A Reduced QWERTY Keyboard for Mobile Text Entry

Nathan Green, Jan Kruger, Chirag Faldu, and Robert St. Amant
Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8207 USA
+1 919 515 7938
stamant@cs.ncsu.edu

ABSTRACT
In this paper we describe a specialized keyboard for text entry that maps four rows of a standard keyboard onto the home row, with different characters encoded via modifier keys and multi-tap input. Use of the keyboard also relies on lexicon-based disambiguation. This design has two motivations: limiting physical space requirements and capitalizing on user knowledge of the standard QWERTY keyboard layout. The resulting "stick" keyboard is between 15% and 25% of the size of a standard keyboard. In a preliminary empirical study, users reached half of their normal typing speed using lexicon-based disambiguation (22.5 wpm) and a reasonable but lower speed with multi-tap input (10.4 wpm) with only a few minutes of practice.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation]: User Interfaces

General Terms
Keyboards, text entry, multi-tap, input devices.

INTRODUCTION
A significant limitation of mobile and special-purpose computing devices is the comparative inefficiency of text entry. Cell phones, PDAs, game controllers, and so forth have adopted a variety of different conventions for text entry, with correspondingly varying performance. Each software or hardware innovation in text input technology must address tradeoffs between several issues:

- *Input speed.* How closely does the new device approach the speed of the best devices already in common use?
- *Accuracy.* How accurate is the new device in comparison with existing devices?
- *Physical form factor.* Is it small enough (especially important for mobile devices and platforms intended for home/personal use)? Does it require a supporting surface, two-handed use, stationary use, and so forth? How much screen real-estate required?
- *Learning time.* How long does the device take to learn? Does it rely on existing skills (e.g., touch typing) or new skills?
- *Cost.* Does the device add significant cost to an existing system?

Text-entry input technologies trade off these factors in different ways. For example, a soft QWERTY keyboard is more easily learned but less efficient than layouts tailored to stylus use; multi-tap text input adds no hardware cost or size to a cell phone, but input speed is lower than for a standard keyboard. T9[1] improves on multi-tap, but the interaction is more complex and requires marginally more screen space. Folding keyboards for PDAs allow efficient text entry but hardware size is effectively doubled (or more, with the keyboard in use.)

We have developed a keyboard that addresses these tradeoffs in a novel way. We refer to the device as a "stick" keyboard, describing its shape. In the next sections we discuss the design of the keyboard and an empirical study of its performance. We conclude with a brief account of the potential uses of the technology.

THE STICK KEYBOARD DESIGN
Our design efforts focused on balancing three of the above factors: input speed, learning time, and physical size. Our thinking was particularly influenced by the interaction design of cell phones that use T9 input, which relies on a database of words to disambiguate cell phone keypad keystrokes that are associated with more than one letter. If we could design an input device that combined the best of the standard keyboard (fast text input, familiar layout) and the 12-key cell phone keypad (small size), we would have a device that could be used in off-the-desktop situations, potentially for extended typing tasks, with little degradation in performance.

Figure 1 gives our design. The upper diagram shows the layout of the stick keyboard, in which three rows of

CHI 2004, April 24-29, 2004, Vienna, Austria.
ACM 1-58113-703-6/04/0004.

[1] Description available at http://www.t9.com.

Figure 1. The stick keyboard, diagram and hardware prototype

alphabetical keys are merged into a single row. Numbers and symbols are also placed on the same row. Uppercase letters, numbers, and symbols are all accessed by modifier keys. The color coding in the diagram shows the relationship between the modifier keys and the character produced. The modifier keys behave in the same way as the Shift key on a standard keyboard, with the added feature that pressing a modifier key twice in quick succession locks that mode until it is turned off by another double key press. The Previous/Next key and Smart Type key support lexicon-based disambiguation and word completion functionality (which for conciseness we will refer to as lexicon-based interaction for the rest of this paper.) Function keys are not included.

