Touch Screen Typing on Mobile Devices

A Dissertation Proposal

Melissa A. Gallagher

September 16, 2013

Rice University

Abstract

With the popularity of mobile devices steadily increasing, validated models of common interactions with these devices would aid designers in usability testing. Typing interactions with mobile devices have been studied with the focus of improving algorithms to increase the fidelity of interaction as well as to make predictions about the interaction. This works hopes to expand the cognitive modeling architecture ACT-R by implementing some of these theories. Measuring and comparing both the speed and the accuracy of interaction is important in typing tasks. The results of an initial study of typing on a mobile device are presented. These results will aid the modeling efforts. A modified follow up study is proposed, to provide data to compare the model's predictions to.

Touch Screen Typing on Mobile Devices

A Dissertation Proposal

As mobile devices become more ubiquitous, creating new applications and adapting existing desktop programs to be successfully used on them continues to be a primary focus for designers. The most recent reports from the Pew Research Center (Zickuhr 2013, Smith 2013) found that over 55% of the Americans surveyed own a smartphone and 34% own a tablet. The percentage of the population that owns a mobile device has been steadily growing for the past five years. One method of doing usability evaluation without collecting extensive empirical data is to create a computational cognitive model of the task. Cognitive models can predict response time and error rates without having to run subjects through an experiment. This allows designers to explore alternative interface designs across tasks, and predict potential usability problems prior to implementation and deployment. Typing is a common subtask in many Human-Computer Interaction (HCI) tasks, and having the ability to model typing with mobile devices has long-term implications for usability testing. While there are well-developed models of desktop typing, these models likely need revision to be useful in modeling typing on mobile devices. Due to mobile device constraints, such as soft keyboards with smaller key sizes, typing tends to be slower and more error prone. In tasks where the cost of errors is high being able to predict errors and mitigate their consequences is especially useful. For example when typing a password on a system that only allows a limited number of attempts before locking the user out of their account. The goal of this research is to expand on an existing cognitive modeling architecture, ACT-R, to be able to type on mobile devices.

This research draws from work in a number of different domains: text entry, cognitive modeling, and usable security, to create and evaluate the model. The first domain, text entry, has been studied on a number of different devices for many different tasks, beginning with the typewriter. This literature review begins with a summary of those text entry performance metrics and error classifications that will be used to evaluate the empirical data from the currently proposed studies. A summary of the work focusing on mobile devices that use the finger as the input device for text entry will be given. To review the literature in cognitive modeling, the past work modeling both typing and interactions with mobile devices in ACT-R will be presented. Then theories of how to accurately model the motor movement associated with tapping and typing on mobile devices will be discussed. Finally, a discussion of usable security asking both the question of why we care about special characters as part of passwords and how mobile typing ties in to usable security. This sets up the motivation for the proposed work.

Text Entry

Different text metrics have been developed to focus on two aspects of typing, speed and accuracy. The most frequently used measure of speed is words per minute (WPM). The calculation for WPM is shown in Equation 1 (MacKenzie & Tanaka-Ishii, 2010).

$$WPM = \frac{|T|-1}{S} \times 60 \times \frac{1}{5} \tag{1}$$

WPM is standardized so that every "word" is a five character long segment of the transcribed text (*T*). The first character is excluded from the calculation for greater accuracy, as there is variable visual search time at the beginning of the trial. Only the final text that is typed, including spaces and symbols, is considered the transcribed text; corrective actions and/or entered text that does not show up in the final submission are not considered. *S* is the time from the entry of the first character to the last character. An alternative, but less commonly used measure is keystrokes per second (KSPS) show in Equation 2 (MacKenzie & Tanaka-Ishii, 2010).

$$KSPS = \frac{|IS|-1}{S} \tag{2}$$

KSPS is advantageous in that it takes into consideration the entire input stream, *IS*, including backspaces, shifting, and corrected characters. Since it is not commonly used, making comparisons across studies can be difficult. If there is no corrective action taken on the input text then *IS* is equal to *T* and *KSPS x 12 = WPM*. KSPS can be considered the theoretical upper bound of input speed if all text is entered error-free.

When considering the accuracy of text input, Soukoreff and MacKenzie (2003) created a keystroke taxonomy for the components of the input stream. The purpose was to capture and quantify more than what is available in the transcribed text. The input stream is divided into four categories: Correct (C), Incorrect Fixed (IF), Fixes (F), and Incorrect Not Fixed (INF). Correct characters and Incorrect Not Fixed are the only two components of the transcribed text. Incorrect Not Fixed can be any type of error that a user commits: insertion, substitution, and omission. The two categories that are not present in the transcribed text that are in the input stream are the Incorrect Fixed and the Fixes. Incorrect Fixed are any errors that are made, and are noticed and corrected by the user. These corrective actions are any editing function and counted as Fixes. This taxonomy helps to clearly define the error rate, Equation 3, and its two components Not Corrected Error Rate, Equation 4, and Corrected Error Rate, Equation 5. These standardized error rates derived from the input stream allow better comparison across devices.

$$Total\ Error\ Rate = \frac{INF+IF}{C+INF+IF} \tag{3}$$

$$Not\ Corrected\ Error\ Rate = \frac{INF}{C+INF+IF} \tag{4}$$

$$Corrected\ Error\ Rate = \frac{IF}{C+INF+IF} \tag{5}$$

Analyzing the errors made in the input stream as opposed to the transcribed text provides a richer understanding of all of the errors that the subjects make. By definition, the input stream yields and equal amount of more data per trial than the transcribed text alone. Additionally, when subjects are instructed to enter text quickly and accurately they tend to fix most-if not all-errors (Soukoreff & MacKenzie, 2003). The main challenge in classifying errors in the input stream is that one must infer the intention of the subject. To address this, Wobbrock and Myers (2006) proposed a method-agnostic error taxonomy for classifying errors made in the input stream. Their method goes beyond looking at aggregate error rates and to a more fine-grained analysis to target problematic keys at the individual character level. There are three major categories for errors: insertion, omission, and substitution. Insertions occur when letters that appear in the transcribed text do not appear in the to be transcribed, or presented, text. Omissions occur when letters appear in the presented text and do not appear in the transcribed text. Substitutions occur when corresponding letters in the presented text do not agree with the letters in the transcribed text. These three categories are further broken down into whether the error was noticed and corrected or whether it remained in the final transcribed text. In addition to these error categories, there are two categories for error-free text. Correct text and text that had a corrective action taken on it even though it was not an error. A category they define that is not relevant to this work is recognition errors for gesture-based input.

This taxonomy is based on a set of four assumptions used to address the ambiguity in assessing the users intention. The first assumption is that subjects proceed sequentially through the presented text. This assumption helps resolve the ambiguity between substitutions and insertions. The second assumption is that subjects insert or omit only one character in a row. The third is that backspaces are made accurately and intentionally. The fourth is that omissions in the transcribed text are also omitted in the input stream. Wobbrock and Myers acknowledge that their taxonomy is method-agnostic. This work will further subdivide their categories to capture specific errors that are caused by the stimuli and the devices. To account for the fact that not all errors get corrected, Arif and Stuerzlinger (2010) proposed a model to predict the likelihood that subjects would notice and correct an error made on mobile devices based on the probability of perceiving the error and the cost to fix it.

A number of studies have been conducted on different mobile devices, with most of the studies motivations falling into two distinct categories. The first is to compare novel designs or input methods to existing ones as implementations for improved speed or accuracy. The second is to test theories of typing of to determine a theoretical upper bound for performance if perfect technology existed. Table 1 shows a summary of a number of studies that have examined the speed and accuracy of typing on touch screens with the finger as the input method. Input speed varies from eight to 50 WPM with error rates ranging from 5% to 25%. In experiments where subjects were told to not correct their errors, the typing speed was greater than ones where they were instructed to type as quickly and accurately as possible. Arif and Stuerzlinger (2009) found no significant relationship between experimentally manipulated error correction and input speed. Since mobile devices are used in many different settings, the way that they are held and typed on, or the input posture, is an important factor to take into consideration. Many subjects report having a preferred input posture but will use more than one posture and even switch between them as situational demands change(Azenkot & Zhai, 2012).

Table 1.

Summary of input speed and error rate from studies of text input on mobile devices.

Reference	Hardware	Operating System	Orientation	Input Posture	WPM	Error Instructions	Error Rate
Allen, McFarlin, and Green (2008)	iPhone	iOS	Portrait	N/C	6.43	Q&A	9%
Arif, Lopez, and Stuerzlinger (2010)	iPhone 3G	iOS	Portrait	N/C	5.92	Q&A	10%
Parisod, Kehoe,	iPhone	iOS	Landscape	2 Thumbs	4.69	Q&A	11%
and Corcoran (2010)	iPad	iOS	Landscape	N/C	8.91		5%
Azenkot and Zhai (2012)	Samsung Galaxy S	Android 3.2	Portrait	1 Finger	6.34	No Delete	8.17%
				1 Thumb	3.78		7%
				2 Thumbs	0.03		10.8%
Nicolau and Jorge (2012)	HTC desire	Android 2.1 or 2.2	Portrait	1 Finger	20	No Delete	6%
			Portrait	2 Thumbs	25		6%
			Landscape	2 Thumbs	29		3%
Castellucci and MacKenzie (2011)	Samsung Galaxy S Vibrant	Android 2.1	Portrait	2 Thumbs	1.4	Q&A	11.8%
Martin, Isokoski, Jayet, and Schang (2009)	Hewlett Packard iPAQ HX2490b	Windows Mobile 5	Landscape	2 Thumbs	8	Q&A	25%
Rudchenko, Paek, and Badger (2011)	-	Windows phone 7	N/C	N/C	40	No Delete	13%

Notes: *Q&A denotes Quickly and Accurately *N/C denotes Not Controlled

Cognitive Modeling

ACT-R (Anderson, 2007) is a multi-domain cognitive architecture for simulating and understanding human cognition. Researchers have had success in modeling a wide variety of tasks in ACT-R, ranging in complexity from basic psychology experiments to piloting an airplane. To have success in modeling this task, ACT-R's typing ability and touch screen interaction capabilities must be expanded. ACT-R's current typing ability is comparable to that of a moderately skilled touch typist. Moderately skilled is defined as typing about 30-40 WPM and knowing the location of the keys without having to look for them but not performing as rapidly as an expert typist (John, 1996). Das and Stuerzlinger (2007) built a model simulating expert text input on a cellular telephone with a 12-button telephone keypad using multi-tap as the input method. Multi-tap is a text entry system where the user presses a key to cycle through the letters associated with that key; for example pressing “6” twice would produce up an “N”. With the proliferation of smartphones, this older input method is much less common today as smartphones supply full keyboards as well as other methods for text input. In the past researchers have simulated aspects of touch screen interaction with some success. Salvucci, Taatgen, and Kushleyeva (2006) worked around using the finger as the input device by using the mouse instead. CogTool has been used to build models of ACT-R interacting with a touch screen (John, Blackmon, Polson, Fennell, & Teo, 2009). This implementation is an approximation of movement time and has not been validated for tasks on smartphones. There has also been work in done to extend ACT-R to user gestures to interact with a touch screen (Greene & Tamborello, 2013). ACT-R’s capabilities may not be complete and may have to be extended to accurately reflect typing on a smartphone. Previous work into modeling the motor component of touch screen interaction will be discussed in the following section.

