A Probabilistic Character Layout Strategy for Mobile Text Entry

Tom Bellman
Department of Computer Science
University of Toronto
Toronto, ON M5S 1A1 Canada

Phone: 416-535 6620 Fax: 416-978-5184
e-mail: bellman@dgp.utoronto.ca

I. Scott MacKenzie
Dept. of Computing & Information Science
University of Guelph
Guelph, ON N1G 2W1 Canada

Phone: 519-824-4120 x8268 Fax: 519-837-0323
e-mail: smackenzie@acm.org

Abstract

We present a text entry strategy designed for small, input-limited devices in mobile situations. In an existing text entry interaction style, arrow keys move a cursor around the character set, displayed on a 3 or 4 line liquid crystal display, and a select key outputs the highlighted character to the display. The Fluctuating Optimal Character Layout (FOCL) strategy aims to improve entry rates with this interaction technique. After each character *c* entered, the layout is rearranged so that the most likely next characters are closer to the cursor. Each new layout is optimal with respect to *c*, given digram (letter-pair) probabilities in common English. FOCL significantly reduces *kspc*--the average number of keystrokes per character--, a non-empirical measure that partly accounts for text entry speed. However, it also requires the user to locate each character in an unfamiliar layout, thus adding visual search time to the task. In a longitudinal experiment comparing the fluctuating layout approach to a fixed QWERTY layout, we found no significant difference in entry speeds. We discuss our design rationale and various modifications to the design that may yield a performance improvement.

Nous présentons une méthode d'inscription de texte creé pour des appareils à entrée limitée dans des situations mobiles. Dans un style d'interaction existant, les flèches du clavier déplacent le pointeur entre les lettres, et une touche de selection permet d'éclairer la lettre qui se trouve sous le pointeur. Le but de notre méthode, stratégie de disposition optimal, variable des lettres (FOCL), est d'améliorer la vitesse d'inscription de caractères avec cette technique d'interaction. À chaque fois qu'une lettre est sélectionnée la disposition des lettres change de façon à ce que les plus vraisemblables à suivre celle choisie se retrouvent le plus près que possible du pointeur. Chaque nouvelle disposition des lettres est optimisée en fonction des probabilités d'occurrence des digraphes (double lettre) en Anglais commun. La méthode FOCL réduit le *kspc*--la moyenne de nombre de touches pour chaque lettre inscrite--une mesure non-empirique qui rend compte partialement de la vitesse d'inscription. Mais elle éxige de l'utilisateur de trouver chaque lettre dans un disposition peu familier, et cree donc une perte de temps pour faire la recherche visuelle. Une éxperience longitudinale comparant notre méthode avec un disposition QWERTY fixe, n'a pas montrée de différence significative entre les vitesses d'inscription de caractères. Nous discutons notre raisonnement de dessein ainsi que les modifications qui pourraient produire une amélioration en performance.

Keywords: Text entry, mobile systems, probabilistic character layout, digram frequencies, design rationale

Introduction

This research focuses on the problem of text entry on portable communications devices, such as pagers and cellular phones. Such devices have been able to *receive* text messages for some time now, but with the growing ubiquity of email, products with text message-*sending* capability are beginning to appear.

The requirement of mobility imposes a unique set of restrictions on the design of a text entry strategy. Typically, there is space on the device for only a small number of buttons. A full keyboard, even a miniature one, is not feasible, so that the character set must appear on the device's liquid crystal display. The LCD has a small number of lines, which must accommodate both the character set and the output, possibly via scrolling. The ability to enter text with one hand is desirable, given that users may not have both hands free. In this paper we will use the term 'mobile text entry' to refer to text entry limited by this set of restrictions. The style of target device for this research is sketched in

Figure 1. Our research question is the following: Given these restrictions, can we design techniques for mobile text entry that yield reasonable entry speeds?

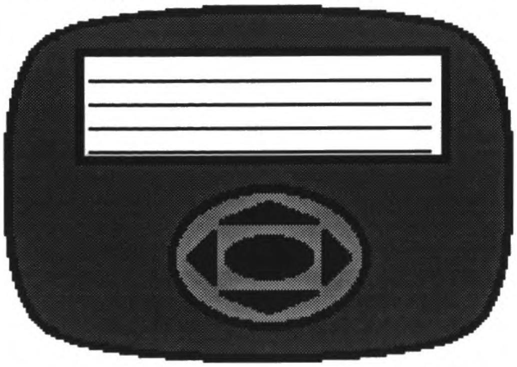


Figure 1: Portable communications device with small LCD and a few of buttons for input

Though the solution may ultimately lie in voice recognition approaches to the problem, such systems are not yet robust enough for mobile situations, where external noise cannot be controlled. As well, mobile products usually have very stringent limitations in terms of power requirement, memory size, processor speed, etc. Pen-based solutions are practical only if one-handed use is not a requirement and the device can accommodate a sufficiently large writing surface.

