An Analysis of Novice Text Entry Performance on Large Interactive Wall Surfaces

Andriy Pavlovych *Wolfgang Stuerzlinger*

Dept. of Computer Science, York University
Toronto, Ontario, Canada
www.cs.yorku.ca/{~andriyp|~wolfgang}

Abstract

This paper presents a comparative study of several soft keyboard layouts for text entry on interactive whiteboards. We compare the traditional QWERTY with several other layouts, including theoretically optimal ones, such as FITALY. In contrast to previous work, we concentrate on novice performance, as few people ever gain expert status on whiteboards.

The results show that for a population of regular keyboard users, QWERTY is the best alternative and even the slightest deviations from that arrangement degrade the performance significantly. Based on the experiments, we present an in-depth analysis of the behaviour of users for several layouts from a cognitive viewpoint.

Keywords: Text entry, soft keyboards, interactive whiteboards.

1 Introduction

The problem of entering text on large interactive wall surfaces such as whiteboards comes up in various contexts. Single Display Groupware (Stewart, Bederson, & Druin, 2003) is defined as a system that can support collaborative work between people that are physically in relative proximity. Since users are sharing a display in the same room, they are free to interact face to face, which facilitates collaboration. Most often the collaboration involves manipulating and rearranging of the existing objects. Being able to easily enter data shared display would be an important step towards building a successful collaborative environment. Similarly, even in case of a single user interacting on large display (e.g. a demonstration of an interactive system, annotating a document), the need to enter text is encountered quite often.

Several approaches have been proposed for this problem. Solutions include regular (possibly wireless) keyboards, as the most obvious and easy to implement idea, using individual handheld devices, such as PDAs (Magerkurth & Stenzel, 2003), using handwriting recognition (Guimbretière, Stone & Winograd, 2001), using some gesture based character-recognition system, such as *Graffiti* or *Quickwriting* (Perlin, 1998) and soft keyboards. The main disadvantage of physical devices is the fact that they need to be carried around, thus restricting the user's freedom of movement and overall comfort. Also, usually there is no convenient horizontal surface available which reduces the user to single hand text entry. This negates the potential performance increase especially if text entry is needed only rarely. Character- and gesture-recognition systems are non-trivial to implement, often not sufficiently reliable and tend to have a steep learning curve, which makes their use by novices impractical. The disadvantage of soft keyboards is that they require some additional screen space when in use and may temporarily obscure the underlying content; however, this is offset by the benefit of no additional hardware that needs to be held and easier implementation.

In this paper, we explore several text entry techniques that use stylus-based interaction on a whiteboard. For a short overview of such methods, see (MacKenzie & Soukoreff, 2002). We will analyze how much the familiarity with the keyboard layout plays a role, compared to optimality of the keyboard layout, for novices, which is in contrast to previous work.

1.1 Decision to Test Novice Users

Our decision to focus on novice users is based on the fact that whiteboards are rarely used on a daily basis. As the frequency of use is not high, very few people ever become experts in using them and retain that status. Hence, a typical population of users will consist of novices to intermediates.

1.2 Previous Work

Previous work in this area (MacKenzie & Soukoreff, 2002; MacKenzie & Zhang, 1999; MacKenzie, Zhang & Soukoreff, 1999; Smith & Zhai, 2001) considered various soft keyboard layouts. There, the authors evaluate the techniques both analytically and experimentally. In particular, in (MacKenzie et al., 1999) they evaluate several techniques, including QWERTY, ABC[1] and FITALY, with novice users. The obtained performance statistics were 20.2, 10.6 and 8.2 words per minute, respectively. Note that in particular, FITALY is theoretically more optimal then QWERTY, so this result is a bit surprising. The authors also made some attempt to explain the discrepancies between the predicted and the measured figures. However, they did not elaborate why times other than the physical movement time, called "visual scan time" in the paper, differ significantly among layouts. In particular, the authors mentioned the Hick-Hyman law (Hick, 1952) as a potential reason, but do not explain why this should give different results for different layouts with the same number of buttons! Our work will address this issue in more detail.

