Text Entry with the Apple *iPhone* and the Nintendo *Wii*

Mauricio H. Lopez, Steven Castelluci, I. Scott MacKenzie
York University, Toronto, Canada

ABSTRACT

We conducted an empirical study with 16 participants comparing text entry speeds and error rates between the Apple *iPhone* and the Nintendo *Wii*, both using onscreen keyboards. Over eight blocks, the average entry speeds were 18.5 wpm for the *iPhone* and 9.2 wpm for the *Wii*. Error rates were 7.7% for the *iPhone* and 2.8% for the *Wii*. While the *Wii* fared very poorly compared to the *iPhone* on entry speed, its error rates were substantially lower. The lower error rates with the *Wii* were attributed to an overall tendency for users' to use slow and deliberate actions in controlling pointer position. An analysis of errors revealed that the majority of errors were on keys adjacent to the intended key. This was true for both devices.

Author Keywords

Text entry, keyboard, evaluation, onscreen keyboard.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces – evaluation/methodology.

INTRODUCTION

Text entry is an important aspect of human-computer interaction. While typically performed using a keyboard for desktop computing, text entry on home entertainment systems or mobile devices often uses an onscreen keyboard (a.k.a. virtual or soft keyboard). To enter text, the user selects the desired character from the onscreen keyboard using either a finger or stylus (in the case of handheld mobile devices) or a remote point-select device (in the case of home entertainment systems).

Because onscreen keyboards are only required for text entry, they can be hidden when not in use. This frees-up precious screen space for other purposes. Furthermore, an onscreen keyboard can dynamically change the layout and characters it displays based on the task (e.g., a numeric keypad to enter a phone number, a QWERTY keypad to enter a name).

This paper presents an evaluation of text entry on two recent and successful devices that use virtual keyboards: the Apple *iPhone* and the Nintendo *Wii*. Evaluating text entry on these devices is important since few evaluations of these devices exist, and little research exists with touchscreen devices in general [10]. Zhai [17] suggests that more research is needed to understand human performance and learning while using onscreen keyboards. We describe the text entry techniques used by the *iPhone* and the *Wii*, and review related work. Details and results of our evaluation follow. We then present our conclusions, as well as opportunities for future work.

iPhone Onscreen Keyboard

Touchscreens are touch-sensitive displays that allow users to directly interact with an interface. Selection is accomplished by simply touching the desired area of the interface with a stylus or finger. Thus, the need for a mouse, or other pointing device, is eliminated. By displaying an onscreen keypad, a physical keypad also becomes superfluous. Touchscreens also provide direct interaction, and allow the user to select, resize, and drag with their fingers.

A touchscreen's ability to facilitate input and display output in the same area is especially advantageous for tablet PC devices and mobile phones, which often have a compact form factor. The *iPhone*'s touchscreen occupies almost its entire face; there is no other form of selection and no keyboard for text entry. Figure 1a shows the typical, QWERTY form of the *iPhone* keyboard. (Additional layouts facilitate input of numbers, punctuation, and special characters.) Users enter text by touching the desired character with their finger or thumb. The *iPhone* onscreen keyboard also provides animation feedback as well as audio feedback to the user. Each key has a popup animation effect when it receives focus and a sound is played when the key is selected. Figure 1b shows the popup animation effect.

CHI 2009, April 4–9, 2009, Boston, MA, USA.

(a)

(b)

Figure 1: The *iPhone*'s onscreen keyboard (a) and the popup animation effect when a key is selected (b).

Wii Onscreen Keyboard

Nintendo's *Wii* video game console has garnered much attention due to its controller, the *Wii Remote* (henceforth referred to as the *Wiimote*). The *Wiimote* (Figure 2) uses an accelerometer to provide motion-based input for in-game actions. However, for pointing tasks, it uses an infrared camera. A series of infrared LEDs in the *Wii Sensor Bar* (placed above or beneath the television display) provide a point of reference for the *Wiimote*. The onscreen pointer's location corresponds to the *Wiimote*'s orientation to the *Sensor Bar*, and consequently, to the screen.

Figure 2: The *Wiimote*.

With an Internet browser and instant messaging functionality, text entry is important on the *Wii*. It is performed using an onscreen keyboard (Figure 3). To enter text, users point the *Wiimote* at the desired character, and press the selection button (the round "A" button on the top of the *Wiimote*). Again, additional onscreen keyboards allow entry of numbers, punctuation, and special characters.