Interactions Styles

One can imagine numerous interaction styles for mobile text entry, some borrowing from existing techniques in other domains. We will briefly outline three that we are aware of which meet the requirements of mobility.

The telephone keypad method

In this well-known if little used method, each of the keys 2 through 9 reference 3 letters of the alphabet. The 0, 1, # and * keys are available for other functions. Entering a letter involves selecting the key it appears on as well as specifying which of the three letters is intended, which can be accomplished in several different ways. Thus, the system may interpret input from the same key differently at different times.

The date stamp method

The method is thus named because, as with a date stamp, a desired character must be made 'visible' by rotating a wheel containing the entire character set. As a text entry method, cursor keys allow navigation of the output space, and increment and decrement keys cycle sequentially through the character set at the cursor position. Players of video arcade games are familiar with this technique, which is used to input one's initials into a list of high scorers. Unlike a physical date stamp, selection of a character is performed implicitly by simply moving the cursor to another location. In effect, the cursor position is a moveable editing window of one character. This technique is the standard for entering text in many electronic musical instruments as well.

The soft keyboard method

In this method the display is functionally split into an input and output section. The input portion of the display either shows the full character set or allows different parts of it to be viewed by scrolling, while the output appears in ticker tape fashion along a line or two of the display. Arrow keys like those in Figure 1 move a cursor around the character set, and characters are entered by an explicit press of a select key. We know of one product, the AccessLink™ 2-way pager by Wireless Access, that uses this method for text entry.

The problem with these three techniques is that an excessive number of keystrokes is required for text entry. This research aims to increase mobile text entry rates by devising ways to reduce keystrokes. Though applicable to any number of text entry techniques, our efforts at reducing keystrokes have been directed at the soft keyboard method.

Optimizing layouts

Many researchers have devised "optimal" layouts for particular kinds of text entry, e.g. [7, 13]. These approaches, knowingly or not, take advantage of a language's redundancy--the unequal occurence of characters, and strings of characters, that is a feature of all languages [9]. Optimal layouts apply first order redundancy (single letter frequencies), second order redundancy (letter pair or digram frequencies), or a combination of the two by positioning the most common letters and letter pairs so they can be entered as efficiently as possible given the interaction technique. (Higher orders of redundancy carry the tradeoff of exponentially greater complexity.) Efficiency has different design implications within different interaction styles; for example, in stylus tapping the goal would be proximity of frequently adjacent letters, while in two-handed typing it would be placement of each of the two letters of a common digram under a different hand.

Fluctuating Optimal Character Layout (FOCL)

The method we are proposing adds a variable aspect to the optimal layout approach. Rather than build one optimal layout for all situations, we propose a layout for every distinct situation, in this case a layout for each character. We call this approach Fluctuating Optimal Character Layout (FOCL). With each character *c* entered, a new layout is produced based on digram frequencies in common English [8], so that the more likely a letter is to follow *c*, the closer it will be to the cursor in the new layout.

Frequencies of digrams, and of the more generalized n-grams (strings of n characters), are found in abundance in books on code breaking, or cryptanalysis, for example [2]. They are also of interest to researchers in the psychology of reading and learning (e.g. [11]), as well as to linguists (e.g. [8]). For the purposes of a text entry system, it is necessary to account for the space character, which is not typically included in the frequency tables of the aforementioned disciplines. However, by inferring digrams containing spaces from digram data that also recorded the position of digrams within words [8], the researchers in [10] produced digram frequency data that includes the space character. This makes it possible to generate fluctuating optimal layouts.

The rows of letters in Figure 1 are orderings of the alphabet in order of likelihood to follow the bold-faced letter to the left of the list. These orderings were derived from digram frequencies in [10]. So, for example, the most likely letter to follow A is R, then N, then T, and so on. The space character, indicated by the underscore character in Figure 2, has its own row, representing the order of likelihood to begin a word. The space character has been removed from the orderings themselves because it receives special treatment in the interface, similar to the prominence afforded the space bar on the standard keyboard. This will be described later in the paper.