One of the strongest and well-studied models of human motor movement is Fitts’s Law (Fitts, 1954). It has been used in HCI to model tasks that involve pointing with a mouse or tapping with a stylus. Fitts’s Law is a formulization of the speed/accuracy trade-off in aimed movement.

$$T = a + b\ log_2\left(\frac{A}{W} + 1\right) \tag{6}$$

Equation 6 is the MacKenzie (1992) formulation, and is the form commonly used in HCI. Fitts’s Law predicts the movement time to reach a target of size W at a distance of A. The constants, a and b, are empirically derived and reflect the efficiency of the system. In these traditional Fitts’s Law tasks the target is larger than the device being used to acquire it. When using a touch screen and a finger as the input device, the area that the finger touches can be larger than the target itself.

This case is called the “Fat Finger” problem and with it the predictive ability of Fitts’s Law declines.

To compensate for the decline in predictive ability of Fitts’s law, work has been done to examine at what target size the users start to miss the target more frequently (Parhi, Karlson, & Bederson, 2006), whether using the preferred hand as opposed to the non-preferred hand affects the speed and accuracy of target acquisition; and to see if final touch locations are biased in a particular manner (Perry & Hourcade, 2008). Henze, Rukziok, and Boll (2011) conducted one of the largest Fitts’s Law studies by deploying a game to the Android market and collecting data for over 15,000,000 separate touch events. Their analyses found an inverse logarithmic relationship between target size and error rate. They also examined the bias in final tap location to inform design of an algorithm to correct for it. These studies have been able to make design recommendations for the size of targets on the screen, and have also informed the design of touch detection algorithms to compensate for the typical errors that users make. To improve the predictive ability of Fitts’s law when using a finger as the input device, Bi, Li, and Zhai (2013) proposed a modification to Fitts’s Law called FFitts law. The theory states that the whole variability of end point error should be the sum of two independent distributions. The first distribution is the noise that naturally arises from the speed-accuracy trade-off. The second distribution is the absolute precision that arises from the human motor system and the implement used. The theory states that only the variability from the first distribution should be considered when predicting movement time and the other source of error should be removed from the calculation. The modified version of Equation 6 becomes Equation 7.

$$T = a + b \, \log_2\left(\frac{A}{\sqrt{2\pi e(\sigma^2 - \sigma_a^2)}} + 1\right) \tag{7}$$

In Equation 7 σ is the standard deviation of the whole end point distribution and σ_a reflects the input precision of the finger. The fit from this model was better than traditional Fitts's law models on a one-dimensional tapping task, a two-dimensional target acquisition task, and a single-word touch screen typing task.

It has been hypothesized that Fitts's Law can be used to calculate the movement time between keys on a soft keyboard (Zhai, Sue, & Accot, 2002). The modified version of Equation 6 becomes Equation 8.

$$MT = a + b\, log_2\left(\frac{D_{ij}}{W_j} + 1\right) \tag{8}$$

D_{ij} is the distance between keys *i* and *j* and W_j is the size of key *j*. When the movement time derived from this formula is averaged across all key combinations on a keyboard the theoretical upper bound for performance can be found for any keyboard configuration. MacKenzie and Soukoreff (2002) used this equation to formulate a theory of two-thumbed typing. In addition to timing between keys, the model makes assumptions about which thumb would be used for which character to predict the input time for different keyboards. The predictions their model generated were within 10% of the subjects' data. Clarkson, Lyons, Clawson, and Starner (2007) later extended this model to include dynamic thumb assignment to predict higher input speeds and lower error rates. While these models account for typing lowercase alphabetic characters, Sears and Zha (2003) proposed a Keystroke Level Model for predicting text entry time when switching between screens, which is a necessary part of the input process on most smartphone keyboards. This model is shown in Equation 9.

$$T = t_1 + [(t_d + t_r)c_t] + [t_k(c + c_s - c_t - 1)] \tag{9}$$

In this model *T* is the total time to complete the task, t_1 is the time for the first key press, t_d is the time to decide to switch the keyboard, t_r is the time to recover from a transition, and t_k is the time for all additional keystrokes not addressed by the other *t* values. *c* is the number of characters required in the task, c_s is the number of shifted characters and c_t is the number of transitions between keyboards required by the task The *t* values would be derived from empirical data and the *c* values are task-dependent. They found a strong correlation between the predicted data and the subject means. They recommended using this model as a tool for evaluating alternative keyboard designs.

Usable Security

When designing secure systems, the goals of security professionals and usability professionals are often at odds with each other. There is a growing recognition that the user is an integral part of the security of the overall system (Adams & Sasse, 1999). A widely deployed security measure is requiring a password to login to an account to access the information associated with that account. As users create more accounts with different systems, they are required to remember an ever-increasing number of passwords (Florencio & Herley, 2007). Many systems now set a minimum length requirement for passwords as well as require users to include special characters. Passwords of longer length provide more opportunities for input error. Inputting passwords on mobile devices presents a unique challenge; on a traditional desktop computer all characters are visible, where as on mobile devices characters may be two to three screens deep. High entropy secure passwords are unlike words used in traditional transcription typing experiments. To better understand the source and location of errors, results from preliminary experiments as well as a set of studies and models are proposed to examine this problem.

SMARTPHONE TYPING STUDY

Since typing strings of random characters had not been studied on mobile devices, a study was devised to collect data to better inform the changes that need to be made to ACT-R to model this type of typing.

Method

Subjects

Thirty Rice University undergraduates (19 female) participated in the experiment for credit toward a course requirement. The subjects' ages ranged from 18-24 years, with a mean age of 19.8 years. All subjects were right-handed. Twenty-four of the subjects owned smartphones and twenty of those twenty-four owned a model of the iPhone.

Apparatus

Stimuli for the experiment were displayed on a fourth-generation iPod Touch running iOS 5.1.1. Two identical iPods were used in the experiment. Subjects were seated at a table and asked to hold the device for the duration of the experiment. Subjects could rest their arms on the table but were not required to.

Stimuli and Design

During each trial a string of characters was displayed at the top of the screen in a white non-editable text field on a grey background. The string was present through the entire duration of the schedule. Below the stimulus text field was a second text field that acted as a password field. The password field was set so that when a character was typed, it was displayed for a short period of time before being masked by an asterisk. Below the two fields the keyboard was displayed so that it did not occlude either of the fields, regardless of whether the device was in portrait or landscape mode. A representative trial is shown in Figure 1 in portrait mode and Figure 2 in landscape mode.

Figure 1. An example trial from the Smartphone Typing Study in portrait mode.

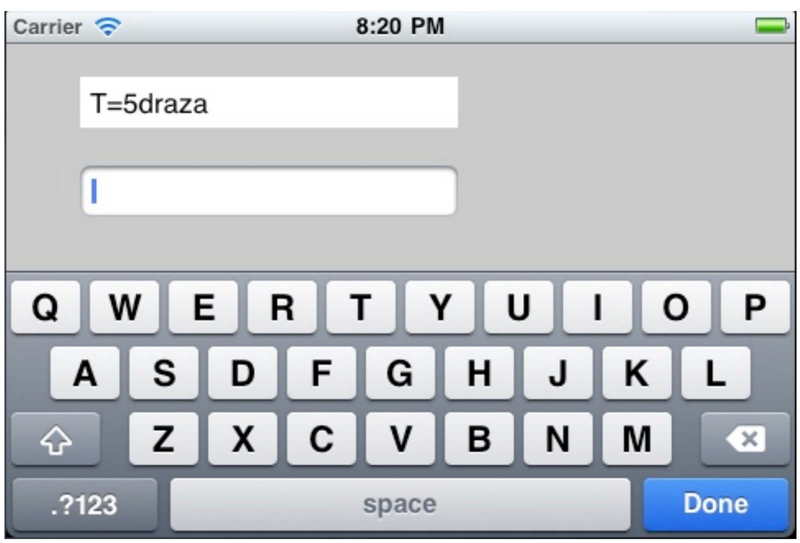


Figure 2. An example trial from the Smartphone Typing Study in landscape mode.

Device orientation was manipulated within subjects. The stimuli for the experiment were 16 randomly generated strings.

Device Orientation. Text can be input on the device in two orientations, portrait and landscape; refer to Figure 1 and Figure 2 respectively. The orientation of the device determines the size and spacing of the keyboard on the screen. To determine whether orientation had an effect on text input both orientations were used.

String Composition. Sixteen unique strings were generated for the subjects to input during the experiment. The strings were composed of a mix of uppercase characters, lowercase characters, numbers, and symbols. The strings varied in length from nine to fourteen characters. They were generated using PC Tools's strong password generator so that no dictionary words appeared in the string and there were at least two symbols in each string. Similar characters were excluded to improve readability of the string i.e. i, I, o, O, 1, 0, l. The strings that were used in this experiment are listed in Appendix A.

The characters that comprised the strings were grouped into six categories based on which page they were on, whether a shift key was required, and the type of character. These categories were Lowercase, Uppercase, Number, Symbol1, Symbol2, or Symbol 3. The categories Lowercase and Uppercase were the letters separated by case due to the shift key having to be pressed prior to inputting an uppercase letter. The category Number represented the numbers that were visible on the first page of symbols; Symbol1 represented the symbols that were visible on the first page of symbols, the same page as the numbers, and one screen deep. The decision to separate these two groups was due to the hypothesis that subjects would be more familiar with where the numbers are on the screen and in relation to each other than they do the symbols. Symbol2 represented the symbols that were visible on the second page of symbols or two screens deep. Symbol3 were the symbols that were visible on both the first and second page of symbols. Figure 3 shows the location of the keys categorized by their type. iOS does not allow designers to easily record screen switches and shift key presses. To examine the effect of multiple pages of characters and additional keystrokes on input time, the mean time between the keystrokes by category will be examined.