Figure 3: The *Wii*'s onscreen keyboard.

Related *Work*

Touch screen onscreen keyboards

Though evaluations of text entry on touchscreens exist, very few involve the *iPhone*. One exception is by Allen et al. [2] who performed a comparative study between *iPhone* onscreen keyboard and the *BlackBerry* hard-key keyboard. They evaluated entry time and error rate. They found that the hard-key keyboard was more efficient and had and fewer errors. Unfortunately, they do not provide any results in terms of word-per-minute text entry speed.

Recent studies focus on the effect of tactile feedback on touchscreen devices. Hoggan, Brewster, and Johnson [4] present a study on the use of tactile feedback for a touchscreen keyboard. They evaluated and compared text entry speed on a physical keyboard, an onscreen keyboard, and an onscreen keyboard with enhanced feedback. The results suggest that adding tactile feedback to the device increases entry speed. While tactile feedback is possible with the *Wiimote*, we were unable to control such feedback on the *iPhone*.

Other studies have compared touchscreen devices with other text entry methods. Wobbrock, Myers, Aung, and LoPresti [15] evaluated text entry speed in wheelchair devices. They compared a touchscreen device and a joystick. They concluded that touchscreen devices are more efficient.

Point-and-click onscreen keyboards

Currently, there are no studies of the Nintendo *Wii* for text entry. Most studies used the *Wii* as a method to recognize gestures. *UniGest* [3] is an alternative text entry technique. This technique uses a gesture alphabet and a *Wiimote* to capture gestures. Gesture durations ranged from 296 to 481 ms. Combined with the proposed gesture alphabet, an upper-bound text entry rate of 27.9 wpm is possible. However, a full evaluation was not reported.

Text entry with game controllers

While other studies have used game controllers to enter text, none involves the *Wiimote*.

Költringer, Isokoski, and Grechenig [6] implement a method called *TwoStick*, which employs a non-QWERTY layout. Participants used the Microsoft *Xbox 360* game controller to enter text. A user study yielded initial text entry rates of 4.3 words per minute (wpm), increasing to 14.9 wpm after five hours of practice.

Wobbrock, Myers, and Aung [15] describe *EdgeWrite* which uses a game controller to enter text. They compared this method against a selection keyboard. Results showed that *EdgeWrite* was at least 1.5 times faster than a selection keyboard.

In the following section, we describe our methodology to evaluate and compare *iPhone* and *Wii* text entry using onscreen keyboards. Our results are intended to form a baseline from which follow-up design improvements and alternatives can be empirically compared.

METHOD

Participants

Sixteen paid participants (9 male, 7 female) were recruited from the local university campus. All were students or staff at the university with an age range of 18 to 30 years with an average of 24.7 ($SD = 5.7$). Two participants were left-handed. Four participants had previous experience with the *iPhone*, seven with the *Wiimote*. All were classified as novices, since they used either the *iPhone* or the *Wii* less than five hours per week.

Apparatus

The experiment was conducted in a quiet lounge. An *iPhone* and a *Wii* console were used for the experiment. A Samsung DLP Television model *HL-T50765* was used as display for the *Wii*. Because the devices support different programming languages, separate applications were written for each. The *iPhone* firmware was modified to allow access to the file system. This procedure was necessary since is it the only way to install third-party applications using functionality not supported by the original firmware. More information about the firmware modification method is available elsewhere [16]. We implemented the *iPhone* application using Objective-C and the free *iphone-dev* Toolchain [1]. The *Wii* application was implemented using Adobe *Flash*, which is a multimedia suite using ActionScript as a programming language. The application was accessed via the *Wii*'s *Internet Channel*. The two applications were designed to present the same appearance and interaction effects (e.g., audio feedback and popup keys when entering a character).

Figure 4 and Figure 5 depict our applications. The topmost text area displayed the presented phrase, while the lower one displayed the participant's transcribed text. Both applications featured audio feedback on a key press event (when a key is pressed) and a popup animation effect on a rollover event (when a key receives focus).

Due to programming difficulties in capturing the delete key events with the *iPhone*, we made the decision to disable error correction. These programming difficulties arose because the lack of documentation of some functions of the *iphone-dev* Toolchain. The applications included audio feedback when an error was made. This alerted the participant to the presence of an error and the need to synchronize the entered text with the presented text. Other text entry research has also used this method [9].

