Proceedings of the XX Annual International
Occupational Ergonomics and Safety Conference
Chicago, Illinois, USA
12-13 June 2008

DESIGNING A HYBRID LAYOUT FOR A FIVE-KEY TEXT ENTRY TECHNIQUE

Barbara Millet[1,2], Shihab Asfour[1], and James R. Lewis[3]

[1]Department of Industrial Engineering
University of Miami
1251 Memorial Drive
268 McArthur Engineering Bldg.
Coral Gables, FL 33146
Corresponding author's e-mail: b.millet@umiami.edu

[2] Motorola Inc.
8000 W Sunrise Blvd
Plantation, FL 33322

[3]IBM Software Group
8051 Congress Ave, Suite 2227
Boca Raton, FL 33487

Abstract: Five-key text entry methods are useful for limited text entry on small devices. They use four directional keys to move a selector over an on-screen keyboard and an Enter key for selection. Although other researchers have described five-key character layouts using alphabetical order and predictive layouts based on digraph frequencies, there is considerable latitude in designing the rest of a comprehensive on-screen keyboard. Furthermore, it might be possible to capitalize on the relative strengths of the alphabetic and predictive layouts by combining them in a hybrid layout. In this paper, we describe the design factors considered and tools used to develop three five-key text-entry techniques: alphabetic, predictive, and hybrid.

1. INTRODUCTION

There is a strong demand for mobile computing devices such as two-way pagers, PDAs, two-way radios, and cellular phones. Indeed, the mobile domain is overwhelmed by consumer demands for micro devices that offer full functionality, including some tasks that require text entry. By definition, micro devices have limited physical space for text input and output, and there is still a need for efficient text entry methods that can work within these constraints.

To support text entry on smaller devices, some researchers have investigated text entry using only 5 keys. Five-key text-entry techniques rely on four physical directional keys to move a selector over an on-screen keyboard and a fifth key for selection. Although clearly less efficient than a full-sized standard keyboard, five-key methods require relatively little space and are easy to learn (Wobbrock, Myers, & Rothrock, 2006).

Several researchers have conducted studies of five-key text-entry techniques using various keyboard layouts, such as alphabetic, QWERTY, and predictive. Bellman and Mackenzie (1998) proposed a hybrid layout, which combines fixed and predictive layouts, but did not provide any analyses of its efficacy. Such a combination could, however, prove advantageous for extremely small user interfaces. Thus, the focus of this paper is a "hybrid" layout (one that combines a full fixed and limited predictive character set), with comparison to a fixed and a predictive keyboard designed with similar constraints. The fixed keyboard is alphabetical and the predictive keyboard uses Bellman and Mackenzie's (1998) Fluctuating Optimal Character Layout (FOCL) strategy. The FOCL strategy relies on letter-pair (digraph) probabilities to dynamically rearrange letters (based on the most recently selected letter) for the purpose of minimizing selector movement.

In this paper we describe the design space and tools used to develop the three keyboard layouts. Specifically, we used keystrokes per character (KSPC) to quantify the efficiency of the proposed layouts. The overall goal of this effort is to improve text entry speeds on micro-portable devices, using only five keys for text entry.

ISBN # 97809654506-4-5

2. KSPC- AN ANALYSIS TOOL

When designing a new text entry technique, it is important to measure the method's efficiency. Keystrokes per character (KSPC) is a well-known model-based method for quantifying the efficiency of a text entry technique, using the average number of keystrokes needed to produce each character with a specific text entry technique (Mackenzie, 2002a; Wobbrock, 2007). Calculating KSPC requires the use of a language model and keystroke data. For this work, we used a "common" English language model (Mayzner & Tresselt, 1965; Soukereff & Mackenzie, 1995).