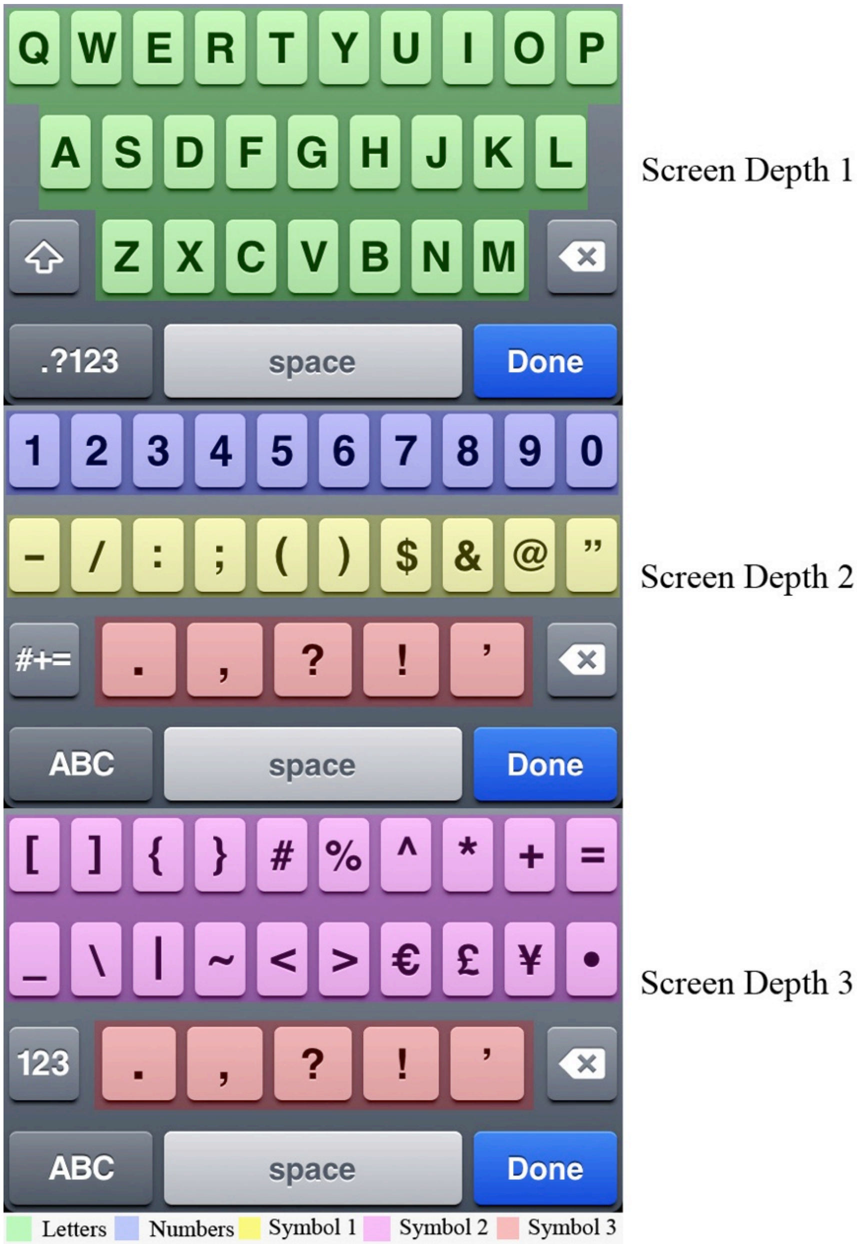


Figure 3. The categories of keys based on their screen depth and type for the Smartphone

The strings were presented twice in total, appearing once in block one and once in block two. The orientation of the device was counterbalanced; half the subjects input the strings in portrait mode in block one then landscape mode in block two with the orientations in the reverse order for the other half of the subjects. In both blocks the strings were presented in a random order. Before the two task blocks, subjects completed a practice block composed of five trials with five unique strings that were eight characters long to become familiar with the task. During the practice block subjects could use either orientation and were informed they could switch between the two. The duration of the experiment was approximately 30 minutes.

Procedure

When subjects were seated at the desk they were given an instruction sheet that explained the procedures of the task. The instruction sheet is shown in Appendix B. Subjects were instructed to type the displayed string into the password field exactly as it appeared as quickly and accurately as possible. If the string was typed incorrectly a notification appeared below the input field that informed the subject of this and asked to try again. After the string was input correctly or there were three unsuccessful attempts, the subjects were advanced to the next trial. At the beginning of the trial the keyboard was presented with all lowercase letters with the password field blank. Subjects did not have to tap on the text field to bring up the keyboard. When they had finished typing the string they were to press the done button located in the bottom right corner of the keyboard. At the beginning of each block an alert window appeared to inform the subjects what orientation they should hold the device in for the next block.

A time stamped output of the keystrokes was recorded throughout the task. All trials were completed in one session. Subjects were asked to fill out a short survey after completing the session. The survey contained questions about their computer use and experience, smartphone use and experience, and frequency of certain activities performed on their smartphone.

Results

Input Posture

Subjects were allowed to use whatever hand posture they liked for each orientation of the device as long as they used the same method for the duration of the block. Twenty-seven of thirty subjects used two thumbs to input the string in both blocks, one subject used only one finger in both orientations, and two subjects used only one finger for input on one of the blocks and two thumbs on the other.

Words per Minute

The subjects' average input speed was 8.23 WPM with a standard deviation of 1.86. As can be seen in Figure 4 the distribution of the mean WPM is approximately normal. Figure 5 shows the distribution of input speed separated by iPhone ownership. Owners of iPhones had a reliably slower input speed then subjects who had never owned an iPhone, 7.32 WPM versus 9.04 WPM respectively ($t(28) = -2.80$, $p = .01$, $d = -1.06$). The distribution of the input speeds for the different blocks is shown in Figure 6. The mean input speed for block one was 7.88 WPM and block two was 8.61 WPM. There was a reliable relationship between block number and input speed ($t(29) = -4.99$, $p < .001$, $d = -0.91$). There was not a reliable main effect of orientation with the subjects input speed ($t(29) = -.185$, $p = .85$, $d = -0.16$) or with preferred input method and orientation on input speed ($t(28) = .838$, $p = .41$, $d = 0.02$).

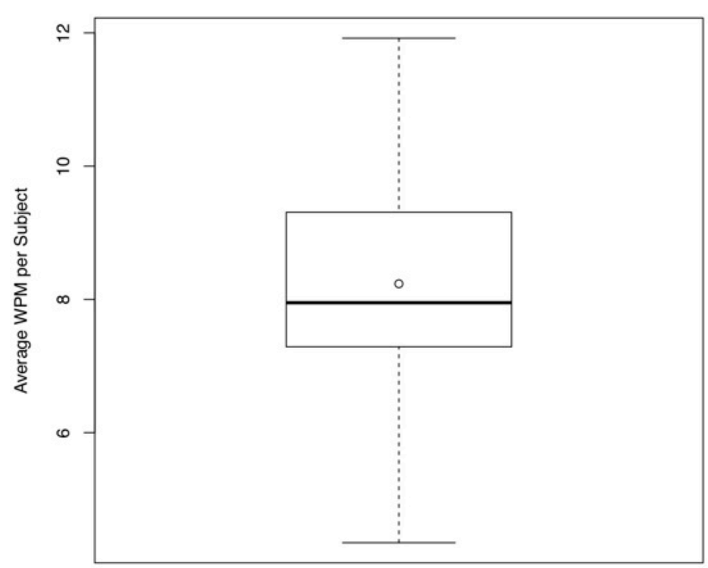


Figure 4. The distribution of the mean WPM in the Smartphone Typing Study.

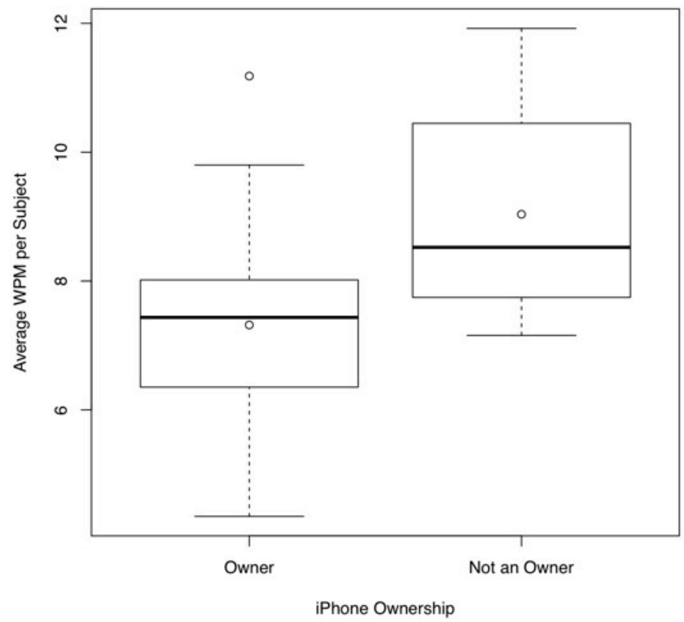


Figure 5. The distribution of the mean WPM by iPhone ownership in the Smartphone Typing study.

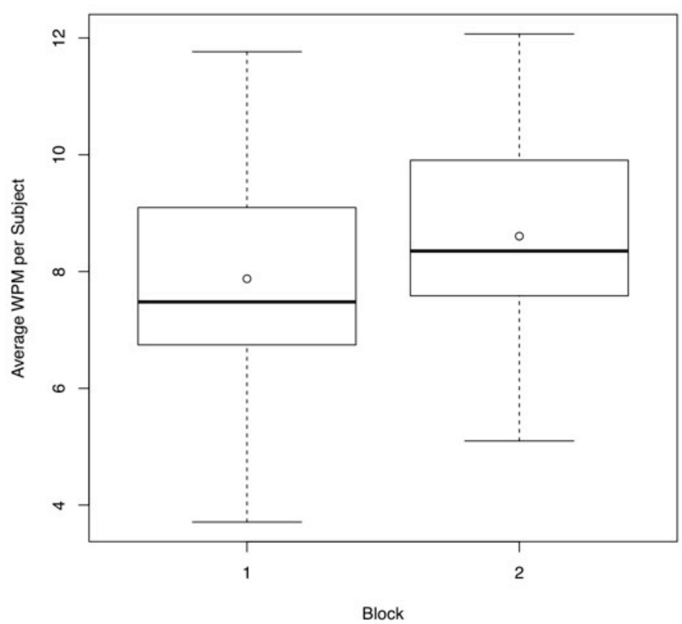


Figure 6. The distribution of the mean WPM between the two blocks in the Smartphone Typing Study.

