentioned previously, to minimize the impact of QWERTY bias on our usability testing and reduce the number of variables in our experiment, we used alphabetical layouts for all keyboards. Among other concerns, our users were already familiar with QWERTY-based keyboards similar to our baseline keyboard. Previous research in this area is unclear on what, if any, impact familiarity with QWERTY-based keyboards has on

---

[17] Brooke, John. "SUS-A quick and dirty usability scale." *Usability evaluation in industry* 189.194 (1996): 4-7.

non-QWERTY keyboards, particularly those that are similar in structure to QWERTY-based keyboards[18]. However, our hypothesis is that when comparing an alphabetical-based penta-key or tri-key keyboard to an alphabetical-based baseline keyboard, users prefer the penta-key or tri-key keyboards. Within the multi-input based keyboards we anticipate there being no preference.

## Experiment Plan

### Experiment Goal

To test the usability of the three alternate keyboard designs and compare them with the baseline keyboard, as well as each other.

### Experiment Participants

In this study, 4 smartphone users, all male, in the age bracket 23 to 27 participated in the user study. All of them are engineering students at NC State University. All of them are familiar with the swipe input method based qwerty keyboard layout and can be categorized as typical users. Although their first language isn't English, it is the language of their entire education and work, and hence they are quite fluent in it. So, we don't believe that this will affect our test results in any way even though we are testing our keyboard specifically designed for the English language. The participants were well informed of the time commitment for the test and the compensation.

### Experiment Design

The experiment was a *within subjects* design with 2 factors:

1. Keyboard Layouts: Bi-key, Tri-key, Penta-key

2. Instrumentation familiarity: The tests were carried out on two different mobile handsets albeit of similar size. But there still might be some unexpected effects on the test results due to minor differences in screen size and/or the sensitivity of the touch screens. We tried to overcome the effect of keyboard layouts by randomizing the order in which they are presented to the participant. The effects of different handsets, on the other hand, were difficult to mitigate due to limited number of handsets available for testing.

The experiment had 2 test administrators and 4 participants. The participants were asked to fill in a small survey after completing the test for each keyboard layout to evaluate their experience. At the end of the entire experiment, they were asked to answer some subjective questions to compare their relative experiences with different layouts.

[18] Norman, Donald A, and Diane Fisher. "Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter." *Human Factors: The Journal of the Human Factors and Ergonomics Society* 24.5 (1982): 509-519.

*Data Collection:* The aim of the experiment being to measure the effectiveness and efficiency of the keyboard layout and the satisfaction the users, we collected numerous measurements on user input. For each keyboard layout, the total number of words entered by the participant in 5 minutes were counted. For each word entered, all successive taps along with their timestamps were logged. We later used this information to evaluate the time taken to enter each word, which gives an idea of the efficiency of the keyboard. We used the log of taps to calculate the accuracy with which each word is entered by the user. This information helped us measure the effectiveness of the keyboard for text input.

### Experiment Procedure

To test our hypotheses, we developed the prototype keyboards discussed previously, and implemented them in an application designed to assist in evaluating the various aspects of usability. The application and test procedure are described below.

To begin the testing process, the test administrator read the prompt provided by our test script, which can be found in Appendix B. This script was created prior to testing to increase the standardization of the testing process across different test administrators and participants. As part of this prompt, users were told that their goal was to enter the words being displayed at the top of the screen. Users were also instructed not to correct errors. The design of the application under test did not include backspace or delete keys to further avoid any confusion on this instruction. This was to facilitate measuring both consistency and error rate although the lack of an "emergency exit" would be a usability flaw as noted by Nielsen's ten heuristics[19]. Hence the application under test was optimized to gather usability data more than actual usability. Once the test administrator read the script, the user was ready to start working with the application. The executable file for the application is listed in Appendix A. A video clip of a user demonstrating the application is listed in Appendix D.

When a user first started the application, they were presented with a home screen where they could choose to run the full version of the test, which ran for five minutes per keyboard, or a trial version of the test, which ran for thirty seconds per keyboard.  A screenshot of this initial screen is shown in figure 5. The trial version was intended primarily for use in debugging the application. The user selected the appropriate version and pressed the "New Test" button, which took them to the first keyboard.