Another paper related to this work analyzed the HALF-QWERTY layout (Matias, MacKenzie & Buxton, 1996), which is designed for one-handed text entry. Two letters are put on each key on one side of the keyboard and a modifier key is used to distinguish the side. The authors report a performance of 13.2 wpm after 50 minutes. They also argue that their layout benefits from learning transfer from the regular QWERTY layout, while being significantly smaller compared to the QWERTY layout. But this argument is not explored further in that paper.

One notable omission of the previous studies is that in none of their papers do the authors report which text entry method was actually preferred by the users. This is important as one can easily switch between different keyboard layouts on interactive whiteboards.

Smith and Zhai (Smith & Zhai, 2001) investigated how partial alphabetical ordering in an optimized layout positively influenced the performance by testing two optimized layouts with and without an alphabetical bias. They also hypothesized that the differences in performance were due to (only) differences in visual search area sizes for the two layouts.

In our current study, we evaluate several soft keyboard layouts on an interactive whiteboard and attempt to address all of the above issues. In particular, we focus on visual search times for novices and on user preferences. We also show evidence that there are more components to text entry performance beyond the mentioned factors of visual search time and physical movement time, one of such components being the cognitive overhead. We will also present some evidence that, even for applications such as text entry on whiteboards, the ubiquitous QWERTY layout is still one of the best choices.

1.3 Choice of Soft Keyboard Layouts

At present, soft keyboards are most often used in devices like tablet-based computers and portable digital assistants. Similar to desktops coupled with regular keyboards, it is usually assumed that the frequency of use for such devices is high (as the devices are *personal*), as well as that the precision of the movements is high. The opposite seems to be true for whiteboards and for other interactions on large surfaces: the systems are used for significantly shorter periods, with long interruptions, in a presence of various distractions and, due to lack of practice, the pointing performance of the user is lower that could be expected for similar targeting tasks done regularly.

Clearly, it is expected that the size of a device would influence the choice of a layout for a keyboard. E.g. a layout designed for dual-handed 10-finger typing on a horizontal surface (keyboard) may perform different from layouts for single-handed text entry, text entry with a single finger/thumb (phone keypad), or entry with a single stylus (PDAs, pocket computers). In a case of a regular full-size keyboard, for example, minimizing inter-key movements is less of a concern, as we can position all 10 fingers almost uniformly distributed across the keyboard's width.

For large interactive surfaces, and based on the results of previous work we postulate that a keyboard layout that minimizes the average movement distance between buttons would provide better performance than wide QWERTY. Consequently, we choose the following four different keyboard layouts for our study.

For the first, we chose the "standard" QWERTY layout. We hypothesize that QWERTY would cause the users to perform slower on a *whiteboard* with it than with the other layouts, as this layout is much wider than other layouts and hence necessitates a larger average movement distance. As a representative for a group of several optimized

---

[1] In (MacKenzie et al., 1999), the authors use the version of the alphabetical layout consisting of letters arranged in two columns with the space as a third column, unlike the layout we used (see Figure 4). The expert prediction for such a layout is smaller than for ours (MacKenzie & Soukoreff, 2002).

layouts, we choose FITALY (*Fitaly*, 2005). This layout is it is one of the fastest for experts (MacKenzie & Soukoreff, 2002) and has been available commercially for some time for Palm pocket computers. Other (equally valid) candidates would have been OPTI (MacKenzie & Zhang, 1999) and METROPOLIS (Hunter, Zhai & Smith, 2000).

Figure 1: Keypad of a *Gmate Yopy YP-3000* Linux PDA.

As third layout we choose the alphabetical layout (ABC) – an ordering that is familiar to most people, even non-computer users, and which is sometimes used in applications targeted at novices, such as self-serve kiosk terminals. Finally, we choose a *split*-QWERTY layout as fourth candidate. Such a keyboard can be obtained by splitting a regular QWERTY keyboard in two halves and putting the right half under the left. This is a relatively small deviation from the most ubiquitous layout, and hence this layout is expected to work well with users familiar with QWERTY, yet may provide performance benefits because it is spatially more compact than the normal QWERTY layout. It is interesting to note that such a layout is used commercially in a line of Linux-based PDAs called *Gmate Yopy YP-3000* (*Gmate Yopy*, 2005; also Figure 1). A somewhat similar concept is also used in *Matias HalfKeyboard* (Matias et al., 1996).