Key Type Latency

Since the screen switches and shift key presses were not recorded, to examine the effect of multiple pages of characters and additional keystrokes on input time, the mean time between the keystrokes was examined by category. Table 2 shows the mean interkey interval depending on the previous character type. Since the pairings between all categories were not available for all subjects, the categories were transformed into the number of screen shifts require to get from key press to key press. For example Lowercase to Lowercase would be zero screen shifts where Symbol2 to Uppercase would be two. The mean interkey interval from all the different number of is shown in Figure 7. A repeated measures ANOVA was run to determine the effect of the number of screen shifts on the interkey interval. The effect of the number of screen shifts on the interkey interval is reliable ($F(2, 58) = 297.33$, $MSE = 0.04$, $p < .001$, $\eta_p^2 = .91$, Greenhouse-Geisser adjustment) and the linear contrast is significant ($F(1, 29) = 307.47$, $MSE = 0.38$, $p < .001$, $\eta_p^2 = .91$).

Table 2.
Mean interkey interval in seconds for all transition types represented in the iPod experiment.

	Preceding Character					
Typed Character	Lowercase	Number	Symbol1	Symbol2	Symbol3	Uppercase
Lowercase	0.44	1.41	1.48	1.56	1.45	1.12
Number	1.49	0.54	1.97	2.48	-	1.54
Return	0.71	0.73	0.79	0.72	0.66	0.86
Symbol1	2.01	2.01	1.84	2.69	0.98	1.56
Symbol2	2.47	3.06	0.78	0.42	2.54	2.62
Symbol3	1.45	2.04	-	-	3.86	1.60
Uppercase	1.51	2.22	1.99	2.01	1.84	0.95

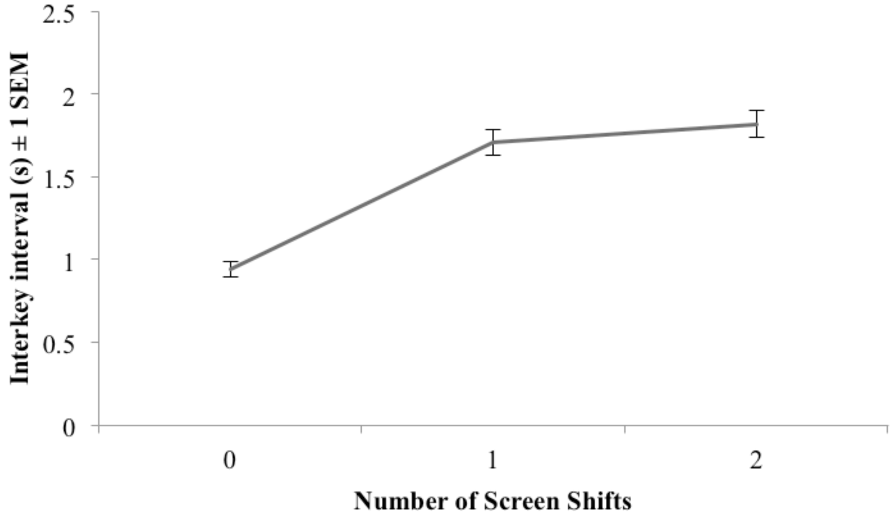


Figure 7. The mean interkey interval depending on the number of screen shifts.

Errors

Error Rate. To examine the error rates and types of errors that subjects made, common text entry metrics were used. The “not corrected error rate” had a mean of 1.46% with a standard deviation of 0.85 and the “corrected error rate” had a mean of 3.60% with a standard deviation of 1.51. A paired samples t-test showed that the mean error rate of the two different types are reliably different ($t(29) = -7.88$, $p < .001$, d = -1.53). The total error rate is the sum of these two types of errors; the mean total error rate was 5.06% with a standard deviation of 1.93. There was not a reliable main effect of iPhone ownership with the subjects total error rate ($t(28) = 0.42$, $p = .68$, $d = 0.15$). The mean error rate in landscape was 4.53% with a standard deviation of 1.96 and for portrait 5.53% with a standard deviation of 2.61. Figure 8 shows the error rate distributions for the two orientations. Orientation had a reliable effect on error rate ($t(29) = -2.18$, $p = .04$, $d = -0.41$).

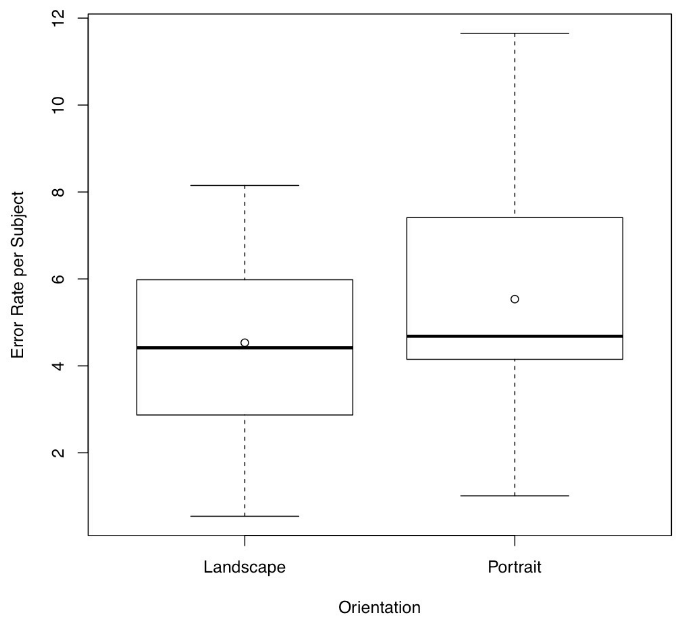


Figure 8. Error rate distributions in the two orientations in the Smartphone Typing Study.

Error Type. The input stream for each string was examined to determine what types of errors were made during input. Most of the errors were grouped into the common types: substitution, omission, duplication, and transposition. Two less common types, incorrect shifting and extra character, were also used. Substitution errors and omission errors were broken up into subcategories because of their high frequency of occurrence. The fours subcategories of substitution errors were: adjacent character (i.e. a character on a neighboring key to the target character on the keyboard was typed), incorrect screen (i.e. a character in the correct position as the target character was typed but at a different screen depth), visual similarity, and other. The three types of omission errors were missing character, missing ending punctuation, and early submission (i.e. submitting the string when it was only partially complete). The final category was that which encompassed correct characters that were removed by fix actions. Table 3 shows the percent of the total errors that fall into each category.

To examine the effect of orientation and error type on the number of errors made, a repeated measures ANOVA was run on orientation by error type. Characters that had a corrective action taken on them unnecessarily were omitted from the analysis. The mean number of errors for each type by device orientation is shown in Figure 9. The largest effect on the number of errors that occurred was type ($F(10, 290) = 25.17$, $MSE = 13.35$, $p < .001$, $\eta_p^2 = .47$ Greenhouse-Geisser adjustment). Orientation and the interaction of orientation on error type also had a reliable effect on the number of errors ($F(1, 29) = 8.98$, $MSE = 1.01$, $p = .006$, $\eta_p^2 = .24$, $F(10, 290) = 5.91$, $MSE = 3.06$, $p < .001$, $\eta_p^2 = .17$, Greenhouse-Geisser adjustment). A post hoc custom contrast was run to determine whether there were more adjacent character errors than the other types of errors combined. This relationship was determined to be reliable ($t(29) = -5.72$, $p < .001$, $d = -1.04$, Scheffe adjusted). Additionally the number of adjacent character errors made in each orientation were compared and the different was found to be reliable ($t(29) = -3.577$, $p = .001$, $d = -0.72$, Scheffe adjusted).

Table 3.
Error types and percentages for all errors made in the Smartphone Typing study. Subcategories are shown in italic.

Error Type	Percent of Total Errors	Error Count
Substitution	46.97	382
Adjacent Character	*35.45*	*234*
Incorrect Screen	*3.33*	*94*
Visual Similarity	*4.24*	*28*
Other	*3.94*	*26*
Omission	14.39	95
Missing Character	*4.70*	*31*
Ending Punctuation	*9.09*	*60*
Early Submission	*0.61*	*4*
Incorrect Shifting	14.24	94
Correct Character	14.09	93
Duplicate Character	3.94	26
Extra Character	3.33	22

Transposition	3.03	20
Total	100	660

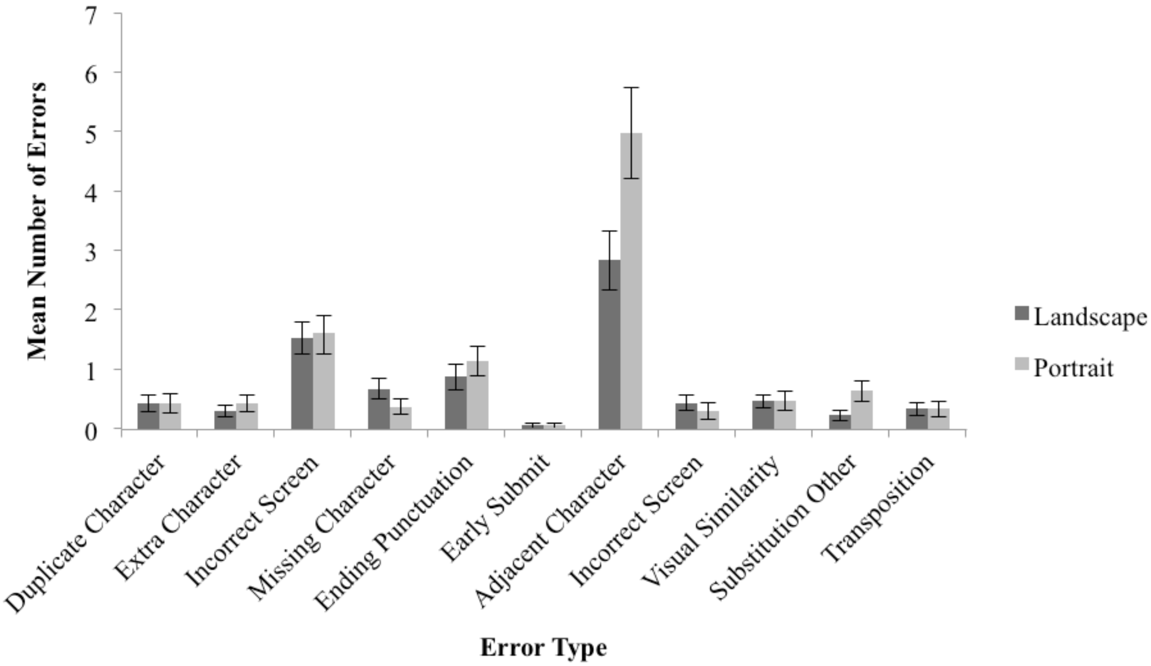


Figure 9. Estimated marginal means for each error type across each orientation.