---

[19] "10 Usability Heuristics for User Interface Design - Nielsen ..." 2012. 25 Nov. 2015 <http://www.nngroup.com/articles/ten-usability-heuristics/>

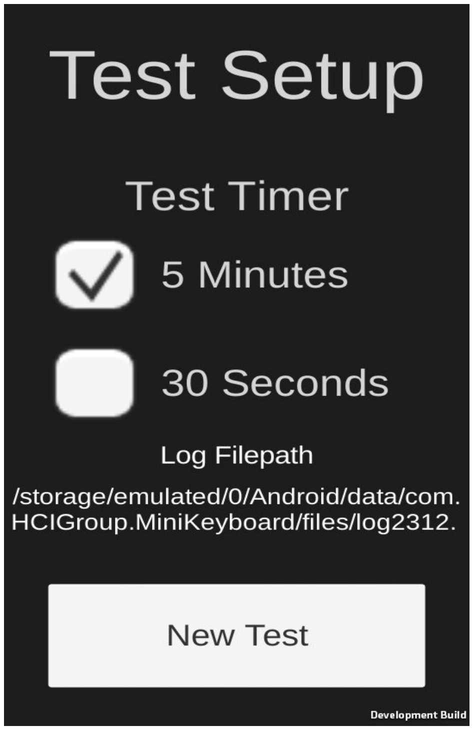


**Figure 5: Test Setup Screen**

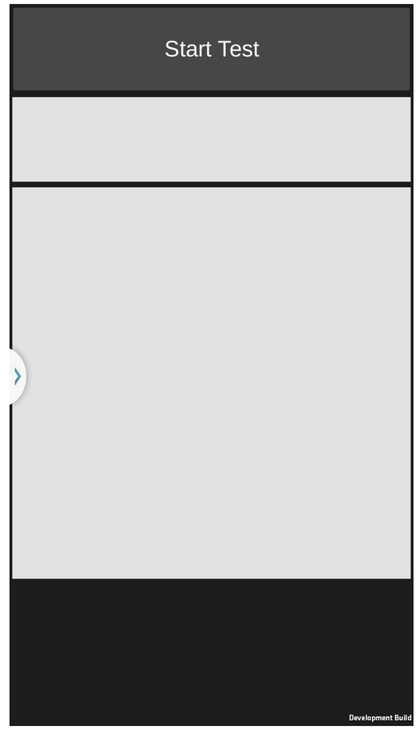


**Figure 6: Pre-Test Screen**

Once the user had selected the test, they were taken to the start screen, shown in figure 6, for the first keyboard. Since the user could remain on the start screen for a keyboard indefinitely between keyboard design tests while feedback was taken for the survey, the start screen for the keyboard did not display the keyboard itself. When the user was ready to test the keyboard, they pressed the “Start Test” button.

Once the user pressed “Start Test” they would be taken to one of the four keyboards. Our application presented each of the keyboard designs in a random order. We did not know whether using any one of the keyboards would make subsequent keyboards easier to use. Randomly varying the order in which the keyboards appeared for each user mitigated the impact of potential bias in this area. For each keyboard test, the keyboard was displayed at the bottom of the screen, the task at the top of the screen, and the part of the task that had already been completed (feedback) in the middle of the screen. For example, if the user’s task was to input the word “their” and the keyboard was the penta-key design; “their” would appear at the top of the screen; the penta-key keyboard would appear at the bottom of the screen; and any characters that the user entered, correctly or erroneously, were displayed in the center of the screen. The space and carriage return buttons used to indicate that the user was ready for the next task appeared at the bottom of the screen. This layout is shown figure 7.