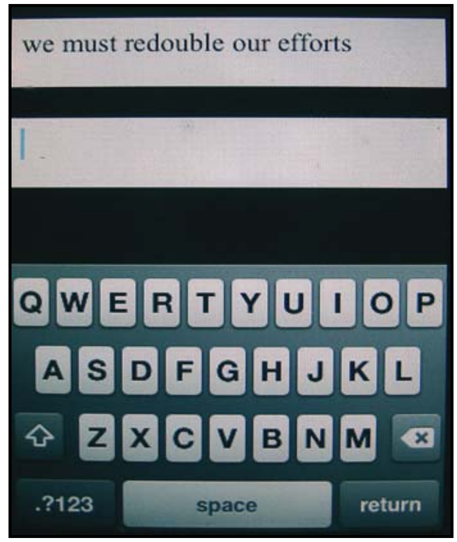

Figure 4: The *iPhone* application used for the experiment.

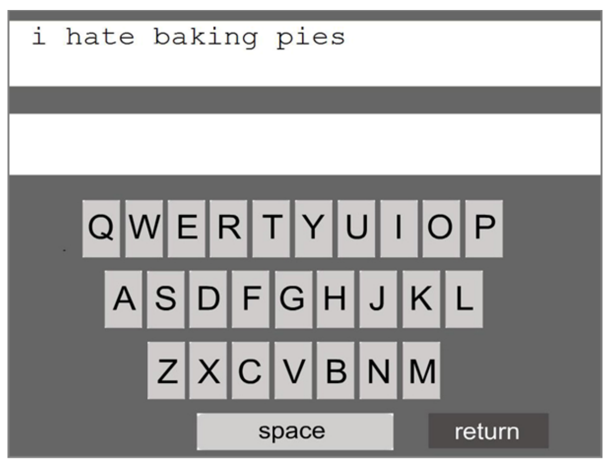

Figure 5: The *Wii* application used for the experiment.

Procedure

After signing a consent form and reviewing instructions for the study, each participant completed the first part of a questionnaire to gather demographic and usage data (previous experience with either the *iPhone* or the *Wii* onscreen keyboards). Participants were also asked to make predictions about each device's perceived ease of use, based on media advertising and word of mouth.

Participants were given basic instructions on how to enter text with each device. The instructions stated that they were to enter the presented phrases “as quickly and accurately as possible”. Because error correction was not allowed, participants were also instructed to keep their input synchronized (i.e., aligned) with the presented text. Audio feedback was added so that participants would hear a “beep” if they made an error or fell out of alignment. Participants were instructed to sit during the study. For the *iPhone*, they were instructed to use only one finger. For the *Wii*, each participant held the *Wiimote* with one hand, and used the “A” button for selection. The distance from the television was two meters. A new phrase appeared each time the participant clicked the return key. Each study lasted approximately 45 minutes. Participants were allowed to rest between blocks.

At the conclusion of the study, each participant completed the second part of the questionnaire. This consisted of choosing the best technique and rating each device on ease of use and their perceived learning curve. Figure 6 and Figure 7 show a participant during the experiment.

Figure 6: Participant doing the experiment with the *iPhone.*

Figure 7: Participant doing the experiment with the *Wiimote.*

Design

The experiment was a 2 × 8 within-subjects design. There were two independent variables: input device with two levels (*iPhone* and *Wii*) and block with 8 repetitions. Each block contained 3 trials of text entry. The phrases were chosen randomly from a 500 phrase set [8]. All instances of capital letters were converted to lowercase. The dependent variables were entry speed (measured in wpm) and error rate (measured in % error). In total, the number of phrases entered was 16 participants × 2 input techniques × 8 blocks × 3 phrases/block = 768.

The order of conditions was counterbalanced to neutralize possible learning effects. Participants were randomly assigned to one of two groups. One group entered text using the *iPhone* followed by the *Wii*, while the other group followed the reverse order.

Entry speed for each phrase was measured from the first key press event to the first return key event. Entry speed was calculated by taking the reciprocal of the mean movement time, multiplying it by 60 seconds per minute, and dividing by 5 characters per word [10]. Error rate was calculated using the MSD error rate formula given by MacKenzie and Soukoreff [12, Eq. 4], computed by dividing the number of incorrect keystrokes by the sum of incorrect and correct keystrokes, and multiplying by 100.