A	NRTSLDIYCVMKBGPUFWXZHJAEOQ
B	EUOALRIJSBYTMVWCDFGHKNPQXZ
C	HAOEKTLRIUCYSNQBDFGJMPVWXZ
D	EIOASRUYDLGVFTNWBJMCHKPQXZ
E	RNDASELTYMCVIWFPXGOKHQBUZJ
F	OIREATFLUSYMBCDGHJKNPQVWXZ
G	EHORIAULSNGTYBMWCDFJKPQVXZ
H	EAIOTRUYSWLPHMBCDFGJKNQVXZ
I	NTSLCRDMGEVOKFPAZXBUQIHJWY
J	UOEIABCDFGHJKLMNPQRSTVWXYZ
K	EINSLYOUAMPBCDFGHJKQRTVWXZ
L	ELIOADYSUTFKPMWVRGCBNHJQXZ
M	EAOIUPSBMYFLNTCDGHJKQRVWXZ
N	DGETOSIYACLKNUJFHQVBRWXMPZ
O	URNMTWOLSPVKDIFBACYEGZJXHQ
P	ELOARIUPSTHYWCFBDGJKMNQVXZ
Q	UABCDEFGHIJKLMNOPQRSTVWXYZ
R	EOSIATYDNULRMGKCVHFBPWJQXZ
S	TEHIOAUSPKLCYWMNBQGDFJRVXZ
T	HEOAISRTUYWLCBGNDFMZJKPQVX
U	TRSNLGCMEIBPDAFYOZQXHJKUVW
V	EIOAYUBCDFGHJKLMNPQRSTVWXZ
W	AHIEONRSLDTBUKCFGJMPQVWXYZ
X	TPAICEUOBDFGHJKLMNQRSVWXYZ
Y	OESIPABLTWGHMUCDFJKNQRVXYZ
Z	EZLASYIOBCDFGHJKMNPQRTUVWX
_	TAWSHFBMCOLPDRNGYEIKUVJQZX

Figure 2: Alphabet orderings in order of likelihood to follow each letter or the space character

Mean Keystrokes per Character (*kspc*)

We can compute an estimate of the mean number of keystrokes per character (*kspc*) for a particular layout with (1) a table of the number of keystrokes required to get from any character to any other character and (2) the probability of all possible digrams in the language. The formula is as follows:

$$\overline{kspc} = \sum_{\alpha \in \Phi} \sum_{\beta \in \Phi} \left(\rho_{\alpha\beta}\right) \bullet \left(\delta_{\alpha\beta}\right)$$

where α is the first character of a digram, β is the second character of a digram, $\rho_{\alpha\beta}$ is the probability of the digram $\alpha\beta$, $\delta_{\alpha\beta}$ is the smallest number of keystrokes to get from α to β, and Φ is the character set. With the interaction technique in question we need to add 1 to each value of $\delta_{\alpha\beta}$ to account for the explicit selection keystroke.

This measure tells us, on average, how many keystrokes are required to travel from one character to the next, and select that character. It assumes optimal user behaviour, meaning that a shortest path is taken from one character to the next. To illustrate the dramatic effect on keystrokes of applying the FOCL strategy, imagine the 26 letters plus space character arranged on a single line of the display. Figure 3 compares *kspc* values for a fixed alphabetic arrangement and a fluctuating FOCL arrangement. In this case, the *kspc* for FOCL is 2.15 keystrokes or 35% less than that of the alphabetic layout. Later we will look at *kspc* values for more complex layouts, as well as factors that further reduce *kspc*.

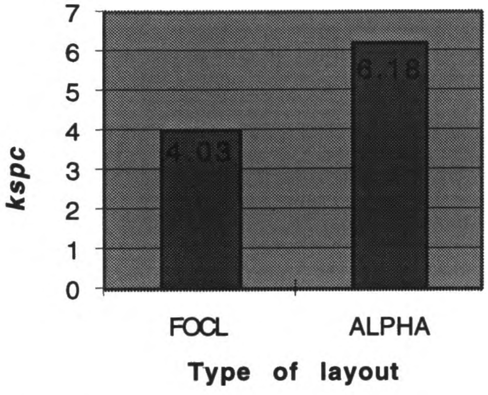


Figure 3: Comparison of kspc values for single line character set displays using FOCL (fluctuating) and alphabetic (fixed) layouts

It should be noted that this research is at the moment restricted to the 26 letters of the English alphabet and the space character. Punctuation and other symbols, as well as the digits are excluded from our analysis, as is the distinction of upper and lower case letters.