In 1965, Mayzner and Tresselt documented a table giving the 26 x 26 letter-pair frequencies in common English. Their work included the sampling of 200,000 words from various sources. However, their work did not include the space character. Because the space is the most common character (and e-space the most common digraph) in text entry, Soukereff and Mackenzie (1995) extended this table to include the space. Their table contains 729 (27 x 27) entries and the associated frequencies for each pair of characters (the number of times the second letter occurred immediately following the first, converted to a probability by dividing the frequency by 107,199). For example, the five most frequent diagrams (using "_" for the space) and their probabilities are "e_" (4.57%), "_t" (3.65%), "th" (3.52%), "he" (2.94%) and "d_" (2.45%).

Other researchers have constructed digraph tables using different sources and sampling more words than Mayzner and Tresselt (1965). For example, Zhai, Hunter, and Smith (2002) constructed two letter-pair tables, one based on on-line chat logs and the other based on a corpus of several newspapers. They noted that the differences between their tables and that of Mayzner and Tresselt were minute, and consequently decided, for the promotion of consistency in the literature, to use the Mayzner and Tresselt table in their continuing work (Zhai, Kristensson, & Smith, 2005) – a decision that we respect.

2.1. Calculating KSPC for Five-Key Text Entry

To compute an estimate of the average KSPC for any five-key design given a table of the letter pair probabilities and a table detailing the number of keystrokes needed to navigate from any character to the next, use:

$$\overline{KSPC} = \sum_{i \in N} \sum_{j \in N} (p_{ij} \times d_{ij}) \qquad (1)$$

where i is the first character, j is the second character of the digram, p_{ij} is the probability of occurrence for the letter pair, d_{ij} is the smallest number of keystrokes needed to get from i to j (plus one for the keystroke required to make the selection), and N is the character set. The use of KSPC as a measure of efficiency assumes expert behavior (use of the shortest path without errors). Limitations of KSPC are that it measures only a single aspect of performance and does not consider the layout of numbers, punctuations and other special characters.

3. DESIGN GOALS

The motivation for this research was to design and evaluate text input strategies suited to small, input-limited, mobile devices, satisfying the following objectives for the purpose of optimizing for minimal cursor movement and maximum usability:

1. The character set must appear on an on-screen keyboard and must not occupy more than two lines on the display. This model is appropriate for displays that are more wide than tall, such as those found on pagers, two-way radios and handheld gaming consoles.
2. Support one-handed operability.
3. Use only four keys to navigate (Up, Down, Left, and Right) through the character set and one for selection.
4. Reduce keystrokes as much as possible.
5. Provide an extended character set that, at minimum, includes the following:
    a. Numbers: 0-9
    b. Punctuations: period (.), comma (,), question mark (?), exclamation mark (!), single quote ('), double quote ("), and the colon (:)
    c. Symbols: underscore (_), dash (-), forward slash (/), and the at sign (@)

6. Include a subset of edit and modifier keys, such as Space, Delete, Shift, and Enter.
7. Develop a hybrid layout; combining fixed alphabetic and fluctuating predictive keys. To evaluate performance gains in using a hybrid keyboard, it is also necessary to develop a fixed (alphabetical) and a predictive (FOCL) keyboard.

4. DESIGNING FIVE-KEY TECHNIQUES

There are numerous factors that influence the design of on-screen keyboards. Some of these factors tend to interact, so it is important to consider the associated tradeoffs.

4.1. The Character Set

The characters occupy only two rows on the display. Any of the three keyboards display, at minimum, 46 characters (26 letters, 10 numbers, 6 punctuations, and 4 special characters). Numbers, punctuations, and symbols are in logical groups. The top row is mostly letters and the bottom row mostly the extended character set.

4.2. Fixed, Predicative and Hybrid Keyboards

Because a QWERTY layout would require three rows of letters, the fixed keyboard uses an alphabetic layout. The main advantage of a fixed keyboard is that users can easily become familiar with the layout, reducing visual search time to zero for the expert user. Unfortunately, this layout requires a greater number of keystrokes for entering text.

The predictive keyboard uses Bellman and Mackenzie's (1998) Fluctuating Optimal Character Layout (FOCL) strategy. These researchers proposed a dynamic reorganization of the letters to minimize the distance from the starting point based on probabilistic modeling of the next most likely character. Their FOCL keyboard used three-rows to display the character set and included a fixed space positioned to the left of the letters. User tests comparing the three-row version of FOCL with a QWERTY layout showed no improvement relative to QWERTY.