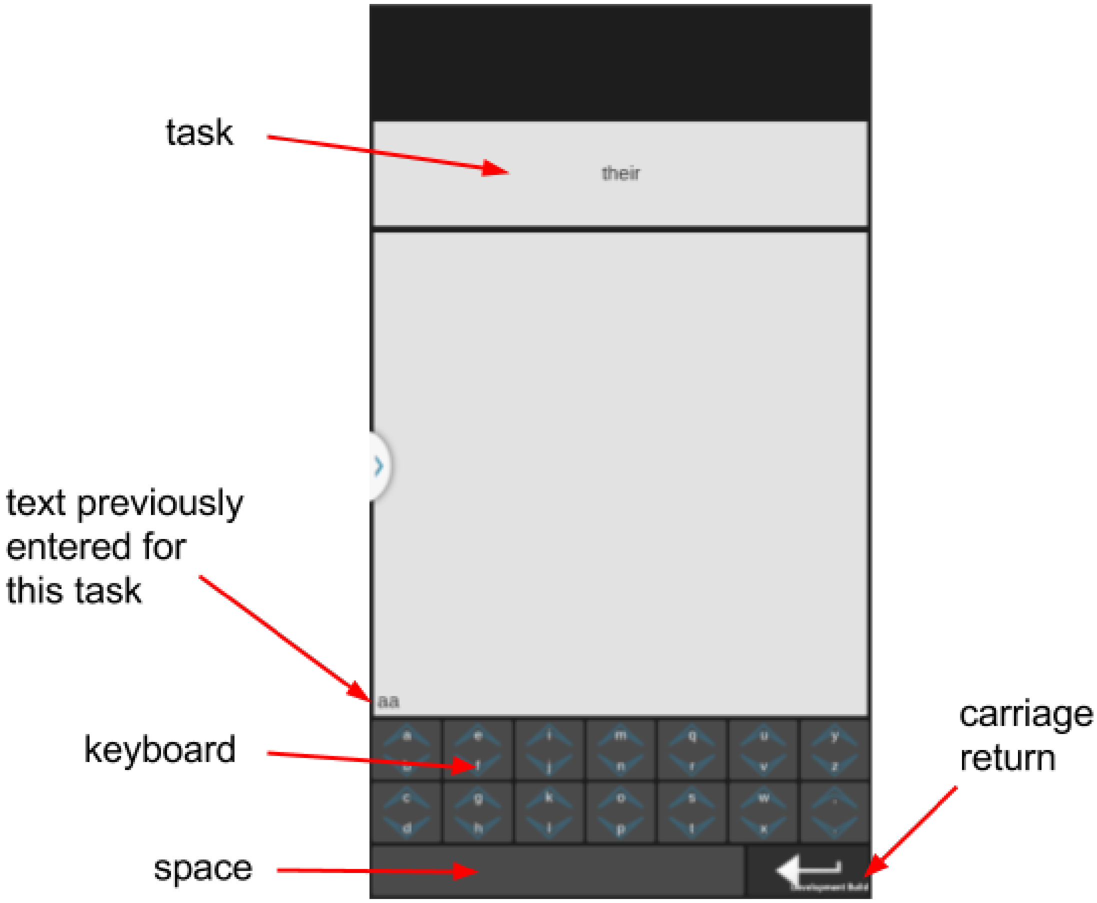


**Figure 7 - Keyboard Screen**

The first task for each keyboard was the training task, which was to input the entire alphabet. This task was not included in our overall calculations of accuracy and efficiency. Inputting the alphabet enabled the user to familiarize themselves with the keyboard, similar to practices used in other studies[20] [21]. Figures 8 and 9 show the penta-key and bi-key designs respectively during this initial task.

Subsequent tasks for the keyboard consisted of entering single words that were specified. Each word was chosen at random from a list of 500 common English words. These tasks were used to measure efficiency and effectiveness as described in the previous section. Screenshots of different words displayed on the tri-key and baseline designs are shown in figures 10 and 11.

---

[20] Költringer, Thomas, and Thomas Grechenig. "Comparing the immediate usability of graffiti 2 and virtual keyboard." *CHI'04 Extended Abstracts on Human Factors in Computing Systems* 24 Apr. 2004: 1175-1178.

[21] Page, Tom. "Usability of text input interfaces in smartphones." *Journal of Design Research* 11.1 (2013): 39-56.