RESULTS AND DISCUSSION

Performance

Figure 8 shows the results for entry speed. On average, participants entered text at 18.5 wpm (SD = 1.13) using the *iPhone* and 9.2 wpm (SD = 0.71) using the *Wii*. In relative terms, the *iPhone* was 102% faster than the *Wii* – a considerable difference. The effect of input device on entry speed was statistically significant ($F_{1,14} = 75.8$, $p < .0001$).

The entry speed results of the *iPhone* are higher than other results reported in studies involving onscreen keyboards. Költringer and Grechenig [5] obtained an entry speed of 13.6 wpm using an onscreen keyboard on a graphic tablet, while Hoggan et al. [4] obtained an entry speed of 12.6 wpm using a Samsung *i718***.** As mentioned earlier, Allen et al.’s *iPhone* text entry evaluation [1] did not report any results in terms of word-per-minute entry speed.

The *Wii* entry speed is in line with other studies involving game controllers. Költringer et al. [6] obtained an average entry speed of 12.9 wpm using a Microsoft *Xbox 360* controller to enter text on an onscreen keyboard. But Wilson and Agrawala [14] obtained an average entry speed rate of only 6.5 wpm using a single analog joystick on a Microsoft *Xbox* controller.

These results suggest that using a touchscreen to enter text is by far faster than using a remote pointer-based technique such as the *Wii*. The reason for the significant difference is that selecting a target key is faster with a touchscreen. The proximity of finger and keys and the directness of interaction favor the selection time and the movement time for onscreen keyboards. Clicking a key using a *Wiimote*, on the other hand, requires more time, in part due to the greater distance between the onscreen keyboard and the controller. As well, a user must focus on the target key and maintain stability in the hand holding the controller. This increases the selection and movement time for an onscreen keyboard when interacting from a distance with an indirect pointing device.

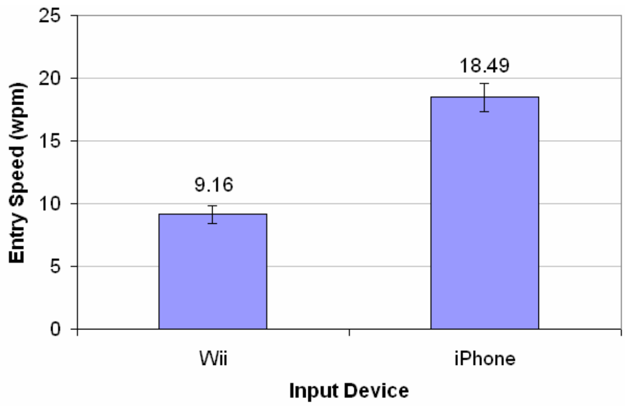

Figure 8: Entry speed (wpm) vs. input device (with standard deviations)

Small improvements in the entry speed across the blocks were observed in both input devices. Figure 9 shows the progression. Although the effect of block on entry speed was statistically significant ($F_{7,98} = 7.4$, $p < .0001$), we deemed the improvement small enough as to not be of practical significance. This is in part due to the relatively small amount of text entered. Had a full longitudinal study been conducted, there would be more improvement with practice.

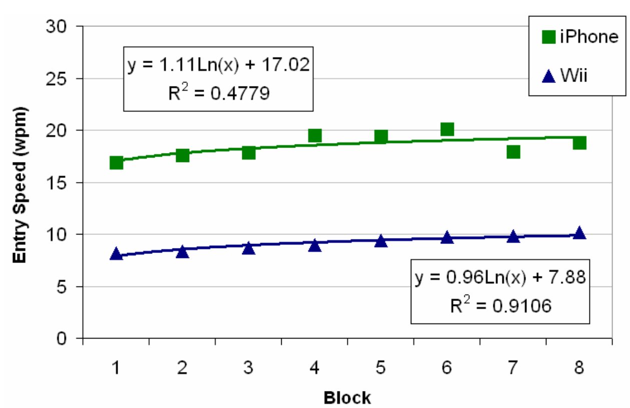

Figure 9: Entry speed (wpm) vs. block

One possible factor that affects the improvement over blocks is the size of the keys. To explain this better, we had a closer look at the types of errors for both devices.

Error Rate

Figure 10 displays the results for error rate. The error rate for the *iPhone* was 7.7% (*SD* = .85). The error rate of the *Wii* application was 2.8% (*SD* = 0.41). With such a dramatic different, it is no surprise that the effect of input device on error rate was statistically significant ($F_{1,14} = 15.4$, $p < .005$).