The lower image in Figure 1 shows the physical prototype, the implementation of which was strongly constrained by budget limitations. We removed the keys from a standard keyboard until only an appropriately shaped subset remained. We shortened the space bar and put it in place, and relabeled all the keys. The resulting keyboard has similarities to a cell phone keypad, in overloading single keys with multiple characters (both numbers and letters), but the layout and proportions are derived from the QWERTY design.

The main input area of the original keyboard (i.e., the area bounded by the Shift key on the left, the Enter key on the right, the number keys and space bar on the top and bottom) measured 295 mm by 145 mm. The total footprint of the keyboard, including numerical keypad and other key groups, is over twice as large. The main input area of the stick prototype measures 210 mm by 47.5mm, 23% of the main area of the original keyboard. We expect that a high-fidelity physical prototype of the stick keyboard, with a border around the keys, would end up around 15% of the footprint of a full-sized desktop keyboard, a significant savings in size.

IMPLEMENTATION

Rather than build a general-purpose driver for the stick keyboard, we decided to simulate driver functionality within a single application on the PC. To do this, we built a specialized text editor application that supports standard editing activities, such as opening a file to edit, saving the current file, and creating a new file. We wanted our system (keyboard and editor) be as close to conventional systems as possible, for the sake of evaluation. We used the traditional keyboard and the Microsoft Windows Notepad application as the basis for our design, in order to make the system easy to learn, use, and remember.

Figure 2 shows the application. Although it follows desktop conventions, the interface could be much smaller (e.g., to fit

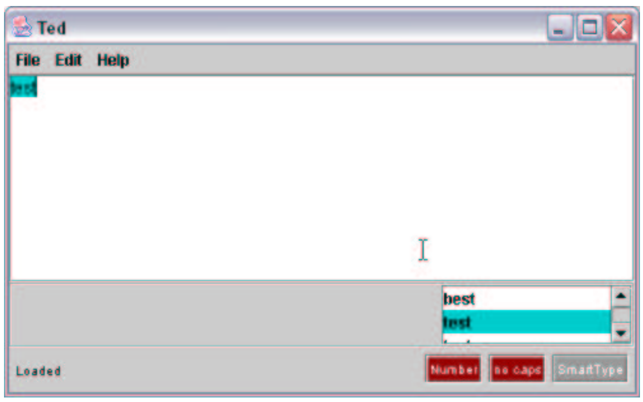

Figure 2. Text editor application

on a PDA or other specialized display) with no change in functionality. The application supports both multi-tap and lexicon-based interaction. In multi-tap mode, ignoring the modifier keys, the user taps each a key once, twice, or three times, in order to access one of the characters associated with the key. To support lexicon-based interaction, we generated our own database of about 30,000 words for the application, in an effort to mimic T9 functionality. In this mode, the user types in the usual way, but because of overloading, a sequence of keys does not represent a unique word. Matching words from the lexicon are displayed on screen; the Next and Previous keys are used to navigate between choices. The application visually indicates the status of the modifier toggle keys (e.g., Num Lock on or off) and whether lexicon-based interaction is active. An alternative solution beyond our capabilities would have been to present this information on the physical keys.

Our formative evaluation with the prototype was carried out with the help of six users. On first seeing the keyboard, the users were able to find the appropriate letters/characters quickly because they were familiar with standard keyboards, but they had the initial impression that multi-tap was the only input technique supported. We addressed this problem by introducing color-coding for the modifier keys and pointing out the keys supporting for auto-completion. With the help of a brief explanation, the users had little difficulty figuring out how to use the keyboard effectively.

During the formative evaluation, we asked users to type a single sentence into the application, three times using a standard keyboard and three times using the stick keyboard. Over all users, the average typing speed for the stick keyboard was 40% of the speed for the standard keyboard, with a minimum of 35% and a maximum of 54%. The users were generally pleased with their experience; the only complaint came from a system developer who missed a Control key. This could be addressed by rearranging the layout (e.g., vertically orienting the arrow keys to open a new key space on the bottom row), but this change has not yet been made.