Towards higher text entry speeds

It is hoped that by reducing keystrokes, higher text entry speeds will be achievable. If no other factors affected speed, then this would indeed be the case, given the lower *kspc* values. However, rearranging the layout after each character typed also adds a visual search (VS) time, as the user must locate the next desired letter in an unfamiliar layout. FOCL will only outperform a fixed layout approach if the savings gained by reduced keystrokes outweigh the cost of VS, or if VS can be reduced over time by user familiarity with the various layouts. This suggests the following two design goals for FOCL-based text entry:

1. Make design choices that afford familiarity with the layouts
2. Reduce mean keystrokes per character as much as possible

Each of these is discussed below.

Affording Familiarity with Layouts

The great advantage of a fixed layout over a fluctuating approach is that users can achieve total familiarity with the layout. VS time effectively drops to zero. A fluctuating layout strategy can only hope to reduce, not eliminate, VS. For a character set of size *n*, the number of possible layouts under FOCL is in the order of n^2 because each character might, theoretically, be dropped at some time in every possible position, although in reality some layouts will never occur. There is no chance of users ever becoming totally familiar with a system having as many as 27^2 different layouts. However, by snapping the cursor back to a fixed home position after each character entry, the number of possible layouts is reduced from n^2 to *n*. We believe it is not unreasonable to expect users to achieve some familiarity with 27 layouts over time.

Another factor with the potential to increase FOCL text entry speeds is letter pattern chunking or unitization [12]. Words such as THE, YOU, and THERE are made up of the most probable digrams in common English. Entering these words does not require the use of any arrow keys as the next letter, being the most probable in that situation, appears at the cursor position. It is hoped that users will enter such strings as a unit rather than as separate letters, and that over time their repertoire of such patterns will grow larger and more complex.

Design factors affecting *kspc*

As Figure 3 illustrates, merely applying digram probabilities reduces keystrokes. The following factors can further affect *kspc* within the FOCL approach:

- Number of rows in the character set layout
- Location of cursor home position
- Treatment of the space character

The following discussion examines each of these independently, and then identifies some of the tradeoffs that arise from their interdependencies.

Number of Rows

Kspc is inversely related to the number of rows, until there are more columns than rows, at which point we have essentially rotated the layout ninety degrees. Intuitively, increasing the number of rows decreases the longest distance between any two characters in the layout, equivalent to the graph theory concept of *diameter* of a graph or network. Since kspc is an average of distances, it makes sense that a layout of lesser diameter would have a lower kspc.

Cursor Home Position (CHP)

We compared the effect on *kspc* of the following three categories of cursor home position.

- at a corner (Figure 4a)
- at an edge, but not at a corner (Figure 4b)
- in the centre (Figure 4c)

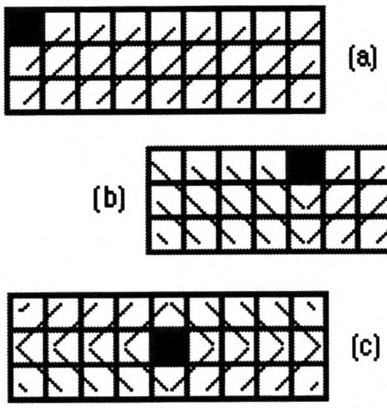


Figure 4: Alternative cursor home positions and associated optimal visual search patterns as dictated by the principle of positioning most probable characters closest to the cursor

A centred cursor has the advantage of having more keys adjacent to it, and thus more keys at shorter distances from it. As a result, centred cursor layouts have lower *kspc* values than identical layouts in which the cursor is either in a corner or at an edge.

Figure 5 summarizes the combined effect of number of rows and cursor home position on *kspc*. *Kspc* values for QWERTY and alphabetic 3-row layouts are included for comparison.

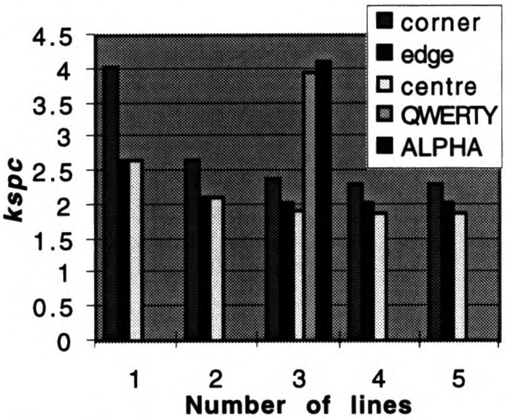


Figure 5: Summary table of combined effect of number of rows and cursor home position on *kspc*