The error rate for the *iPhone* is higher to the results obtained by Költringer and Grechenig [5], which yielded an error rate of 4.1%.

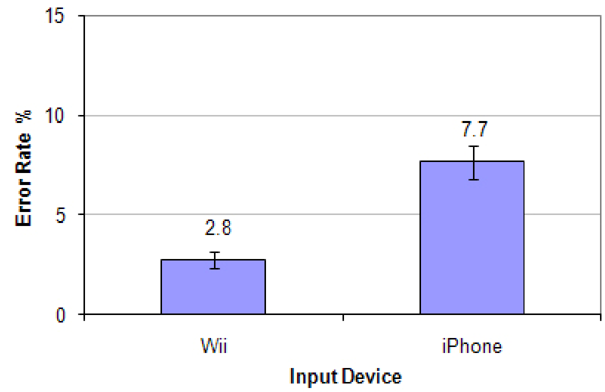

Figure 10: Error rate (%) vs. input device (with standard deviations)

The effect of block on error rate was not statistically significant ($F_{7,98} = 0.79$, ns), meaning that the error rate remained steady over time in both devices. Figure 11 shows the error rate change over blocks.

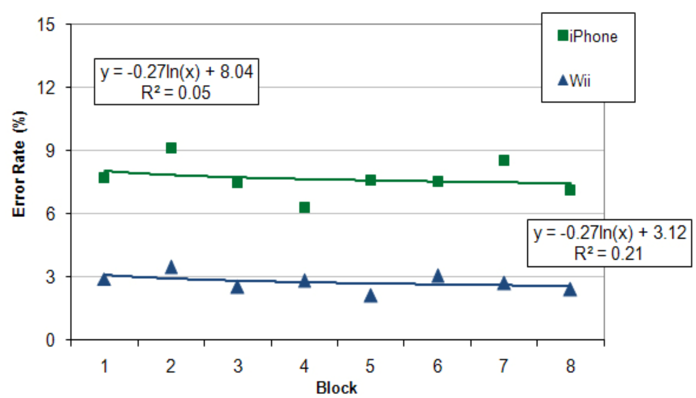

Figure 11: Error rate (% error) vs. block

The high error rate and the lack of improvement over blocks on the *iPhone* can be explained by analyzing the keys that had more errors. We performed this analysis using the following technique: All the phrases were analyzed and then the letters with more errors identified and counted. Finally, the number of occurrences of those letters was represented as a percentage of the total number of errors. Table 1 and Table 2 show the results of the analysis. The results only include the top 10 errors with more occurrences in case of the *iPhone*. For the *Wii*, we only show the top 9 errors, since the number of occurrences was the same beyond rank six. The first column shows the presented letter – the character that was supposed to be typed. The second column shows the entered letter – the letter that the participant entered instead of the correct one. The third column shows the number of occurrences (as %) of that error. The final column indicates if the typed letter was adjacent to the correct letter.

Table 1: Frequency of errors in the *iPhone*.

Presented Letter	Typed Letter	Occurrences (%)	Adjacent Key
O	P	19.46	YES
I	O	14.29	YES
E	R	6.65	YES
T	Y	5.17	YES
N	M	4.68	YES
Y	U	3.45	YES
H	J	2.96	YES
U	I	2.46	YES
K	L	2.22	YES
D	F	2.22	YES

Table 2: Frequency of errors in the *Wii*.

Presented Letter	Typed Letter	Occurrences (%)	Adjacent Key
O	I	8.44	YES
L	K	5.84	YES
A	S	5.84	YES
E	R	4.55	YES
M	N	4.55	YES
H	G	3.25	YES
S	A	3.25	YES
R	E	3.25	YES
R	T	3.25	YES

By checking the relative frequencies of letters of the English language [7], we noticed that all the presented letters in both tables have high frequencies. However, it is interesting to note that all incorrectly entered letters were adjacent to the presented letters.