EVALUATION

With promising feedback and good anecdotal performance in hand, we moved on to a more detailed (though still preliminary) evaluation, focusing on one question: *How fast can users type when encountering the stick keyboard for the first time, with minimal training, in comparison with their usual typing speed?* We are less concerned with absolute typing speed, because most keyboard users are not expert typists and will never approach a theoretical upper bound. At this point we are also more interested in first-time use than practiced use of the stick keyboard, believing that if users know that they can reach some acceptable level of performance on initial use, they will be more likely to adopt the device than another that requires much more practice.

We recruited ten participants, all working in technical occupations. Their ages ranged from 23 to 56; three were female, seven male. The participants were experienced though not expert typists, spending a mean of 4.75 hours at the keyboard every day (with a minimum of 1.5 hours for one participant and a maximum of 8 hours for two others). None of the participants was familiar with T9 interaction and none was a frequent user of multi-tap input.

User	**QWERTY**	**Stick + multi-tap**		**Stick + lexicon**	
1	74	13	17.5%	31	41.9%
2	65	12	18.4%	18	27.7%
3	57	8	14.0%	21	36.9%
4	40	14	35.0%	12	30.0%
5	53	7	13.2%	35	66.0%
6	39	10	25.6%	23	59.0%
7	27	12	44.4%	23	85.2%
8	54	9	16.6%	15	27.8%
9	40	9	22.5%	17	42.5%
10	46	10	21.7%	30	65.2%
Means	49.5	10.4	22.9%	22.5	48.2%

Table 1. Comparison of typing speed for standard and stick keyboard, in words per minute and as a percentage of typing speed on standard keyboard.

Each user was asked to carry out a 60 second typing test using a conventional keyboard and a fragment of expository English text. We then gave a short demonstration of the use of the stick keyboard. Each user repeated the typing test on a different text fragment, first using the multi-tap entry method and second the lexicon-based disambiguation method, in both cases using the stick keyboard. Users were allowed to experiment with the new keyboard for as long as they liked between the latter two trials, under the assumption that multi-tap experience would be more easily acquired than interaction with the lexicon. User experimentation lasted up to several minutes in each case. Limitations of this preliminary study include no treatment of error rates and no counterbalancing between multi-tap and lexicon-based input.

Table 1 shows the results. The stick keyboard, with multi-tap input, allowed users to type 10.4 words per minute (wpm), 22.9% of their standard typing speed. Using lexicon-based interaction, users averaged 22.5 wpm, 48.2% of their standard typing speed.

Our results are most directly comparable to T9 interaction. They are consistent with Tegic Communication’s findings that T9 is twice as fast as multi-tap. The absolute speed for the stick keyboard compares favorably with existing studies of T9 use. (Presumably because cell phones do not compete with standard keyboards, we have found no studies of per-user performance differences between T9 and QWERTY.) In one of the few empirical studies on T9 carried out to date, James and Reischel (referred to here as J&R) measured typing speeds for multi-tap and T9 interaction, for novice and expert users in T9 interaction [3]. The stick keyboard users in our study outperformed J&R's novices and experts by a good margin, on roughly comparable text. J&R's novices typed 7.98 wpm using multi-tap, 9.09 wpm using T9; the numbers for experts were 7.93 and 20.36 wpm. Our stick keyboard users were about 30% faster using multi-tap than either novices or experts in the J&R study. For lexicon-based interaction, our analog to T9 interaction, stick keyboard users were 10% faster than expert T9 users and 147% faster than novice T9 users. Keeping in mind that our users had no experience with T9 or the stick keyboard prior to our study, we find these comparisons significant.