Discussion

One of the surprising results of this study was that non-iPhone owners had a faster input speed than iPhone owners. One possible explanation is that iPhone users are more practiced with using iPhone specific features like autocomplete and autocorrect. When those functions are not available users may be slower because they are normally reliant on them. Another explanation is that iPhone users are more conscious of the difficulty of the task and are accordingly more careful. A baseline typing task on the smartphone may be able to elucidate the underlying cause of this effect. An additional effect that is worth exploring is that subjects are faster in block 2 than in block 1. This could be due to learning the location of the special characters. On the initial presentation subjects have to spend more time searching for the target character than on the later trial where they only have to recall it. The increase in interkey latency as the number of screen shifts increases is not surprising and being able to disentangle the differences between categories can further explain this effect. The difference in corrected error rate and not corrected error rate is consistent with past work. Past work has found that subjects notice and correct most of the errors that they make. Another unsurprising effect is that of the frequency of adjacent character errors with respect to other character types. The smartphone has a small keyboard making it likely that users will miss the target key they are trying to type. Additionally this error rate is higher in the portrait orientation as the keys are smaller than in the landscape orientation.

IPAD TYPING STUDY

To be able to model multiple types of mobile devices, the same experiment was run on an iPad with only slight alterations to the layout to be adapted to the device. Except where otherwise noted the method is the same. The only addition to this experiment was the introduction of a baseline typing task on the computer to examine the relationship between input speed on a physical and virtual keyboard.

Method

Subjects

The subjects in this experiment were 30 Rice University undergraduates (12 female) that participated for credit toward a course requirement. The subjects' ages ranged from 18-22 years, with a mean age of 20.1 years. Twenty-six subjects were right-handed and the other four were left-handed. Twenty-one subjects reported owning an iPhone and eleven of them reported owning an iPad.

Apparatus

Stimuli for the baseline-typing task were displayed on a Viewsonic VA503b 15” LCD monitor set at a resolution of 1024 x 768 pixels. Subjects were placed directly in front of the display and interacted using an Apple Keyboard. Two identical Macintosh mini 1.83 GHz Intel Core 2 Duo machines running Mac OS X 10.6.8 were used.

Stimuli for the experiment were displayed on an iPad 2 running iOS 6.0.1. Subjects were seated at a table and were to keep the device on the table for the duration of the experiment. Subjects were given a support for the iPad that angled the device so that it did not lie flat on the table and was in a more ergonomic position for use. Subjects were instructed to leave the device on the table during the experiment except, when it was necessary to change the orientation.

Stimuli and Design

Baseline Typing Task. To gather information about the subjects’ typing speed on a physical keyboard the subjects were presented with two timed typing tasks on a computer. The first task was one minute long and the second was two minutes long. The subjects were asked to type as much of the stimuli as they could as quickly and accurately as possible. The stimuli were stories in English. Four measures were calculated by the application for each task: words per minute, number of characters typed, number of errors made, and percent accuracy. This was done to test the relationship between input speed on the iPad and on a desktop computer since the iPad keyboard is more similar in size to a desktop.

iPad Typing Task. A representative trial is shown in Figure 10 in portrait mode and in Figure 11 in landscape mode.

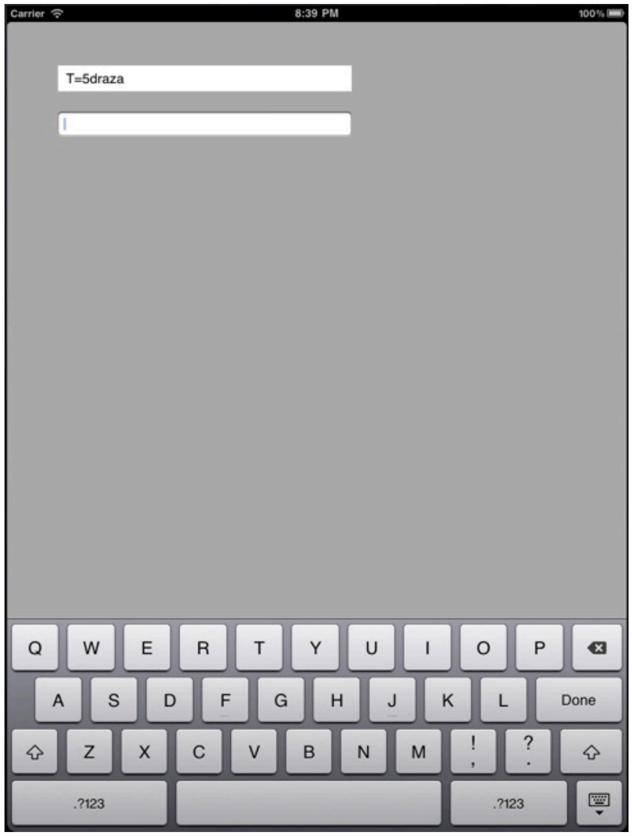


Figure 10. An example trial from iPad typing study in portrait mode.

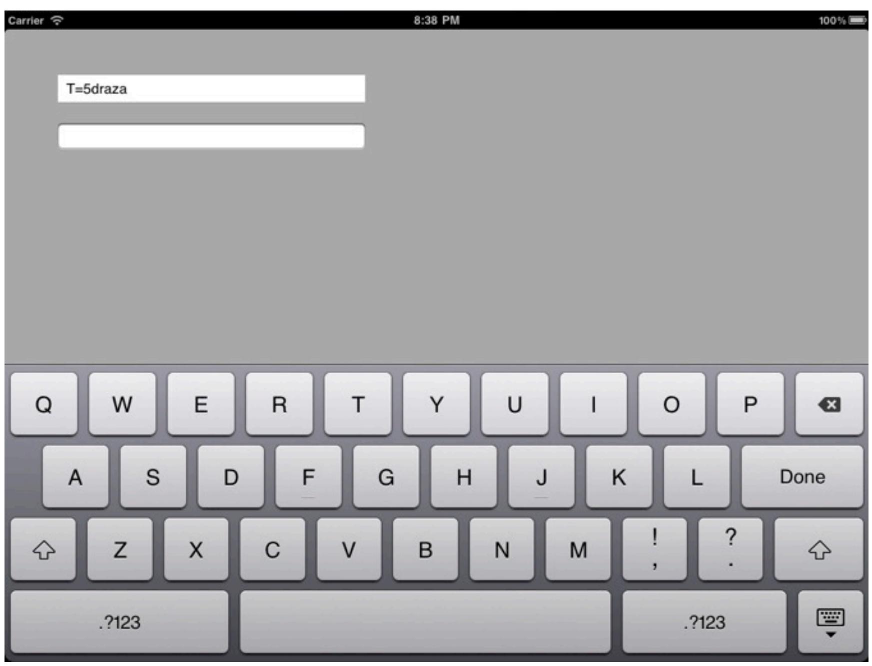


Figure 11. An example trial from iPad typing study in landscape mode.

String Composition. The keys were classified into the same categories with the exception of Symbol3. On the iPad the symbols that were in this category were visible on all three screens whereas on the iPod they were only visible on two screens. Figure 12 shows the location of the keys categorized by their type.

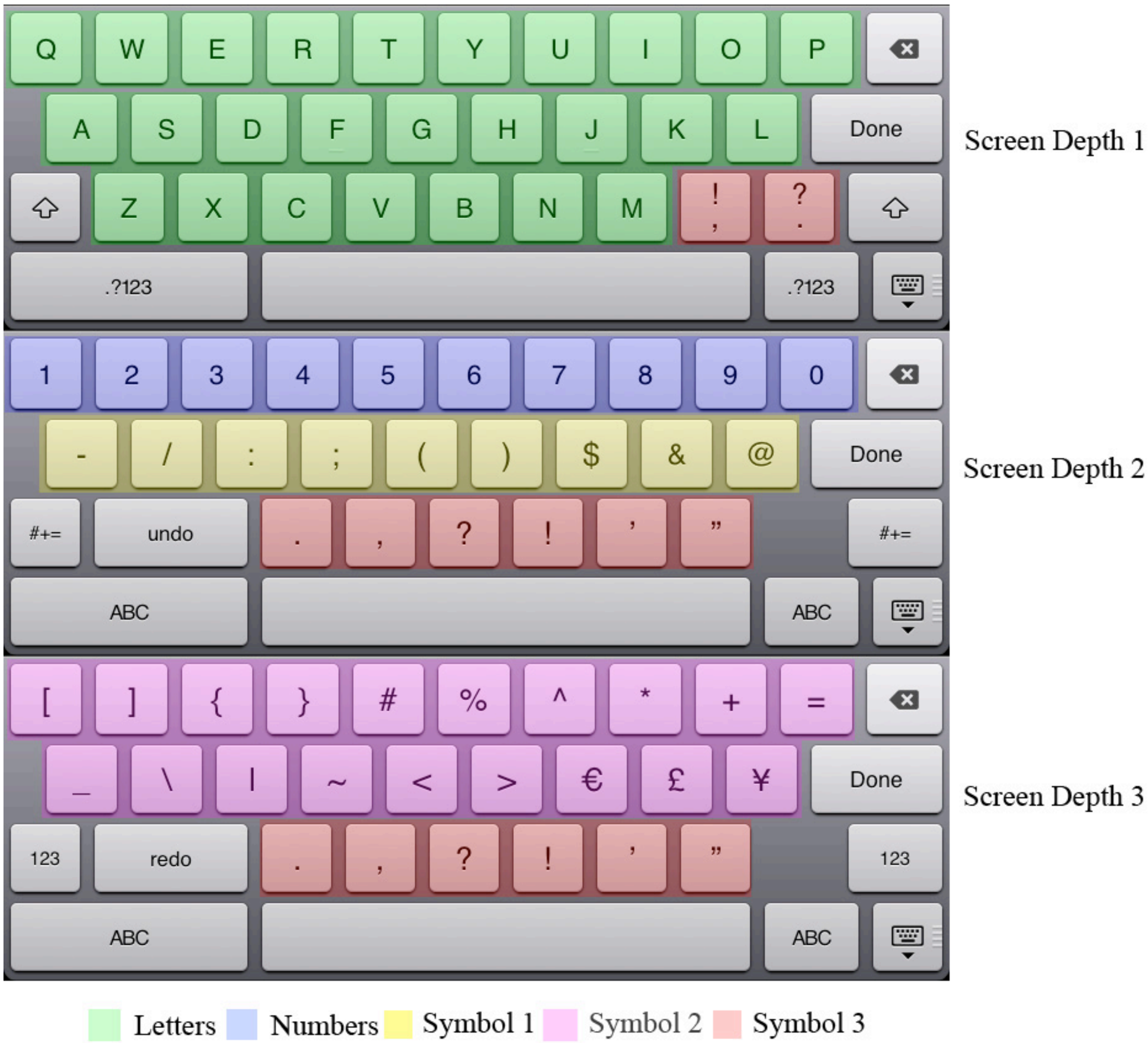


Figure 12. The categories of key types on the iPad.