Treatment of the Space Character

The space character receives special treatment in the physical keyboard due to its overwhelming prominence in typing. In English, the space character occurs with a probability of .185 [10]. The space bar's size and position under the thumbs insures that accessing it never requires a reach or stretch of the hand. This was, in our opinion, one aspect of the physical keyboard that ought to be emulated in mobile text entry. The solution that came to mind was to place the space character adjacent to the cursor home position, so that it would always be a single keystroke away after a character selection (see Figure 6, below)

Another way of benefitting from the high frequency of occurrence of the space character is to allow for its implicit rather than explicit selection. Using the layout in Figure 6 to demonstrate, one way to achieve this would be via a single left arrow stroke that would momentarily highlight the space, then snap the cursor back to its home position. Such a feature would reduce by half the number of keystrokes associated with space character selection.

Tradeoffs

Cursor home position vs. optimal search strategy

Although superior with respect to keystrokes, the centred cursor may not be recommended because it requires the user to search in a concentric circular pattern (Figure 4c), which is highly unnatural. Visual scanning is most natural from left to right [3]. Edge- (Figure 4b) and corner-positioned cursors (Figure 4a) require concentric searching but to a lesser degree. The principle of placing characters as close to the cursor as possible dictates that the most natural search pattern for people will never be optimal for FOCL use. The search pattern in Figure 4a is the least unnatural of the three, so that under this criterion a corner-positioned cursor is the best choice.

Cursor home position vs. movement decision time

Another hypothesized disadvantage of a centred cursor is that it requires equal use of all four arrow keys, thereby increasing the decision time between locating a character and beginning to move toward it. In general, decision time is proportional to the number of options there are to choose from (Hick, 1953; Hyman, 1953). In error-free user behaviour a corner-positioned cursor requires at most two arrow keys to navigate the character set, while an edge-positioned cursor requires at most three. Under this criterion a corner home position is the best choice.

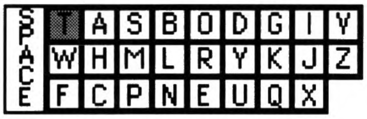
Location of space character and cursor home position

A centre cursor home position is cut off, as it were, from the sides of the layout. Therefore, at least two--depending on the shape of the layout--keystrokes would be required to enter a space character under the treatment proposed. Positioning the space within the layout, like all other characters, is not a good solution unless implicit space character selection is abandoned. Without doing so, there would be numerous errors in cases where the path from the cursor to a letter passed through the space position. Another possible treatment is to always put the space character at the cursor home position, but this would forfeit the possible benefits of chunking. A centre cursor home position appears to conflict with the desirable space character treatment features.

Implicit space selection and cursor home position

A potential problem with implicit selection of the space character is that it may confuse the function of the arrow key used to access the space bar, possibly leading to a higher error rate. It also introduces an inconsistency into the interface -- now one character is selected in a different way than all the others. This issue further recommends either a corner or edge cursor home position, each of which, in error-free usage, have at least one arrow key that is unused. One of these idle keys may be treated as a dedicated space bar. Under the reasonable assumption that users approach error-free usage over time, the confusion and inconsistency described above would be greatly reduced, or eliminated altogether, with either a corner or edge cursor home position.

Our prototype



SPACE	T	A	S	B	O	D	G	I	Y
	W	H	M	L	R	Y	K	J	Z
	F	C	P	N	E	U	Q	X	

Figure 6: Layout of experimental prototype

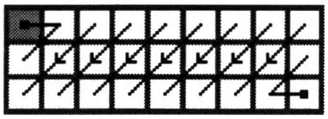


Figure 7: Sequence of positioning of letters from most to least probable

In order to explore this interaction technique and test its viability experimentally, we created a prototype in Macromedia Director. At this early phase of investigation, rather than modify a small device to our purposes we used a Macintosh keyboard and monitor for input and output. Figure 6 shows our design choices for the prototype, consisting of a 3-row layout, top-left corner cursor home position and space character to the left of the layout.

The particular layout of Figure 6 is an arrangement of the letters in the last line--corresponding to the space character--of Figure 2 according to the sequence of positioning in Figure 7. In a similar way, the remaining 26 layouts used in our prototype can be generated.

Design Choice Rationale

Number of lines

The choice of a three-row layout is based on a conservative estimate of four lines as the typical display height of portable communications devices. With one line reserved for output, this leaves three for the character set. Another factor in this choice was the decision to use the QWERTY layout, itself made up of three rows, as the comparison condition in an experiment. This eliminated a possible source of confounding.