Note that the second, third and fourth layout are most easily implemented by putting buttons in a 6-by-6 grid, which makes their sizes directly comparable.

1.4 Hypotheses

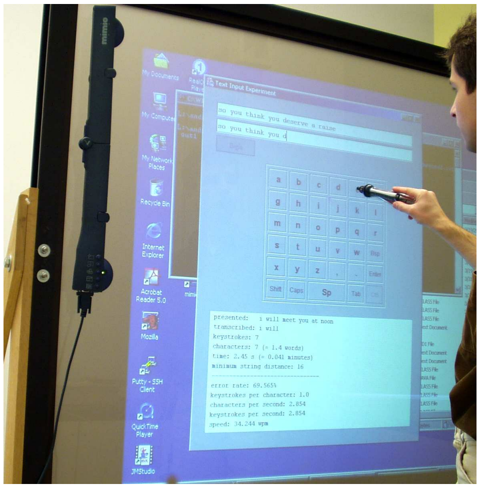
Based on the arguments presented above, we came up with the following hypotheses:

- We expected the QWERTY layout to perform worse on a whiteboards than on a relatively smaller tablet (MacKenzie et al., 1999).
- Also, we expected the *split*-QWERTY layout to outperform the ABC layout due to users' better familiarity with QWERTY layout.
- Finally, we expected that for novices the FITALY layout perform worse than other layouts, because users will find it hard to learn.

2 Test method

2.1 Participants

We recruited 8 participants through advertisements posted around the university campus. Four participants were female and all were right-handed. The ages ranged from 21 to 32 with the mean of 25.4. All had extensive computer experience (five years and more), used their computer daily and had the tying speed ranging from medium to fast (though no one rated their speed as "expert"). Participants were compensated upon completion of the user study.



so you think you deserve a raise

so you think you d

```
presented:   i will meet you at noon
transcribed: i will
keystrokes: 7
characters: 7 (= 1.4 words)
time: 2.45 s (= 0.041 minutes)
minimum string distance: 16
------------------------------------
error rate: 69.565%
keystrokes per character: 1.0
characters per second: 2.854
keystrokes per second: 2.854
speed: 34.244 wpm
```

Figure 2: Equipment used in the experiment. The black bar attached to the left of the screen is the *mimio Xi* device, which digitizes the input position.

2.2 Apparatus

2.2.1 Hardware

We used an interactive whiteboard (Figure 2) for our study. A projector, with a resolution of 1024 by 768 pixels, was located behind the diffuse screen. The zoom setting of the projector was adjusted so that the size of the buttons on the screen was 50 x 50 mm (2″ x 2″). We choose this size to minimize the chances of hitting the wrong key during fast entry, while still allowing for a reasonably compact layout. A pointer, slightly larger than a common flat tip marker, was utilized to interact with the virtual buttons on the screen. The position of the pointer was digitized by a *mimio Xi* device, which was attached to the screen surface with suction cups.

2.2.2 Software

The core of the software that we used in the experiment was written in Java and had been used previously for text entry experiments (MacKenzie, Kober, Smith, Jones & Skepner, 2001; Pavlovych & Stuerzlinger, 2004). We modified it to show the on-screen keyboards and to interpret the characters entered on it. Four separate versions of the program were created, one for each keyboard layout. Figures 3 and 4 show the user interfaces of the programs for each of the layouts. The size of the buttons on the screen was the same for all methods, and was equal to 50x50 mm with a spacing of 6 mm.