Procedure

The subjects were first seated at the computer where they completed the two baseline typing tasks. After the subjects completed the baseline desktop typing task, they were given an instruction sheet that explained the procedure for the transcription typing task on the iPad. The instruction sheet is shown in Appendix B. One of the differences in the keyboard layouts between the two devices is that on the iPad the done button is located in the center on the right side of the keyboard and on the iPod it is located in the bottom right.

Results

Input Posture

The input posture that the subjects used in this experiment was constrained in the same manner as the iPod experiment. For input posture, twenty-seven subjects typed with one finger on each hand, two subjects used only one finger in both orientations, and one subject typed using all ten fingers.

Words per Minute

Figure 13 shows the distribution of the subjects' mean input speed on the iPad, which was 8.18 WPM with a standard deviation of 1.67. There was not a reliable relationship between iPhone owners and input speed or with iPad owners and input speed. Figure 14 shows the distribution of the subjects' mean input speed separated by block, with block one having a mean of 7.82 WPM and block two with a mean of 8.56 WPM. There was a reliable effect of block number on mean input speed ($t(29) = -3.12$, $p = .004$, $d = -0.58$). There was not a reliable effect of orientation on input speed ($t(29) = 1.025$, $p = .314$, $d = 0.19$). There was a moderate positive correlation ($r = .35$, $p = .058$) between typing speed on the computer and input speed on the iPad. The relationship was not reliable at the traditional .05 alpha level, but it is suggestive of an effect.

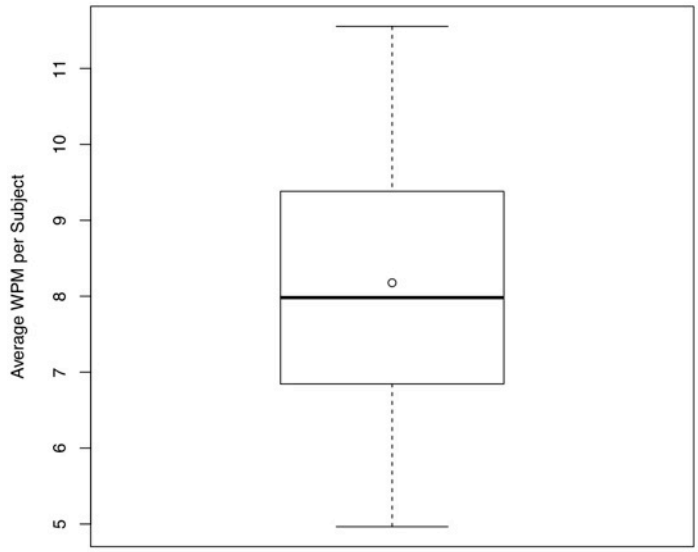


Figure 13. The distribution of mean WPM in the iPad Typing Study.

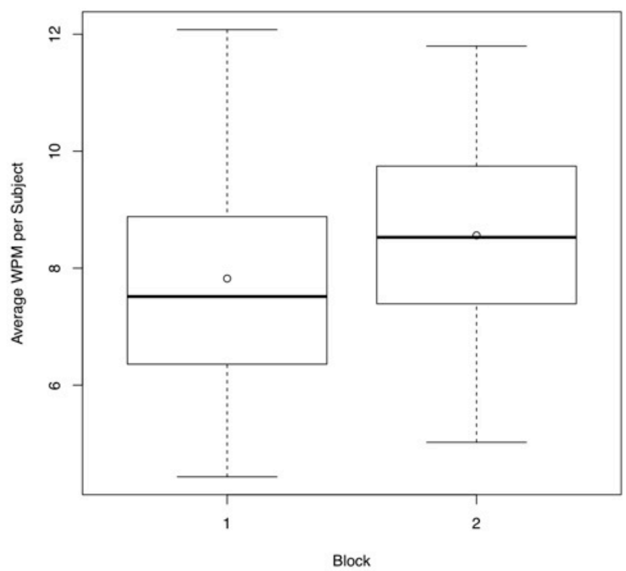


Figure 14. The distributions of mean WPM between by block in the iPad Typing Study.

Key Type Latency

The effect of additional keystrokes on the mean time between keystrokes was again examined by key category. Table 4 shows the mean interkey interval depending on the previous character type. Since the pairings between all categories were not available for all subjects, the categories were in the same manner. The mean interkey interval from all the different number of is shown in Figure 15. A repeated measures ANOVA was run to determine the effect of the number of screen shifts on the interkey interval. The effect of the number of screen shifts on the interkey interval is reliable ($F(2, 58) = 240.25$, $MSE = 0.06$, $p < .001$, $\eta_p^2 = .89$, Greenhouse-Geisser adjustment) and the linear contrast is significant ($F(1, 29) = 259.68$, $MSE = 0.06$, $p < .001$, $\eta_p^2 = .90$).

Table 4.
Mean interkey interval in seconds for all transition types represented in the iPad experiment.

Preceding Character						
Typed Character	Lowercase	Number	Symbol1	Symbol2	Symbol3	Uppercase
Lowercase	0.48	1.48	1.53	1.71	1.61	1.18
Number	1.50	0.65	2.20	2.49	-	1.69
Return	0.75	0.75	0.91	0.76	0.71	0.76
Symbol1	2.28	2.11	1.02	3.36	-	1.70
Symbol2	2.66	3.51	0.37	0.31	3.02	2.91
Symbol3	1.75	-	-	-	0.49	1.89
Uppercase	1.50	2.18	2.10	2.17	2.03	1.07

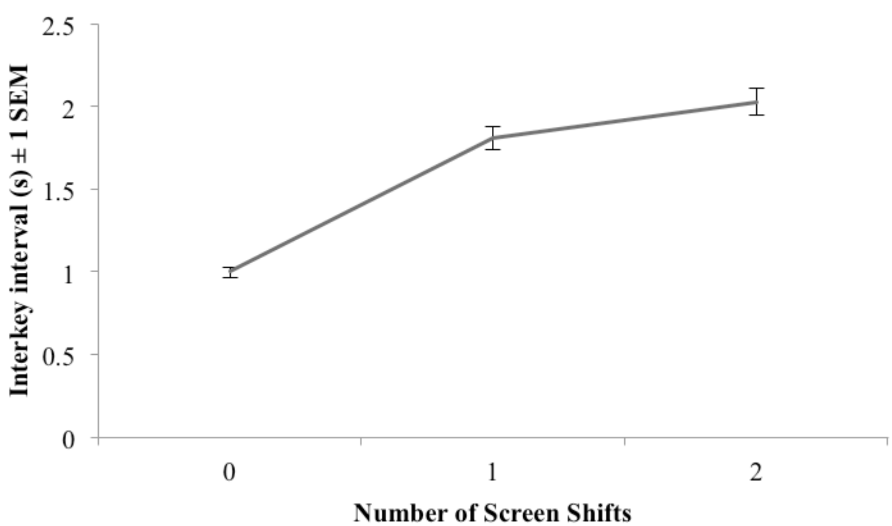


Figure 15. The mean interkey interval depending on the number of screen shifts.

Errors

Error Rate. The average not corrected error rate across all subjects was 2.65% with a standard deviation of 2.51 and the corrected error rate had a mean of 8.36% with a standard deviation of 3.38. A paired samples t-test showed that the means of these two error rates were reliably different ($t(29) = -6.91$, $p < .001$, $d = 1.496$). The average total error rate was 11.01% with a standard deviation of 3.88. There was not a reliable main effect of iPhone ownership with the subjects total error rate ($t(28) = 0.08$, $p = .94$, $d = 0.03$) or with tablet ownership with the subjects total error rate ($t(28) = -0.1$, $p = .92$, $d = -0.04$). The mean error rate in landscape was 10.63% with a standard deviation of 4.65 and for portrait 11.1% with a standard deviation of 5.38. Orientation did not have a reliable effect on error rate ($t(29) = -.394$, $p = .70$, $d = -0.07$).

Error Type. The input stream for each string was examined to determine what types of errors were made during input. The same error taxonomy that was used in the Smartphone typing study was used with one additional category. This category was created because subjects ended up using the clear function of the delete key in addition to just the delete function. The clear function is engaged by touching the delete key and holding down. This function was not always intentionally used and subjects would react to the input as if they had only deleted a character instead of everything they had typed on the trial that far. These errors were classified as CND, due to a clear not a delete being engaged. Due to the frequent occurrence of the clear function approximately 67% of the input classified as an error were correct characters that had a corrective action taken on them. Table 5 shows the percent of the total errors that fall into each category.

Table 5.
Error types and percentages for all errors made in the iPad Typing Study. Subcategories are shown in italic.

Error Type	Percent of Total Errors	Count
Substitution	8.16	115
Adjacent Character	*4.47*	*63*
Incorrect Screen	*1.28*	*18*
Visual Similarity	*1.13*	*16*
Other	*1.28*	*18*
Omission	19.72	47
Missing Character	*1.91*	*27*
Ending Punctuation	*6.88*	*97*
Early Submit	*1.63*	*23*
Incorrect Shifting	9.29	131
Correct Character	66.67	940
Duplicate Character	1.42	20
Extra Character	0.78	11
Transposition	1.21	17
CND	2.06	29
Total		410

To examine the effect of orientation and error type on the number of errors made a repeated measures ANOVA was run. The mean number of errors of each type by device orientation is shown in *Figure 16*. Correct characters that had a corrective action taken on them were excluded from the analysis. The type of error was the only reliable effect on the number of errors that occurred ($F(11, 319) = 17.48$, $MSE = 4.79$, $p < .001$, $\eta_p^2 = .38$, Greenhouse-Geisser adjustment). The effect of orientation and the interaction between orientation and error type were not reliable ($F(1, 29) = 1.81$, $MSE = 1.11$, $p = .19$, $\eta_p^2 = .06$, $F(11, 319) = .37$, $MSE = 1.71$, $p = .87$, $\eta_p^2 = .01$, Greenhouse-Geisser adjustment).