Cursor home position

Were we to look solely at keystroke reduction, then a centre cursor would be the best choice. For instance, in Figure 5, a 3-row FOCL layout has a *kspc* value of 1.92 with a centre cursor, 2.04 with an edge cursor and 2.38 with a corner cursor. However, as previously discussed, the centre cursor does not afford a key that is effectively a space bar. In addition, it may require the least natural optimal visual search strategy. For these reasons, we rejected a centre cursor for our prototype. We chose a corner-positioned cursor over an edge-positioned one for the added advantage of 2-key, as opposed to 3-key, navigation that it affords under optimal user behaviour. Having chosen a corner cursor, the choice of top left corner followed naturally in that it corresponds to the starting position of the cursor in standard English typing.

Space bar location

The choice of a top left cursor home position offered two possibilities for space bar location: above or to the left of the layout. We chose the left option.

Method

In this experiment we evaluated FOCL and fixed layout approaches to text entry. The experiment was exploratory in nature, serving to identify key issues in the design of FOCL-based text entry. Although we anticipated the study would help to identify numerous design aspects in need of rethinking, we nevertheless proceeded with the hypothesis that our FOCL prototype would outperform the fixed layout interface, given a moderate amount of practice.

Subjects

A total of 11 subjects (6 male, 5 female), mainly University of Toronto students, participated in the experiment. Though it was not our intention, all subjects were right-handed. Three had touch typing speeds in excess of 60 wpm, 7 had speeds between 20 and 60 wpm, and 1 was a non-typist. Eight of 11 subjects were native English speakers. Subjects were paid for their participation on the condition they completed the study, which all did.

Apparatus

The experiment was conducted using a Power Macintosh computer with a 19" display and a standard Apple keyboard. For right-handed subjects, the 4, 5, 6 and 2 keys on the numeric keypad functioned as the left, down, middle and up arrow keys respectively, while the right arrow key functioned as the select key (Figure 8, right picture). For left-handed participants, the a, s, d and w keys served as arrow keys while the space bar served as the select key (Figure 8, left picture). These keys were chosen so that subjects could manipulate the cursor using the index, middle and fourth finger of a single hand, *and* have a selection key fall conveniently under the thumb. (This requirement ruled out using the keyboard's actual arrow keys, which are at the bottom of the keyboard and thus do not afford such a key).

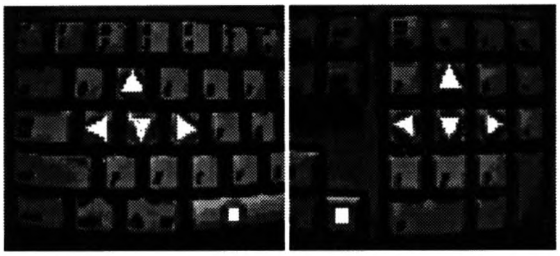


Figure 8: Keyboard preparation for experiment, for left and right-handed subjects

The software was written in Lingo, the programming language within Macromedia Director. The labelled arrow keys controlled a cursor on the screen, allowing users to navigate the character set. The display consisted of a large text field at the top of the screen where stimulus phrases appeared, an output field displaying the characters typed by the subject, and the character set.

We chose the QWERTY layout as the fixed comparison condition in the experiment. The behaviour and appearance of the fixed QWERTY layout differed from that of the FOCL layout in a number of ways.

- The cursor did not snap back to a home position in the QWERTY condition.
- In the QWERTY condition, the space character was below the character set, and could be anywhere from one to three keystrokes away from the cursor position after a character selection. In the FOCL condition, the space was to the left of the character set and was always a single left arrow stroke away after a character selection.
- The QWERTY layout had rows of 10, 9, and 7 characters, corresponding as closely as possible to its physical counterpart, whereas the FOCL layout had rows of 9, 9, and 8 characters. (Figure 6)

Procedure

At the first session, subjects were given a written walkthrough and asked to follow it systematically. The walkthrough was designed to familiarize subjects with the physical interface and with each of the two conditions.

To begin an actual session subjects were instructed by a screen message to press any key, launching the first condition. A phrase would appear at the top of the screen, which the subject would read and begin entering. The phrase remained on screen while it was being entered. Subjects had been instructed not to correct errors, and in fact it was not possible to delete characters. A soft beep sound notified subjects of an error. A highlighted character on the stimulus phrase served as a guide for resuming correctly after an error. At the completion of a phrase, a bell-like sound accompanied the change to a new phrase.