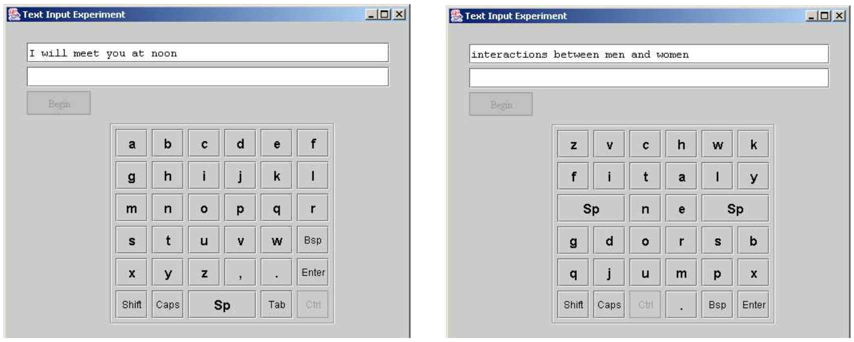
make up a few more phrases

raindrops keep falling on my head

Figure 3: *QWERTY* and *Split-QWERTY* layouts.

The first text field shows the phrase to enter and the second field shows the text that the user has entered so far. The software recorded all key presses and the times when they took place. Furthermore, the software did statistical calculations such as time to enter a phrase, error rate, average WPM speed for the phrase, key repeat time and so on and displayed these values at the bottom of window for control purposes (not shown).

All of the displayed buttons were functioning; however, users had to use only the 26 letter buttons, Space, Enter, and a Backspace button. Furthermore, all phrases utilized only lower case letters. These restrictions were imposed to simplify the testing procedure and to reduce the potential for increased variance due to users employing different strategies to enter capitals or to correct errors, for example.



I will meet you at noon

interactions between men and women

Figure 4: Alphabetic *ABC* and *FITALY* layouts.

In our system, the timer started with the first typed key for each phrase and stopped with the Enter key, so that the participants could rest between the phrases at their discretion. They were informed about this feature of the system.

2.2.3 Set of phrases

We used the set of phrases of English text from (MacKenzie & Soukoreff, 2003). This set has been used before in several text entry experiments and is representative of English. Forty phrases to be entered were chosen from this set and were permanently assigned to each method in such a way so that the number of characters entered with each method was the same and there was no significant difference in the average inter-key movement for each of the four methods (thus removing a potentially confounding factor). We choose this experimental design to increase the sensitivity towards more subtle features of the keyboard layouts by reducing the variance. To reduce the variability even further, the sequence in which the phrases appeared was always the same across participants.

3 Procedure

Each experiment lasted between 30 and 40 minutes. During that time each participants completed 4 sessions, one with each keyboard layout, for a total of 40 phrases. We used a within-subjects design and counterbalanced the order of entry methods via the Latin Square approach to compensate for potential learning transfer effects.

3.1 Instructions

Before the test, the participants were given a brief instruction as to their task, were shown how the software worked, which buttons to press after each phrase and so on. They were told that they were allowed to take short (e.g. 10 s) breaks during the test between the phrases. They were also given the freedom to adjust the position of the text entry window on the screen, mostly to accommodate for different personal height. Before the session, users were encouraged to enter a phrase or two for practice and to get accustomed to the system. The participants were asked to enter text “as quickly and as accurately as they could”. They were told that the errors could be corrected via using a backspace button.

4 Results

4.1 Entry Speed

Overall, the mean entry rate was 17.6 wpm for QWERTY, 11.6 wpm for the *split*-QWERTY, 11.8 wpm for ABC, and 8.0 wpm for FITALY. The main effect of the entry method was significant ($F_{3,7} = 44.58$, $p << 0.05$). A Tukey-Kramer Multiple-Comparison test reveals that the techniques can be grouped into three distinct groups with regards to speed, with QWERTY being the fastest, *split*-QWERTY and ABC being in the middle, and FITALY being the slowest. ($\alpha = 0.05$, *DF* = 21, *Mean Square Error* = 28.40, *Critical Value* = 3.94). Figure 5 demonstrates the entry speed in words per minute for each method for each user.

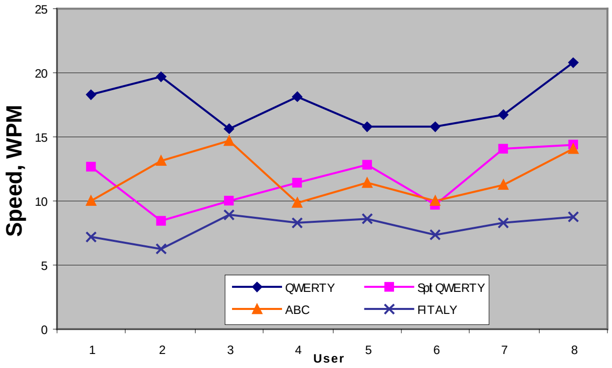


Figure 5: Entry speed (wpm) by entry method and user.