Mackenzie (2002b) revisited the FOCL strategy, examining text throughput for six methods of three-key text entry (Left, Right, and Select). The KSPC of the methods varied from 10.66 to 4.23. He evaluated four different alphabetic keyboards and two FOCL keyboards, and further investigated the two most promising layouts, one alphabetic and one FOCL. Both of these keyboards put the space at the leftmost key position. Again, user tests indicated no difference in text entry rates for these methods. Participants preferred the alphabetic layout.

The main advantage of the FOCL strategy is that it significantly reduces KSPC. However, a fluctuating keyboard is more difficult to learn because the user is exposed to 27 different layouts, which greatly increases the demand of visual search. To gain the advantages associated with alphabetically ordered and pure FOCL keyboards, we propose a hybrid keyboard equipped with fixed and a limited set of dynamic keys. The expectation is that the hybrid keyboard will allow users to take advantage of the reduced KSPC in the dynamic portion of the layout while allowing easy access to a fixed alphabetic layout when the desired next character is not one of the most likely next characters.

4.3. Wraparound Cursor

With wraparound enabled, the cursor will not stop at the left or right end of a row (as it would with wraparound disabled), but will wrap around to the other side.

4.4. Cursor Mode

Cursor modes can be persistent or snap-to-home. A persistent cursor remains at the selected position on character selection. A snap-to-home cursor jumps back to the assigned key (the cursor home position) immediately after each entry. Therefore, evaluating snap-to-home cursor mode also requires assessing the best home position.

4.5. Cursor Home Position

The cursor home position is the location assigned to the cursor in a snap-to-home mode. As in Bellman and Mackenzie (1998), we evaluated a subset of possible cursor home positions to investigate this variable. In English, the probability of the space character is 18% (Soukoreff & Mackenzie, 1995). Given its prominence in typing, it is common to grant the space special treatment (Bellman & Mackenzie, 1998). For this reason and because previous studies have found the approach promising (Mackenzie, 2002b; Wobbrock, Myers, & Rothrock, 2006), we assigned the space to the cursor home position for all layouts.

4.6. Space Location

In general, we placed the space key in a location that would enhance natural left-to-right scanning of the letters in the layout. The specific location differed among the layouts.

4.7. Delete key

Errors and error handling are important dimensions to consider when designing keyboards. As a consequence, we have augmented our designs with a backspace. The backspace appears above the keyboard (in line with the user output). In doing so, adding the backspace did not increase KSPC for any layout. Furthermore, it is possible to acquire the backspace location in no more than three keystrokes.

5. KEYBOARDS

5.1. Fixed Keyboard

As indicated in section 4.2, the fixed keyboard uses an alphabetic layout. To evaluate the multiple designs for an alphabetic keyboard, we calculated KSPC for all possible layouts (a total of 16) given the design factors explored in this work. As detailed in Table 1, the keystrokes per character (KSPC) for the alphabetic keyboard varied from 13.43 to 6.53. Figures 1 and 2 depict main effects and interactions for various key design variables.

Table 1. Sixteen Designs for Alphabetic Five-Key Text Entry

Design	Cursor Wraparound	Cursor Mode	Cursor Home Position	Space Location	KSPC
1	Yes	Snap-to-home	Space	Center	6.53
2	No	Snap-to-home	Space	Center	6.53
3	Yes	Snap-to-home	Space	Top Leftmost	6.86
4	Yes	Snap-to-home	Space	Top Right	6.86
5	Yes	Snap-to-home	Space	Bottom Leftmost	7.23
6	Yes	Persistent	Not Applicable	Top Leftmost	7.89
7	Yes	Persistent	Not Applicable	Top Right	7.89
8	Yes	Persistent	Not Applicable	Bottom Leftmost	7.96
9	Yes	Persistent	Not Applicable	Center	8.16
10	No	Persistent	Not Applicable	Center	9.22
11	No	Snap-to-home	Space	Bottom Leftmost	10.54
12	No	Snap-to-home	Space	Top Leftmost	10.54
13	No	Persistent	Not Applicable	Top Leftmost	10.88
14	No	Persistent	Not Applicable	Bottom Leftmost	10.88
15	No	Persistent	Not Applicable	Top Right	11.56
16	No	Snap-to-home	Space	Top Right	13.43