As we mentioned before in the entry speed subsection, the size of the keys could be an important factor that affects the error rate and the entry speed over blocks. The size of a key in the onscreen keyboard of the *iPhone* is 5 mm high and 4 mm wide, and the spacing between the keys is only 1 mm. This is very small considering the size of the human fingertip. Perry and Hourcade [11] presented a study involving tapping targets of different sizes in a touchscreen. They found that the target size had a statistically significant effect on the accuracy. A target with size of 5.8 mm had a low average of only 79.8% accuracy. Others have called this the "fat finger problem" [13]. These results could explain why the error rate remains the same over blocks and the entry speed does not improve over blocks on the *iPhone*. Every time a participant enters text, he/she has the same chance of hitting an adjacent key due to the small size of the keys and the small spacing between them. At this point, it is instructive to show the mapping of all errors from both tables to support the hypothesis regarding the size of the keys. Figure 12 shows the mapping of all errors. The green keys represent all the presented letters. The red keys show the typed letters. A number indicating the place in the tables was placed between the corresponding pairs of letters.

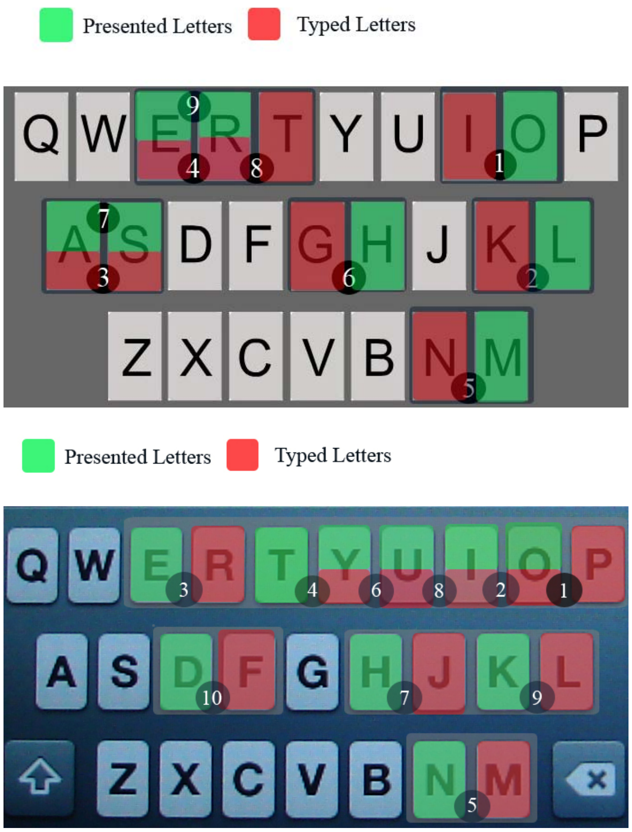

Figure 12: Mapping of the errors in both devices. First the *Wii* onscreen keyboard is shown, followed by the *iPhone* onscreen keyboard (see text for discussion).

Most of the errors in the *iPhone* were located in the upper section of the keyboard. This is similar to the results obtained by Allen et al. [2], who presented a similar figure showing the keys with more errors in the *iPhone*. It is interesting that there are five consecutive presented-typed pairs (T to P) out of ten with high error rates. This supports the idea that the size of the keys directly affects the error rates and the progression over blocks. Most of the errors for the *Wii* were randomly located. One possible explanation for this arrangement is that by using the *Wiimote*, the participants had more precision in selecting the desired key, since they spend more time finding the desired key and checking if that key was correct by observing the rollover effect. That is why error rates were very low – they are a bi-product of the extremely slow entry speed overall! The *Wii* also did not improve in error rate and the entry speed over blocks since the layout of the onscreen keyboard was similar to the onscreen keyboard of the *iPhone*; thus, it had the same problems such as small spacing between keys and the need for on-going attention from the participant.

Questionnaire

Table 3 shows the results of the questionnaire. All the ratings are out of five, with high scores being favorable. Before the beginning of the study, participants gave an average rating of 3.8 (*SD* = 0.7) to the *iPhone* in terms of perceived ease of use and an average rating of 4.1 (*SD* = 0.7) in terms of perceived learnability. The *Wii* received an average rating of 3.3 (*SD* = 0.9) for perceived ease of use and an average rating of 4.0 (*SD* = 0.7) for perceived learnability. After the study, participants gave an average rating of 3.4 (*SD* = 1.0) to the *iPhone* for ease of use, which is a not a significant change from the original rating. Furthermore, they gave an average rating of 4.1 (*SD* = 1.1) on adaptability, which also represents no change from the original rating. The *Wii* received an average rating of 3.2 (*SD* = 1.4) on ease of use, which is almost the same as the original rating. However, it received an average rating of 3.6 (*SD* = 1.4) on adaptability, which represents a small change from the original one.