4.2 Error Rate

The error rate in % is defined here as the ratio of the Minimum String Distance (MSD) to the length of the target entered text (whichever is greater).

The mean error rate over all methods and users was 0.71% and there was no significant difference between the methods ($F_{3,7} = 1.94$, $p > 0.05$), which indicates that the participants were equally conscientious across all layouts.

4.3 Errors: Unnecessary Key Presses

With each layout, only a single key press is required to enter a character. However, the number of keystrokes per character (KSPC) (MacKenzie, 2002) obtained in the experiments was always larger than 1 due to the fact that users occasionally made errors and corrected them via backspaces.

Overall, the average the number of KSPC was 1.075 for QWERTY, 1.019 for the *split*-QWERTY, 1.020 for ABC, and 1.049 for FITALY, and the main effect of the entry method was significant ($F_{3,7} = 3.60$, $p < 0.05$). Tukey-Kramer

Multiple-Comparison Test indicated a statistical difference only between QWERTY and *split*-QWERTY ($\alpha = 0.05$, $DF = 21$, $MSE = 0.0159$, *Critical Value* = 3.94).

4.4 Key Press Rate

Overall, the key press rate was 1.55 keystrokes per second (KSPS) for QWERTY, 1.00 KSPS for ABC, 0.99 KSPS for the *split*-QWERTY, and 0.69 KSPS for FITALY. The Tukey-Kramer Multiple-Comparison Test reveals that there are again three distinct groups observable, with QWERTY being the fastest, *split*-QWERTY and ABC being in the middle, and FITALY being the slowest ($\alpha = 0.05$, $DF = 21$, $MSE = 0.22$, *Critical Value* = 3.94).

4.5 Subjective Responses of Participants

After the test, the users were asked to fill out a short questionnaire, in which they had to order the four systems in terms of perceived speed, ease of learning, ease of use (absence of physical and mental fatigue), and overall preference in a scenario where they had to use the techniques on a regular basis, to enter several phrases at a time. They ranked the systems by assigning a number between 1 and 4 to each. To summarize the responses, all users rated QWERTY as the best and FITALY as the worst, while the other two were rated roughly equally for each criterion. For every one of the parameters, the results look similar. The main effect of the entry method was highly significant (“ease of use”: $F_{3,7} = 63.00$, $p << 0.05$).

Some users commented that they would have preferred a system in which they could type with their fingers, instead of a stylus. Also, there were participants who felt that the displayed buttons were slightly too large.

5 Discussion

Overall, it is clear that the QWERTY layout performed best and the FITALY layout performed worst. Our result for *split*-QWERTY is close to the 13.2 wpm obtained in a previous study for HALF-QWERTY (Matias et al., 1996). The performance of QWERTY respective FITALY is reasonably consistent with the 20.2 and 8.2 wpm obtained by previous work (MacKenzie et al., 1999). Note that the sessions in the experiment in (Matias et al., 1996) were longer and hence we cannot expect the averages to match precisely.

Concerning our hypotheses, we find that our experiments disprove the first two hypotheses and confirm the third one. We discuss this in the following sections.

5.1 Entry Speed

Our hypothesis that the regular, full size QWERTY keyboard would *not* be the fastest was not confirmed; instead the entry speed was about 50% faster than the closest contender, the ABC keyboard. This discovery seems somewhat surprising and counterintuitive. However, in the following text we will try to explain the reasons behind this.

Entry speed is can be decomposed into two components: the number of keystrokes per symbol and the frequency of performing the keystrokes. Let’s consider each in turn:

5.2 Number of Keystrokes per Character

In our case the number of keystrokes per character corresponds roughly to the frequency of “mis-hits”, that is hits of buttons different from the required ones. This equivalence is appropriate here, since the final error rate in the entered text is statistically the same among the layouts.