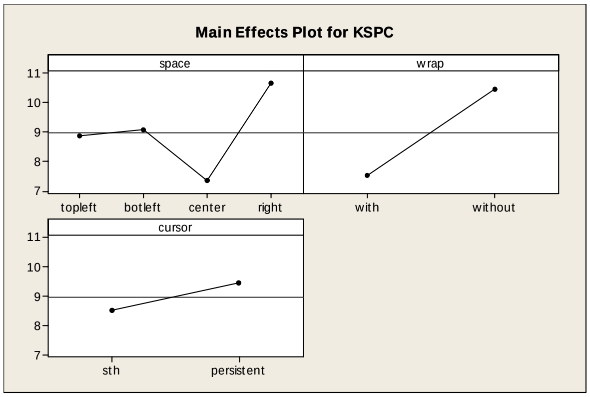

Figure 1. Main Effects Plot for KSPC

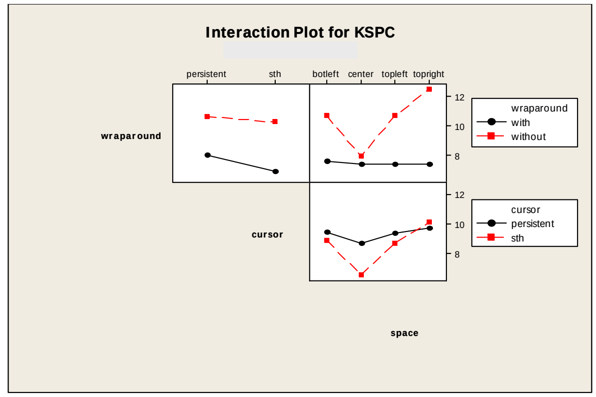

Figure 2. Interaction Plots for KSPC

The main effect analyses showed that enabling wraparound dramatically reduces KSPC, placing the space key in the center of the layout significantly reduces KSPC, and using a snap-to-home cursor slightly reduces KSPC. Analysis with Minitab's "Response Optimizer" tool confirmed that the optimal alphabetic keyboard had wraparound, a centered space, and a snap-to-home cursor. The only interaction was between wraparound and space location. As is clear from Figure 2, there is very high variance in KSPC without wraparound. Enabling wraparound reduces KSPC sensitivity to the other actors, and always results in substantially lower KSPC.

Thus, we designed the fixed keyboard to employ wraparound and a snap-to-home cursor. Although a center space location would have KSPC equal to 6.53, we placed the space in the top leftmost position to enhance natural left-to-right scanning of the characters, which increased KSPC only slightly to 6.86 (a 5% increase – see Figure 3 – note that the assignments for numbers, punctuation, and symbols are not yet final).

Figure 3. Fixed Keyboard

5.2. Predictive Keyboard

The predictive keyboard uses Bellman and Mackenzie's (1998) FOCL strategy, modeled to map closely to the three-key "FOCL Level 1" keyboard proposed by Mackenzie (2002b). The FOCL Level 1 layout places the space at the leftmost position and uses a snap-to-home cursor. Our design (see Figure 4) uses five-key navigation and a wraparound cursor, with an estimated KSPC of 4.52. This indicates that English text produced with the predictive keyboard requires 34% fewer keystrokes per character than the fixed keyboard. The layout shown is the initial layout (also the layout that would appear after entering a space). Note that the assignments for the numbers, punctuation, and symbols are not yet final.