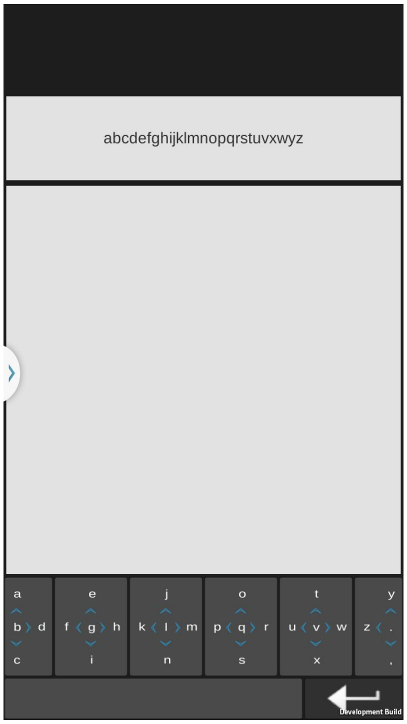


**Figure 8: Penta-key Alphabet**

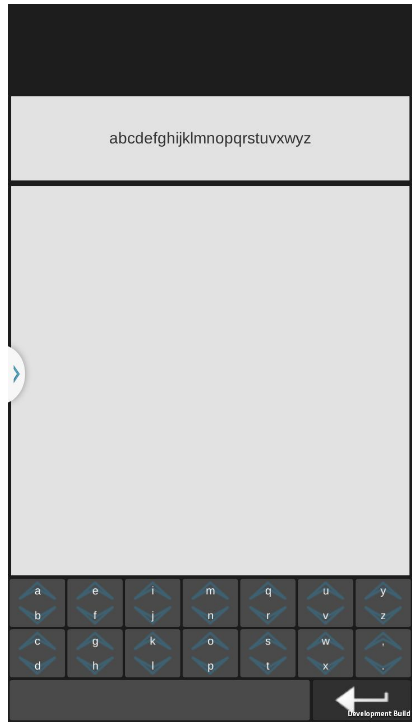


**Figure 9: Bi-key Alphabet**

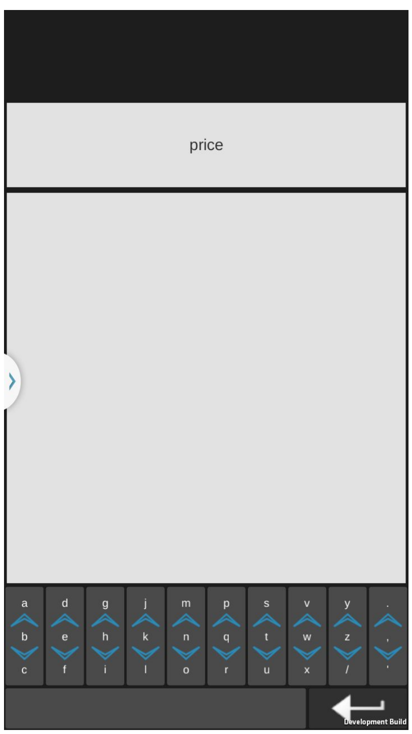


**Figure 10: Tri-key Task**

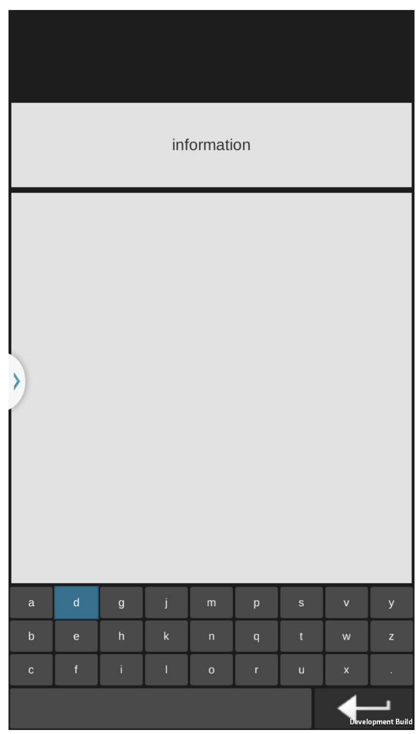


**Figure 11: Baseline Task**

At the end of the selected timeframe for each keyboard, generally five minutes, the user was returned to a keyboard start screen, at which point the user participated in a survey and took a short break. For each keyboard a SUS-based survey was completed. After all four keyboards had been tested, the user completed an additional survey comparing the keyboards against each other and against keyboards commonly used for mobile devices. The survey questions used can be found in Appendix C. The results of this survey are described later in this paper

# Results

## Data

| Keyboard | Error Rate | | Seconds per Word | |
|---|---|---|---|---|
| | Average | Standard Deviation | Average | Standard Deviation |
| Baseline | 0.369047619 | 0.1435169946 | 11.47956435 | 2.545125743 |
| Bi-Key | 0.5043206793 | 0.2762820105 | 13.99100449 | 3.612407642 |
| Tri-Key | 0.5114896545 | 0.337564386 | 11.97538214 | 2.453831384 |
| Penta-Key | 0.5452380952 | 0.4804560932 | 10.2510611 | 0.8314257048 |

**Table 2: Keyboard Measures**

The raw data used for this analysis can be found in Appendix E. The data appears to agree with our predictions on efficiency, disagree with our predictions on effectiveness, and provide mixed results as to the satisfaction of end-users. However, as will be discussed in the Limitations and Conclusion sections, this data may have significant bias and additional testing is needed to produce an optimal system.

All of our keyboards had higher average error rates than the baseline keyboard, though for some individual users error rates decreased on non-baseline keyboards. As the standard deviations for error rate show, some keyboards had very large differences in user error rates. For example, the Penta-key keyboard had error rates ranging from 12% to 96%.