Design

We used a counterbalanced, within-subjects design. The independent variable was type of layout, either QWERTY or FOCL. Dependent variables were text entry speed (words per minute) and error rate.

Each subject attended 10 sessions over a period of 8 to 10 days. Sessions were spaced by no less than one hour and by no more than two days. Each session consisted of 15 minutes of typing phrases in one condition, followed by a short break, then 15 minutes in the other condition. The order of the two conditions alternated with each session. The actual time in each condition was not precisely 15 minutes, because the software would allow subjects to complete whatever phrase they were on before signaling that the condition was over. Though the effect of transitions from one condition to another was considered a possible confound, it was felt that alternating the order of conditions would control for this.

Subjects were instructed to proceed as quickly and as accurately as possible, and to try to keep their error rate to 2-5 errors per phrase. Phrases were selected randomly from a bank of approximately 500. The software kept track of which phrases each subject had seen already, so that no phrase would be presented to the same subject more than once. The phrases, gathered from various Internet sites storing quotations and cliches, varied in length from 7 to 30 words, with an average of 14.

Results

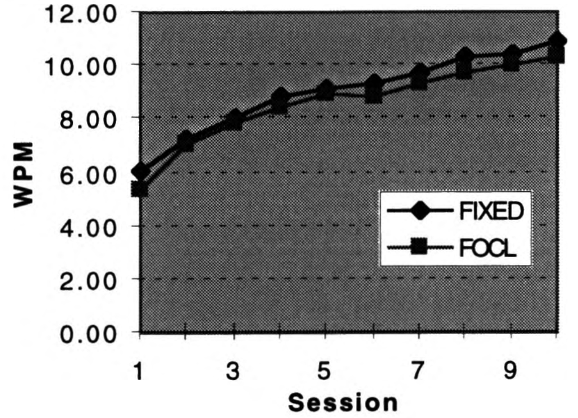


Figure 9: Overall mean words per minute by session

We found no significant difference in text entry speeds between the two conditions ($F_{1,10}$ = 1.39, p > .05). Figure 9 shows our mean text entry speed results. Furthermore, although subjects performed fewer errors in the FOCL condition, the difference in error rate was not found to be significant either ($F_{1,10}$ = 3.24, p > .05). However, there was a significant effect of time on the entry speed ($F_{9,90}$ = 105.3, p < .0001), as expected. As is apparent in Figure 9, speeds continued to improve throughout the study in both conditions.

Observations of subjects during the study, as well as interviews conducted with each of the participants at the conclusion of the experiment revealed a number of issues that will inform further investigation. Some of the more interesting observations are:

1. Though for the most part subjects understood how characters were positioned in the FOCL layout (Figure 7), none of them were able to scan optimally (Figure 4a), according to their own reports.
2. Subjects expressed frustration with certain aspects of FOCL's behaviour. These included:
   (a) the need to relocate a letter in the event of a double letter, rather than being able to select it twice.
   (b) the inability to take advantage of chance adjacencies in the character set that were identical to a letter combination in the word being typed.
   (c) the change in position of letters from one layout to the next within the same vicinity of the layout. This was felt to be a detriment to developing familiarity.
   (d) the perceived poor placement of the letter 'i' in the layout following a space character. (This, we believe, was due to the nature of the sample of English used to generate digram frequencies [10], which excluded words of less than three letters, thus disproportionately reducing the prominence of the letter i)
3. Only limited chunking occurred. While all subjects chunked words such as THE, YOU and THERE, there was not a wide variety of patterns beyond these few. Furthermore, subjects often failed to recognize patterns chunked as separate words in larger words, such as THE in OTHER.

Discussion

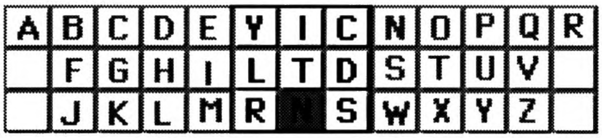
Although we did not achieve the desired crossover, we are encouraged by the fact that FOCL performed no worse than QWERTY, demonstrating its fundamental viability as a text entry method. With mean text entry speeds in the vicinity of 10 wpm, and the fastest subjects achieving speeds of 15 wpm, the technique approaches conventional handwriting speeds of 15-17 wpm [1]. As we did not observe a levelling off of speeds towards the end of the experiment, it is likely that even higher speeds could be achieved.

The inappropriateness of our character positioning pattern (Figure 7) has significant redesign implications. We need to consider positioning strategies that cater to the way in which humans typically scan, eventhough such strategies will increase *kspc*.