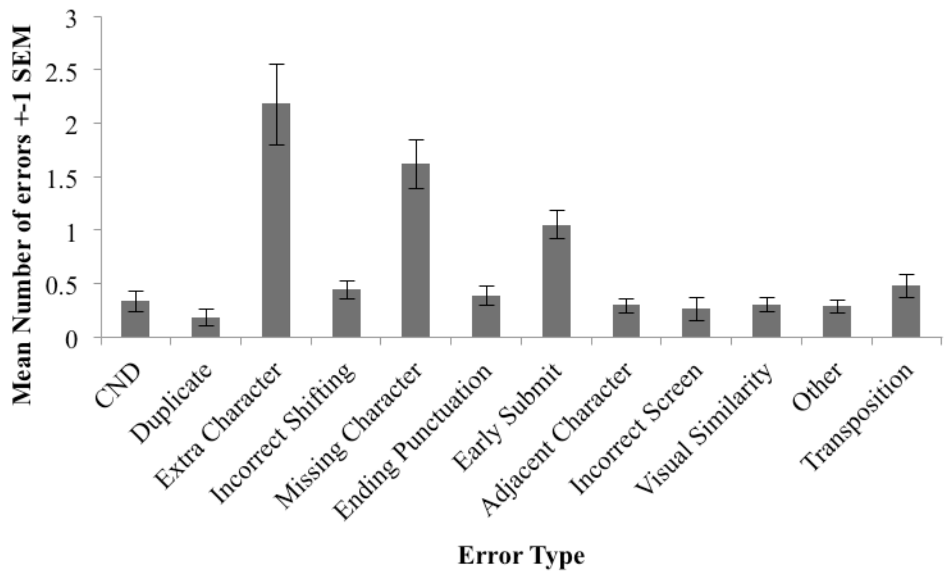


Figure 16. Mean number of errors in the iPad Typing experiment.

Discussion

One of the problems with this study that needs to be addressed in future work is the reason that so many subjects unintentionally cleared the text field. This is problematic due to the fact that it lowers input speed. One possibility to remedy this would be a better explanation of how the delete key functions to prevent unintentionally engaging the clear. Another possibility is to disable this function on the device. This error coupled with the frequency of the duplicate character error could point to a sensitivity problem with the touch screen. With a less sensitive screen it is likely that these errors would occur less frequently. The effect of block on input speed, the increase in interkey latency as the number of screen shifts increases, and difference in corrected error rate and not corrected error rate, are effects that are present in both experiments. These effects are likely to have the same underlying cause. The differences in frequencies of error types that occur between the two experiments provide further motivation to believe that the keyboard size is an important factor.

PROPOSED WORK

Cognitive Modeling

To successfully build a cognitive model of transcription password typing on mobile touchscreen devices there are three major steps that need to be completed: creating an environment for ACT-R to interact with an iOS device, making modifications to the motor module based on existing touch screen interaction theories, and modeling the strategies that subjects use to interact with the device. There are two potential methods of having ACT-R interact with an iOS device. The first way for ACT-R to interact with an iOS device is to create a replica of the application in Lisp that ACT-R can interact with directly. The environment would have to be visually and functionally similar to the application that the subjects used but would only need to replicate the portions of the operating system that are of interest. This method of creating an environment for ACT-R to interact with is commonly used when parts of a device interface can be easily replicated. The drawback to this approach is that it is not as complete a method and it is not as easily adapted to new tasks, as they will have to be coded as both an iOS application and in the Lisp environment for later models to interact with. The second way is to use a network connection to pass messages between the model and the iOS simulator or an iOS device connected to the Internet. This method is more commonly done when the environment is not easily replicated and a higher fidelity of interaction is necessary. This approach can be technically challenging but work has already been done to allow ACT-R to use the JavaScript Objective Notation (JSON) architecture to pass messages to other environments (Hope, Schoelles, & Gray, 2013). The choice between these two methods of interaction depends on which provides the requisite fidelity, ease of implementation, and whether the existing simulator allows for interfacing through a networked model.

Once an environment is built that a model can interact with, it will be necessary to modify ACT-R to be able to perform the task. Modifications will have to be made to the motor module in ACT-R. The motor interactions are currently based on mouse and keyboard input and do not take the size of the input device into consideration. This is important because when using a virtual keyboard, the input device, i.e. the finger, is a similar size to keys on the screen. The work of (Bi et al., 2013) can inform modifications that need to be made to ACT-R. While there are anthropometric data for the size of the last knuckle, there are no anthropometric data for finger pad size. Wang and Ren (2009) have looked at the different touch areas for different postures of touch interaction. Additionally, the proper parameters for Fitts' Law will have to be used to make sure that the predicted tap latencies are correct. Different parameters of the speed-accuracy tradeoff will have to be evaluated; due to the fact that on most keyboards the visually displayed size is not the actionable size of the key. This creates dynamic key sizing based on the system estimating the most likely key to follow the previously input key and adjusting the actionable size of the key based on the one it assumes to be most likely to follow. ACT-R currently operates under the assumption that the true target size matches the visual size.

A common strategy for this task is hunt-and-peck typing. This strategy is a tight loop that includes visually searching of the target key's location and then aimed movement to that location. While the layout of the letters is the same as a physical QWERTY keyboard and the layout of the numbers is similar, the layout of the symbols is different. The major difference between the two keyboards is that on a physical keyboard all the characters are visible where on the mobile devices there are three different screens. How ACT-R represents the memorized locations will have to be explored, especially for the symbols. An additional consideration when creating the model is the role of strategy. People using the most conservative strategy would check the input after each key is pressed to verify that it was correct. People using more aggressive strategies will have a number of characters input with only occasional verification or no verification at all unless the password is rejected. There is a tradeoff between the aggressive strategy and having to retype the password if it is rejected. The input attempt likely has an effect on the strategy that a person employs. If the system limits someone to three attempts before locking them out of their account, then on the third try accuracy may be valued more than speed. A conservative, aggressive, and mixed strategy will be implemented to explore how they affect input time and error rate.

Validation Study

The final aspect of the proposed work will be to run a validation study to compare human data with the predictions that ACT-R generates. While the previous study was not able to log every keystroke, this additional capability will be added so that more accurate estimates of visual search time are available. Keystrokes will be captured for shifting between screens and will include the x,y coordinates of the touch. Additionally the subjects' interactions with the device will be monitored with a video camera to provide additional information, such as where their fingers are during character search and how they alternate entry hands while inputting the string. The stimuli in the previous experiment were generated with a random password generator so the composition of the strings did not include all possible key type transitions. Rather than randomly generating the experimental stimuli, it may be better to use strings of random characters but control for to ensure a sufficient number of each transition type. A baseline-typing task of phrases from MacKenzie and Soukoreff (2003) will be used to compare input speed on transcription typing tasks with all alphabetic characters to the password stimuli. In the initial experiments the subjects were allowed to input text in their preferred finger posture. To be able to verify the latencies controlling input posture will give better information as you can ensure balanced groups.

The main purpose of this study is to test whether the ACT-R model predicts behavior that approximates the subjects' performance and to understand any discrepancies that may appear. Both text entry speeds and error rates will be compared between the model and the subjects. In addition to the standard text entry metrics, it may be worth considering a metric to address the number of screen shifts necessary to complete a string to evaluate the efficiency of input. If a string requires a character that is not at an easily memorable location and requires multiple views per page of symbols to locate, then the increase in page switches compared to the optimal number could aid in the evaluation of the usability of a password. With the recording of page switches added to the experiment, this analysis becomes possible.

The goal of this work is to create a validated model of typing on a touch screen. While there is an expansive body of work on interactions with touch screens, the field lacks an implemented model that combines all the work that has been previously done. Additionally many researchers have neglected characters that are not alphabetic. Since the model would not be limited to typing alphabetic characters it could also be used to test password policies and predict which passwords would be prone to error during input. Typing is a common subroutine on mobile devices, utilizing this model would aid designers when usability testing applications on mobile devices.

References

Adams, A., & Sasse, M. A. (1999). Users are not the enemy. *Communications of the ACM, 42*(12), 40-46.

Allen, J. M., McFarlin, L. A., & Green, T. (2008). An in-depth look into the text entry user experience on the iPhone. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 52*(5), 508-512. doi: 10.1177/154193120805200506

Anderson, J. R. (2007). *How can the human mind occur in the physical universe?* New York: Oxford University Press.

Arif, A. S., Lopez, M. H., & Stuerzlinger, W. (2010). Two new mobile touchscreen text entry techniques. *Poster at the 36th Graphics Interface Conference, 588*, 22-23.

Arif, A. S., & Stuerzlinger, W. (2009). Analysis of text entry performance metrics. *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, 100-105.

Arif, A. S., & Stuerzlinger, W. (2010). Predicting the cost of error correction in character-based text entry technologies. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 5-14. doi: 10.1145/1753326.1753329

Azenkot, S., & Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, 251-260. doi: 10.1145/2371574.2371612

Bi, X., Li, Y., & Zhai, S. (2013). FFitts law: modeling finger touch with Fitts' law. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1363-1372 doi: 10.1145/2470654.2466180

Castellucci, S. J., & MacKenzie, I. S. (2011). Gathering text entry metrics on android devices. *CHI '11 Extended Abstracts on Human Factors in Computing Systems*, 1507-1512 doi: 10.1145/1979742.1979799

Clarkson, E. , Lyons, K., Clawson, J., & Starner, T. (2007). Revisiting and validating a model of two-thumb text entry. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 163-166. doi: 10.1145/1240624.1240650

Das, A., & Stuerzlinger, W. (2007). A cognitive simulation model for novice text entry on cell phone keypads. *Proceedings of the 14th European conference on Cognitive ergonomics: invent! explore!*, 141-147. doi: 10.1145/1362550.1362579

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology, 47*(6), 381.

Florencio, D, & Herley, C. (2007). A large-scale study of web password habits. *Proceedings of the 16th international conference on World Wide Web*, 657-666.

Greene, K. K., & Tamborello, F. P. (2013). *Initial ACT-R extensions for user modeling in the mobile touchscreen domain.* Paper presented at the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University.

Henze, N., Rukziok, E., & Boll, S. (2011). 100,000,000 taps: analysis and improvement of touch performance in the large. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, 133-142. doi: 10.1145/2037373.2037395

Hope, R. M., Schoelles, M. J., & Gray, W. D. (2013). *Connecting ACT-R to the world with JSON over TCP.* Paper presented at the Proceedings of the 12th International Conference on Cognitive Modeling, Ottawa: Carleton University.