For average seconds per word, not all keyboards conform to our prior Cogtool analysis. The Penta-key keyboard actually had the lowest average seconds per word, while Cogtool predicted it would have the largest. Another point to note is that the Penta-Key keyboard has a relatively low standard deviation of seconds per word compared to the other keyboards.

The SUS score data for each keyboard is shown in table 3. Both the Bi-Key and Tri-Key keyboards had significantly lower averages than the baseline keyboard, while the Penta-Key's average SUS was comparable (only roughly 6% less). The standard deviations are relatively large, indicating a large range in user opinions. The descriptive survey questions yielded a similarly wide range of opinions about the keyboards.

| Keyboard | SUS Average | SUS Standard Deviation |
|---|---|---|
| Baseline | 73.75 | 20.25874297 |
| Bi-Key | 55.625 | 19.93478952 |
| Tri-Key | 52.5 | 23.54074482 |
| Penta-Key | 69.375 | 28.01599246 |

**Table 3: SUS Scores**

## Limitations

Our study focused on optimizing preliminary prototypes, and had many limitations. Given the large standard deviation in our results, these limitations were likely to have impacted the results of our work. Limitations and possible impacts include variance in the test environment, limitations imposed by our choice of metrics for error rate, size and demographics of the participant pool, and time constraints.

First, results of this study may have been impacted by variances in the test environment including physical location, time of day, and hardware used. We have reduced the impact of these differences in our analysis by comparing the keyboards within users rather than across users since these factors should be more consistent.

The error metric used in this study itself is a source of limitations. We only measured error rate at the word level, and did not measure the severity of the error or whether the participant was aware of the error. We also did not determine the origin of the error. Although our hypothesis about error rates was disproved, a more detailed metric for error rate might provide information.

A significant limitation of this study was the size and composition of the participant pool. While the keyboard order was randomized, the effectiveness of the randomization was impacted by sample size. For example, two individuals, 50% of our participants, were presented with the penta-key design first. While these two individuals also had the highest error rate on that keyboard, we could not conclusively determine whether the order in which keyboards were used impacted performance. Our participant pool had very little demographic diversity, therefore our participants' performance and preferences may not be representative of the general population of smartphone users. Overall, a larger, more diverse group of participants should be used.

Another limitation of our study was we did not have time to run multiple tests, and could not evaluate how easy the designs were to recall or whether the error rate declined proportionally for all keyboards. Recall can be an important factor in usability, although it is not specifically mentioned in the ISO standard[22]. Comparing results over multiple trials may have further reduced the impact of bias based on previous experience with other text input devices.

---

[22] Harrison, Rachel, Derek Flood, and David Duce. "Usability of mobile applications: literature review and rationale for a new usability model." *Journal of Interaction Science* 1.1 (2013): 1-16.

## Conclusions

The main conclusion we draw from our data is that a new experiment needs to be designed and a significantly larger number of participants tested. The only keyboard to perform worse than the baseline keyboard in terms of error rate across all tests was the Bi-Key keyboard. The performance of the penta-key keyboard for two of the participants suggests it might be the most viable of the keyboards, but with such a small data set it’s not possible to discern whether those tests were predictive. Although the results of our surveys indicate an interest in the more novel input devices, overall user satisfaction was mixed.

# Appendix A - Executable File

The executable file for the application attached to this report is the following file: MiniKeyboard.apk.

# Appendix B - Experiment Script

Hello! As a part of my studies in Human Computer Interaction, I am conducting research about alternate layouts for virtual keyboard, specifically for mobile, touch-screen devices. I have a small experiment to conduct which will take about 30 min to complete. No personally identifying information is being collected. Do you have any questions about the research study? If you have any questions later, you can reach me at *email-address.*

Format of the experiment

We will be testing the usability of 4 different layouts of the keyboard. For each keyboard, there will be an initial training phase to familiarize you with the layout before the testing actually begins. The training phase will consist of typing the letters A-Z. The testing phase will be 5 min long. Your goal is to type the words that appear on the screen as *quickly and accurately* as possible. There will be a 30 second break in between the testing of each keyboard in which we will briefly discuss the ease of use of the keyboard. At the end of the testing of all the keyboards, there will be a small survey to assess the relative usability of the keyboards.