The stick keyboard also performs competitively with other non-standard input techniques on initial use. The Keybowl requires some 5 hours of use to reach 50% of QWERTY typing speed [7]; the half-QWERTY keyboard requires 8 hours [6]. Handwriting recognition speeds range are around 16 to 18 wpm for walk-up use [1]. Based on what little empirical data is publicly available, initial text entry rates for miniature QWERTY keyboards for two-finger or thumb typing appear to be about the same as for the stick keyboard.[2] Initial soft keyboard typing speeds for QWERTY layouts are higher than for the stick keyboard (e.g., 28 wpm [5]), but at the cost of increased use of screen space. Qualitative tradeoffs make straightforward comparisons difficult, but the stick keyboard appears to occupy a promising niche, if our expectations about practiced performance turn out to be true.

We have not yet carried out an evaluation of extended use of the stick keyboard, but based on the anecdotal experience of the system's developers, we are confident that practice will give significant performance improvements. Two factors are relevant in predicting the comparative performance of the stick keyboard by expert users after extended practice. First, we expect the stick keyboard to be faster than forms of text input that rely on one or two fingers or a stylus, all other things being equal. Input actions are distributed over all ten fingers, which means that they can proceed to some extent in parallel, as with typing on a standard keyboard.[3] Second, in contrast to cell phone keypads and other layout variations, the stick keyboard allows QWERTY typing knowledge and experience to transfer directly.

[2] One example is the RIM keyboard, at http://www.rim.net; the Delta II, at http://www.chicagologic.com/, has informal performance data associated with its online description.

[3] Calculations based on Silfverberg et al.'s model of text entry for mobile devices [8] show that even if the stick keyboard were used with a single finger as with cell phones, the proportions of the keyboard (in particular the ratio of key size to the average distance between keys) are such that expert performance would be degraded by only around 15% compared with cell phones.

CONCLUSION

The stick keyboard combines some of the best properties of existing text entry technologies. It will not replace devices that already require a numerical keypad, such as cell phones, or those that must be held in one hand for use, but not all mobile applications have these requirements. For example, in-car systems with GPS functionality could use a small keyboard rather than on-screen typing; for PDAs, the stick keyboard should be competitive with folding keyboards with respect to storage and in-use space requirements; we can even imagine a specialized text messaging device based on the stick keyboard and a small built-in screen. In general, when users have keyboard layout knowledge, space is limited but some type of two-handed input on a fixed surface is possible, and reasonable typing speed is desired, we believe that the stick keyboard will be an input device alternative worth considering.

ACKNOWLEDGMENTS

Thanks to three anonymous reviewers, who pointed out further directions for this research and more detailed evaluation.

REFERENCES

1. Bohan, M., Phipps, C. A., Chaparro, A, and Halcomb, C. G. (1999). A Psychophysical Comparison of Two Stylus-Driven Soft Keyboards. *Proceedings of Graphics Interface*, pp. 92-97.
2. Isokoski, P., and Raisamo, R. (2000). Device Independent Text Input: A Rationale and an Example. *Proceedings of Advanced Visual Interfaces*, pp. 76-83
3. James, C. L., & Reischel, K. M. (2001). Text input for mobile devices: comparing model prediction to actual performance. *Proceedings of CHI '01*, pp. 365-371.
4. MacKenzie, I. S. (2003). Motor behavior models for human-computer interaction. *HCI models, theories, and frameworks*, Carroll, J. (ed). Morgan Kaufman.
5. MacKenzie, I. S., & Zhang, S. Z. (1999) The design and evaluation of a high performance soft keyboard. *Proceedings of CHI '99*, pp. 25-31.
6. Matias, E., MacKenzie, I. S., & Buxton, W. (1993). Half-QWERTY: A One-Handed Keyboard Facilitating Skill Transfer from QWERTY. *Proceedings of INTERCHI '93*. p.88-94.
7. McAlindon, P. J., Stanney, K. M., & Silver, N. C. (1996). The Keybowl: An Ergonomically Designed Document Processing Device. *Second Annual ACM Conference on Assistive Technologies*, pp. 86-93.
8. Silfverberg, M., MacKenzie, I. S., & Korhonen, P. (2000). Predicting text entry speed on mobile phones. *Proceedings of CHI '00*, pp. 9-16.