Table 3: Means and standard deviations of the questionnaire responses. (1 = lowest rating, 5 = highest rating)

Questionaire Item	Before Testing		After Testing	
	iPhone	*Wii*	*iPhone*	*Wii*
The device is easy to use.	3.8 (0.7)	3.3 (0.9)	3.4 (1.0)	3.2 (1.4)
I can adapt easily to the device.	4.1 (0.7)	4.0 (0.7)	4.1 (1.1)	3.6 (1.4)

CONCLUSION

In general, the *iPhone* onscreen keyboard was much faster than the *Wii* onscreen keyboard. There was a statistically significant effect of the input device on entry speed and error rate. Knowing this relationship is important since there is little research on these types of keyboards. This is particularly true for the *Wii*, with is relatively new, novel, greatly hyped, but short on evaluations. Future work is intended to more thoroughly investigate learning over extended sessions of practice and to involve different devices using onscreen keyboards. Further work is needed to device and evaluation methods for small devices with small onscreen keyboard to reduce the number of adjacent-key errors.

ACKNOWLEDGMENTS

We thank our department's technical staff for allowing us to use the lounge facilities for the experiment.

REFERENCES

1. Allen, J. & McFarlin, L., & Green, T. 2008. An In-Depth Look into the Text Entry User Experience on the iPhone. *Proceedings of the 52nd Annual Meeting of the Human Factors and Ergonomics Society (HFES)*. New York City, NY.
2. Castellucci, S. J., and MacKenzie, I. S. 2008. UniGest: Text entry using three degrees of motion. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 3549-3554.
3. Hoggan, E., Brewster, S. A., and Johnston, J. 2008. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy, April 05 - 10, 2008). CHI '08. ACM, New York, NY, 1573-1582.
4. iPhonedev Toolchain: http://code.google.com/p/iphone-dev/
5. Költringer, T., and Grechenig, T. 2004. Comparing the immediate usability of graffiti 2 and virtual keyboard. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1175-1178.
6. Költringer, T., Isokoski, P., and Grechenig, T. 2007. TwoStick: writing with a game controller. In *Proceedings of Graphics interface 2007* (Montreal, Canada, May 28 - 30, 2007). GI '07, vol. 234. ACM, New York, NY, 103-110.
7. Lewand, R. 2000, Cryptological mathematics. Mathematical Association of America, Washington, DC.
8. MacKenzie, I. S., and Soukoreff, R. W. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 754-755
9. Matias, E., MacKenzie, I. S., & Buxton, W. One-handed touch-typing on a QWERTY keyboard. *Human-Computer Interaction, 11*, 1996, 1-27.
10. Soukoreff, W., & MacKenzie, I. S. 2002. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*. *17*. p. 147-198.
11. Perry, K. B., and Hourcade, J. P. 2008. Evaluating one handed thumb tapping on mobile touchscreen devices. In *Proceedings of Graphics interface 2008* (Windsor, Ontario, Canada, May 28 - 30, 2008). ACM International Conference Proceeding Series, vol. 322.

Canadian Information Processing Society, Toronto, Ont., Canada, 57-64.

12. Soukoreff, R. W., and MacKenzie, I. S. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA, April 05 - 10, 2003). CHI '03. ACM, New York, NY, 113-120

13. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. 2007. Lucid touch: a see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology* (Newport, Rhode Island, USA, October 07 - 10, 2007). UIST '07. ACM, New York, NY, 269-278.

14. Wilson, A. D., and Agrawala, M. 2006. Text entry using a dual joystick game controller. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada, April 22 - 27, 2006). R. Grinter, T. Rodden, P. Aoki, E. Cutrell, R. Jeffries, and G. Olson, Eds. CHI '06. ACM, New York, NY, 475-478.

15. Wobbrock, J. O., Myers, B. A., and Aung, H. H. 2004. Writing with a joystick: a comparison of date stamp, selection keyboard, and EdgeWrite. In *Proceedings of Graphics Interface 2004* (London, Ontario, Canada, May 17 - 19, 2004). ACM International Conference Proceeding Series, vol. 62. Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, 1-8.

16. Zdziarski, J.A. 2008. iPhone open application development, O'Reilly, Sebastopol, CA.

17. Zhai, S., Sue, A., and Accot, J. 2002. Movement model, hits distribution and learning in virtual keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 17-24.