When you start the application, you will see two options, 30 second test and 5 minute test. Choose 5 minutes and press the New Test button at the bottom of the screen when you are ready to begin. You will be redirected to a new screen where you will see a Start Test button at the top of the screen. For every new keyboard you can use this “Start Test” button to begin the experiment. Below each keyboard are spacebar and carriage return buttons used for moving from one word to next. For experimental purposes, we have removed the delete button to be able to measure the error rate in a standard manner. There is no need to correct the mistakes. If you feel that you have misspelt a word, please ignore that and carry on with the rest of the word (or press spacebar or carriage return at the end of the word to move to the next word). At the end of 5 minutes, the test will stop and you will be redirected to the Start Test screen for the next keyboard.

Each keyboard accepts input from at least one of the following methods:

1. Tapping the key - the letter listed the center of each key is input by simply tapping the key. On keyboards where there is no “center” letter, tapping the key provides no input.
2. Swiping in one or more directions across the key - The possible directions in which a given key can be “swiped” are shown on a key. For keys that use swipe as an input, the

swipe motion will input the character listed in the location on the key that corresponds with the direction of the swipe

Are you ready to begin? Thanks for your participation.

Legend

1. Bi-Key - Just vertical swipe
2. Tri-Key - Tap and vertical swipes
3. Penta- - Tap and all directions swipe
4. S - Standard just tap

# Appendix C - Experiment Survey

## Keyboard Bi-Key (Just Vertical Swipes)

*Write a number from 1 to 5, with 1 = strongly disagree; 5 = strongly agree*

- Would you like to use this keyboard over the keyboard you typically use with a smartphone?
- Did you find the keyboard very complex?
- Do you think this keyboard is intuitive?
- If not, do you think a well written description can provide enough insight into effectively using this keyboard to anyone?
- Did you find any inconsistencies in the keyboard compared to general terminologies and practices?
- Do you think the keyboard can be learnt easily?
- Did you find the keyboard very cumbersome?
- Do you think you could use it confidently?
- Do you think you would require a lot of practice so that you can better use this keyboard?

## Keyboard Tri-Key (Tap and Vertical Swipes)

*Write a number from 1 to 5, with 1 = strongly disagree; 5 = strongly agree*

- Would you like to use this keyboard over the keyboard you typically use with a smartphone?
- Did you find the keyboard very complex?
- Do you think this keyboard is intuitive?
- If not, do you think a well written description can provide enough insight into effectively using this keyboard to anyone?
- Did you find any inconsistencies in the keyboard compared to general terminologies and practices?

- Do you think the keyboard can be learnt easily?
- Did you find the keyboard very cumbersome?
- Do you think you could use it confidently?
- Do you think you would require a lot of practice so that you can better use this keyboard?

## Keyboard Penta-Key (Tap and All Directions Swipes)

*Write a number from 1 to 5, with 1 = strongly disagree; 5 = strongly agree*

- Would you like to use this keyboard over the keyboard you typically use with a smartphone?
- Did you find the keyboard very complex?
- Do you think this keyboard is intuitive?
- If not, do you think a well written description can provide enough insight into effectively using this keyboard to anyone?
- Did you find any inconsistencies in the keyboard compared to general terminologies and practices?
- Do you think the keyboard can be learnt easily?
- Did you find the keyboard very cumbersome?
- Do you think you could use it confidently?
- Do you think you would require a lot of practice so that you can better use this keyboard?

## Keyboard S (Just Taps)

*Write a number from 1 to 5, with 1 = strongly disagree; 5 = strongly agree*

- Would you like to use this keyboard over the keyboard you typically use with a smartphone?
- Did you find the keyboard very complex?
- Do you think this keyboard is intuitive?

- If not, do you think a well written description can provide enough insight into effectively using this keyboard to anyone?
- Did you find any inconsistencies in the keyboard compared to general terminologies and practices?
- Do you think the keyboard can be learnt easily?
- Did you find the keyboard very cumbersome?
- Do you think you could use it confidently?
- Do you think you would require a lot of practice so that you can better use this keyboard?

## Final Survey

*Please write as detailed answers as possible.*

- Which keyboard are you most likely to use if you could not use the keyboard you typically use on your smartphone?
- Why did you find the keyboard comfortable?
- What are some of the features which you would like to be included into your preferred keyboard?
- What features do you think any of the keyboards can do without?

# Appendix D - Video

A video showing a demonstration of the tool can be found in the attached file UserDemo.mp4

# Appendix E - Raw Data

Survey data can be found in the following file attached to this report:
SurveyAnalysisSpreadsheet.xlsx

Data such as typing speed that was recorded by the application can be found in the following files attached to this report: ResultsAnalysisSpreadsheet.xlsx