Some of the sources of frustration identified by subjects could be addressed individually in the interface. In response to 2(a), regarding double letters, a type-o-matic feature could be added. For selection of coincidentally adjacent letters (2b), an arrow key could be pressed while the select key is held down. In response to 2(c), letters that occur frequently within a small range of positions in the layout could be assigned fixed positions when present in that range. Some of these solutions carry a cost, in that they make the interface more complex, demanding greater expertise of the user. Some also shift the nature of the layout towards that of a fixed layout, without gaining the ultimate benefit of total familiarity. Is another approach to redesign warranted?

In the post-interviews, several subjects stated that they could quickly process a 3 by 3 area of letters and determine whether the target letter was there. If it was not, they would begin to search from left to right. During the experiment we observed numerous occasions when subjects had difficulty finding a letter on the right side of the layout. Though statistical analysis of time data collected during the experiment needs to be done to confirm this, it appears that FOCL outperforms QWERTY when letters are close to the cursor, and thus easily located, but suffers badly when they are further away from it. Conversely, QWERTY--and fixed layouts in general--is far superior to FOCL when the distance between letters is larger, due to the user's familiarity with the layout. If this reasoning is sound, then a goal of further research should be to determine at what point the fluctuating approach ceases to be advantageous.

The preceding analysis suggests that a hybrid layout, such as the one in Figure 10, is warranted. A number of positions fluctuate and the remainder are fixed. Unlike the approach of addressing weaknesses individually, which lessens the benefits of FOCL by shifting the nature of the layout towards that of a fixed layout, this would preserve the benefits of each method.



A	B	C	D	E	Y	I	C	N	O	P	Q	R
	F	G	H	I	L	T	D	S	T	U	V	
	J	K	L	M	R	N	S	W	X	Y	Z	

Figure 10: A possible hybrid layout design

Conclusions

We feel the FOCL approach has promise despite the lack of a significant difference in entry speeds observed in our study. In particular, we are hopeful of the potential of a hybrid approach to yield a performance improvement. Also encouraging is the fact subjects improved in both conditions throughout the study, suggesting that perhaps a longer period is needed to observe a crossover in text entry speeds between the two conditions. As one subject, who felt the approach had considerable merit, noted, there are techniques that do not reveal their superiority to other methods before considerable practice, let alone ten 15-minute sessions.

Acknowledgements

We are grateful for the financial support of the Input Research Group at the University of Toronto, and to the members of the group for their many useful comments and suggestions.

References

1. Card, S. K., Moran, T. P., & Newell, A. (1983). *The Psychology of Human-Computer Interaction.* Hillsdale, NJ: Erlbaum.
2. Friedman, William F. (1960), *Military Cryptanalysis*, Washington, D.C.: United States Government Printing Office.
3. Green, D. W., Hammond, E. J., & Supramaniam, S., Letters and shapes: Developmental changes in search strategies, *British Journal of Psychology, 74* (1983), 11-16.
4. Hick, W. E., On the rate of gain of information, *Quarterly Journal of Experimental Psychology, 4* (1952), 11-36.
5. Hyman, R., Stimulus information as a determinant of reaction time, *Journal of Experimental Psychology, 45* (1953), 188-196.
6. Leventhal, L. M., McKeeby, J. W., & Mynatt, B. T., Screen keyboards: An empirical study of the effects of shape and character layout, in *Proceedings of INTERACT '91,* 108-112.
7. Marmaras, N., & Lyritzis, K., Design of an alternative keyboard layout for the Greek language, *International Journal of Human-Computer Interaction, 5* (1993), 289-310.
8. Mayzner, M. S., & Tresselt, M. E., Tables of single-letter and digram frequency counts for various word-length and letter-position combinations, *Psychonomic Monograph Supplements, 1* (1965), 13-32.
9. Shannon, C. E., A mathematical theory of communication, *Bell System Technical Journal, 27* (1948), 379-423, 623-656.
10. Soukoreff, R. W. & MacKenzie, I. S., Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard, *Behaviour and Information Technology, 14*, (1995), 370-37.
11. Underwood, B. J., & Shulz, R. W. (1960). *Meaningfulness and Verbal Learning.* Philadelphia: Lippincott
12. Welford, A. T. (1968). *Fundamentals of Skill.* London: Methuen.
13. Wilkund, M. E., & Dumas, J. S., Optimizing a portable terminal keyboard for combined one-handed and two-handed use, *Proceedings of the 31st Annual Meeting of the Human Factors Society* (1987), 585-589.