Figure 4. Predictive Keyboard

5.3. Hybrid Keyboard

According to Bellman and Mackenzie (1998), the FOCL benefit of reducing keystrokes only occurs when the target letter is close to the cursor. In the proposed hybrid keyboard, the layout includes a fixed alphabetic row and a small number of fluctuating positions centered within the second row (see Figure 5 – note that the assignments for numbers, punctuation, and symbols are not yet final).

To determine the appropriate number, *n*, of fluctuating positions, we calculated the cumulative sum of probabilities associated with *n* positions, where *n* ranged from 1 to 27. Our goal was that the probability of the target letter appearing in a fluctuating position be greater than 0.80. To satisfy this objective, there should be at least seven fluctuating positions (when $n = 7$, the probability of the target letter appearing in one of the seven fluctuating positions is 0.81).

Consistent with the results of fixed keyboard analysis, the hybrid layout uses wraparound and a snap-to-home cursor. We expect people to use the fluctuating component more often than the fixed portion of the layout, so we assigned the space to a position to the left of the fluctuating characters. As in the fixed keyboard, this will enhance natural left-to-right scanning of the characters in the fluctuating part of the keyboard. For this keyboard, the estimated KSPC is 3.87, requiring 14% fewer keystrokes per character in comparison to the predictive keyboard and about 44% fewer compared to alphabetic.

Figure 5. Hybrid Keyboard

6. CONCLUSIONS

We have investigated three techniques for five-key text on entry on mobile devices. The KSPC for the methods were 6.86 for alphabetic, 4.52 for predictive, and 3.87 for hybrid, suggesting that the hybrid layout has the potential to be the best of the three keyboards. KSPC is a useful tool that allows the characterization and comparison of text entry techniques before development and evaluation. However, using KSPC is not a substitute for a user test of text entry using high fidelity prototypes. Rather, it is a screening tool for the early identification of weak text entry strategies, most appropriately used before the commitment of development and testing resources.

In the future, we plan explore typematic keying and any advantages this may bring (Mackenzie, 2002b; Wobbrock, Myers, & Rothrock, 2006). Additionally, we plan to conduct user trials of the prototype five-key systems, supporting full text entry (including numbers, symbols, and punctuations) to investigate novice user performance. We also plan to conduct a longitudinal study of these five-key text entry methods.

7. REFERENCES

Bellman, T., & MacKenzie, I. S. (1998). A probabilistic character layout strategy for mobile text entry. In *Proceedings of Graphics Interface '98* (pp. 168-176). Vancouver, Canada: Canadian Information Processing Society.

MacKenzie, I. S. (2002a). KSPC (keystrokes per character) as a characteristic of text entry techniques. In the *Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices* (pp. 195-210). Heidelberg, Germany: Springer-Verlag.

MacKenzie, I. S. (2002b). Mobile text entry using three keys. In *Proceedings of NordiCHI 2002* (pp. 27-34). Arhus, Denmark: Association for Computing Machinery.

Mayzner, M.S., & Tresselt, M.E. (1965). Tables of single-letter and digram frequency counts for various word-length and letter-position combinations, *Psychonomic Monograph Supplements, 1(2)*, 13-32.

Soukoreff, R. W., & MacKenzie, I. S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. *Behaviour & Information Technology*, *14*, 370-379.

Wobbrock, J. O., Myers, B. A., & Rothrock, B. (2006). Few-key text entry revisited: Mnemonic gestures on four keys. In *Proceedings of the ACM Conference on Human Factors in Computing Systems '06* (pp. 489-492). New York, New York: ACM Press.

Wobbrock, J.O. (2007). Measures of text entry performance. In I. S. Mackenzie, & K. Tanaka-Ishii (Eds.), *Text entry systems: Mobility, accessibility, universality* (pp. 47-74). San Francisco, CA: Morgan Kaufmann.

Zhai, S., Hunter, M., & Smith, B. A. (2002). Performance optimization of virtual keyboards. *Human-Computer Interaction*, *17*, 89-129.

Zhai, S., Kristensson, P.-O., & Smith, B. A. (2005). In search of effective text input interfaces for off the desktop computing. *Interacting with Computers*, *17*, 229-250.