John, B. E. (1996). TYPIST: A theory of performance in skilled typing. *Human-computer interaction, 11*(4), 321-355.

John, B. E., Blackmon, M. H., Polson, P. G., Fennell, K., & Teo, L. (2009). Rapid theory prototyping: An example of an aviation task. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 53*(12), 794-798.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction, 7*(1), 91-139.

MacKenzie, I. S., & Soukoreff, R. W. (2002). A model of two-thumb text entry. *Graphics interface*, 117-124.

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. *CHI'03 extended abstracts on Human factors in computing systems*, 754-755.

MacKenzie, I. S., & Tanaka-Ishii, K. (2010). *Text entry systems: Mobility, accessibility, universality*: Morgan Kaufmann.

Martin, B., Isokoski, P., Jayet, F., & Schang, T. (2009). Performance of finger-operated soft keyboard with and without offset zoom on the pressed key. *Proceedings of the 6th International Conference on Mobile Technology, Application & Systems*. doi: 10.1145/1710035.1710094

Nicolau, H., & Jorge, J. (2012). Touch typing using thumbs: understanding the effect of mobility and hand posture. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2683-2686. doi: 10.1145/2207676.2208661

Parhi, P., Karlson, A. K., & Bederson, B. B. (2006). Target size study for one-handed thumb use on small touchscreen devices. *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, 203-210. doi: 10.1145/1152215.1152260

Parisod, A., Kehoe, A. , & Corcoran, F. (2010). *Considering appropriate metrics for light text entry.* Paper presented at the Fourth Irish Human Computer Interaction Conference, Dublin City University.

Perry, K. B., & Hourcade, J. P. (2008). Evaluating one handed thumb tapping on mobile touchscreen devices. *Proceedings of graphics interface 2008*, 57-64.

Rudchenko, D., Paek, T., & Badger, E. (2011). Text Text Revolution: A game that improves text entry on mobile touchscreen keyboards. *Pervasive Computing, 6696*, 206-213. doi: 10.1007/978-3-642-21726-5_13

Salvucci, D. D., Taatgen, N. A., & Kushleyeva, Y. (2006). *Learning when to switch tasks in a dynamic multitasking environment.* Paper presented at the Proceedings of the seventh international conference on cognitive modeling.

Sears, A, & Zha, Y. (2003). Data entry for mobile devices using soft keyboards: Understanding the effects of keyboard size and user tasks. *International Journal of Human-Computer Interaction, 16*(2), 163-184. doi: 10.1207/S15327590IJHC1602_03

Smith, A., SmartPhone Ownership – 2013 update. Pew Internet & American Life Project, June 5, 2013, http://pewinternet.org/Reports/2013/Smartphone-Ownership-2013.aspx, Accessed on August 6, 2013.

Soukoreff, R. W., & MacKenzie, I. S. (2003). Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. *Proceedings of the SIGCHI conference on Human factors in computing systems*, 113-120.

Wang, F., & Ren, X. (2009). Empirical evaluation for finger input properties in multi-touch interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1063-1072. doi: 10.1145/1518701.1518864

Wobbrock, J. O., & Myers, B. A. (2006). Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact., 13*(4), 458-489. doi: 10.1145/1188816.1188819

Zhai, S., Sue, A., & Accot, J. (2002). Movement model, hits distribution and learning in virtual keyboarding. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 17-24. doi: 10.1145/503376.503381

Zickuhr, K., Tablet Ownership 2013. Pew Internet & American Life Project, June 10, 2013, http://pewinternet.org/Reports/2013/Tablet-Ownership-2013.aspx, Accessed on August 6, 2013.

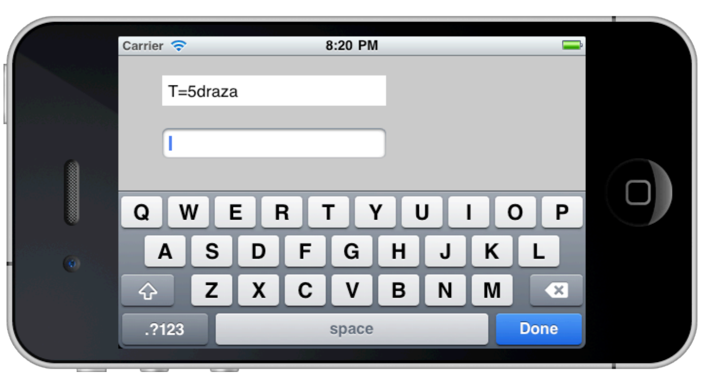
APPENDIX A: EXPERIMENTAL STIMULI

Stimuli	Length
t7$FrApR?	9
c>HA!*ac7	9
qEch<4da?	9
Vu.wap9$G	9
w6at_drAp?	10
j##u38\sp3f	10
Bu4ephA+e-	11
3p:A!a_UKAf	11
me+eSwA)2kup	12
$R3QAc!av%s2	12
sPU+r8jEq&VE.	12
xeZAr=3a9@w*	13
y_3&Z>s2mew=PH	14
V!d-as8*H_pHu6	14
H+maVu#am5w?Th	14
mu!rE4^ec(ey8x	14

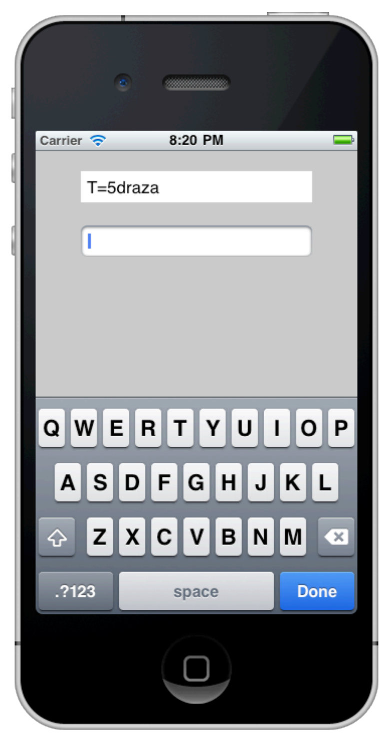
APPENDIX B: INSTRUCTION SHEETS FOR THE TWO EXPERIMENTS

Mobile Typing

The goal of this experiment is to study how people input text on mobile devices. Your task is simple: you will be given sample text, called the string, and asked to type the string into a text box on the iPod touch. The entire experiment will take place in an app on the device. For each trial you will be presented the string in a white box at the top of the screen. Your objective is to type the string exactly as it is presented in a text box that mimics a password field in the line below. Each character you type will appear as a dot shortly after it is typed as when entering a password. When you have completed typing the string you will press the done button located on the bottom right of the keyboard. Please try and type each string as quickly and accurately as possible. If you make a mistake you will receive feedback that the input was incorrect and be asked to try and type the same string again. You will have three tries to type each string correctly. You will complete 5 trials in a practice block to get orientated with the keyboard on the device. You will then complete two blocks of 16 trials holding the device in two orientations. Please stay within the app the entire time you are using the device and always input the string in the device orientation you are asked to. There are two orientations: portrait and landscape. Below, the device and an example screen are shown in each orientation. Please do not set the device down to type on it. You may type with either your thumbs or your fingers in either orientation but please select one style of input and use that for the entire block. If you have any questions please ask the experimenter now. Thank you.



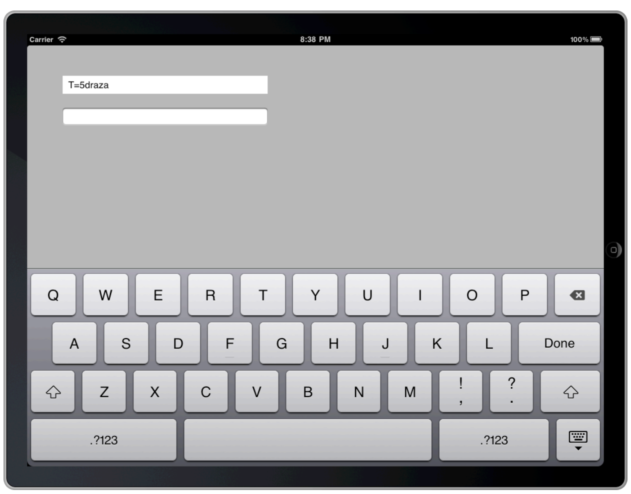
The device in landscape mode



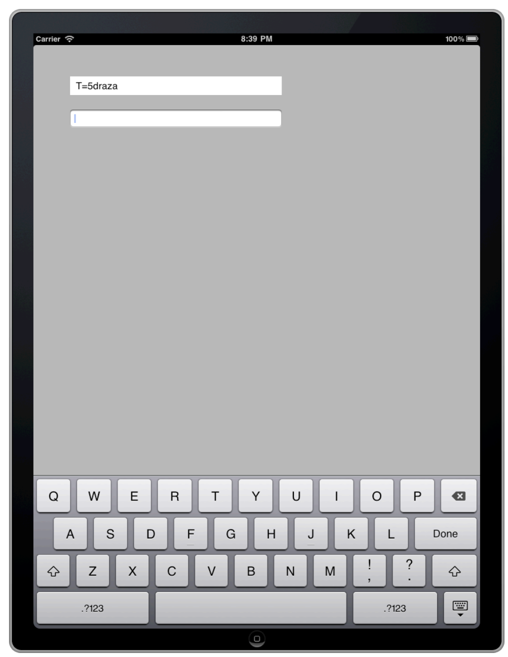
The device in portrait mode

Mobile Typing

The goal of this experiment is to study how people input text on mobile devices. Your task is simple: you will be given sample text, called the string, and asked to type the string into a text box on the iPad. The majority of the experiment will take place in an app on the device. For each trial you will be presented the string in a white box at the top of the screen. Your objective is to type the string exactly as it is presented in a text box that mimics a password field in the line below. Each character you type will appear as a dot shortly after it is typed as when entering a password. When you have completed typing the string you will press the done button located on the bottom right of the keyboard. Please try and type each string as quickly and accurately as possible. If you make a mistake you will receive feedback that the input was incorrect and be asked to try and type the same string again. You will have three tries to type each string correctly. You will complete 5 trials in a practice block to get orientated with the keyboard on the device. You will then complete two blocks of 16 trials holding the device in two orientations. Please stay within the app the entire time you are using the device and always input the string in the device orientation you are asked to. There are two orientations: portrait and landscape. Below, the device and an example screen are shown in each orientation. Please leave the device on the table to type on it. You may adjust the angle of the support but stick to one typing method per block. If you have any questions please ask the experimenter now. Thank you.



The device in landscape mode



The device in portrait mode