Although the QWERTY layout had the largest number of extraneous key presses, it still achieved the highest overall rate. That large number of mis-hits can be explained by the fact that most users know the traditional layout very well and tend to go faster than the rate at which button targeting errors are infrequent. For FITALY, the increase in mis-hits could have been caused by frustrations about the non-intuitiveness of the layout as evidenced in the post-test questionnaire. However, we have to add that in both cases, this is only a speculation.

5.3 Key Repetition Rate

Key press rate was the *major* varying factor between the techniques, with FITALY's being more than two times slower than QWERTY's. For better clarity, we computed the time of making a key press by taking the reciprocals of the corresponding values (Table 1, first column).

5.3.1 Visual Search Time

An earlier study analysed the theoretical and the actual speed of some keyboard layouts (MacKenzie et al., 1999). There, the authors predicted the text entry speed for novices by arguing that the press of a key is preceded by a visual search time. This visual search time can be computed through the Hick-Hyman law (Hick, 1952). For 27 keys/choices and the commonly used slope of 200 ms/bit, the time value for our experiments is 951 ms.

5.3.2 Movement time

The inter-key movement time in previous papers (MacKenzie & Soukoreff, 2002; MacKenzie et al., 1999). was predicted via Fitts' law (Fitts, 1954). Our layouts are identical or almost identical to the ones considered in those papers, thus we used their expert prediction values to derive the average movement time for our methods[2] (Table 1, middle column). One of the properties of Fitts' law is that the computed values do not change if one linearly scales the keyboards. The authors point out that this might not be true in general, and that in extreme cases the numbers would likely increase slightly. As there is no data on the *split*-QWERTY layout in that paper, we estimated the corresponding number to be roughly similar to the one for ABC and definitely no larger than for QWERTY and not smaller than for FITALY.

Finally, the last column of Table 1 shows the differences between the measured time and the time predicted by Fitts' law. One can see that for FITALY the number is relatively close to the predicted 951 ms visual search time, which indicates that for this method, the visual search time appears to dominate and the visual search is likely to encompass the complete set of buttons.

Table 1: Break up of the *time to press a key* (see text).

Technique	Total Time, measured, ms	Predicted Movement Time, ms	Difference, ms
QWERTY	645	399	246
split-QWERTY	1010	approx. 300–400	610–710
ABC	1000	369	631
FITALY	1449	286	1163

For the other layouts, the presence of a visual search time is not unnatural, considering that the buttons on an interactive whiteboard are virtual and provide no tactile feedback, thus requiring at least some visual attention from the users. Moreover, the users had no opportunity to develop any significant kinaesthetic memory to allow them to blindly point at the required letter (assuming they knew where it was located – a reasonable assumption for most QWERTY users).

Although it is evident that all techniques required at least some visual attention, we conjecture that there are different behaviours for the different layouts. For QWERTY, we expect that the visual attention is confined to the immediate neighbourhood of the target button, while in the case of *split*-QWERTY and ABC, we expect that the visual search encompasses only a partial area of the whole layout.

To support this argument, we attempted to derive the average number of buttons visually searched, by taking the number in the last column of Table 1 and applying the inverse Hick-Hyman law (with the same 200 ms/bit slope) to derive the number of buttons searched visually. This is shown in the first column of Table 2.

Not surprisingly, the size of the visual search for QWERTY is relatively small. For the *split*-QWERTY and ABC layouts the visual search encompasses a larger part of the keyboard. It is very instructive to correlate this data with the

---

[2] There is a discrepancy with respect to entry speeds for FITALY between (MacKenzie & Soukoreff, 2002) and (MacKenzie et al., 1999). We used the data from (MacKenzie & Soukoreff, 2002) as it is the more recent result.

expected behaviour of users (shown in the middle column of Table 2. For QWERTY, we expect near-optimal behaviour due to the familiarity, and hence we can assume that the observed visual search time is almost optimal. For the *split*-QWERTY layout, we conjecture that the familiarity with the QWERTY layout decreases the visual search size at least to some degree. Hence, we expect to see evidence of additional mental overhead (in the table we used the average of the measured visual search size for the *split*-QWERTY and QWERTY layouts as an approximation for this). For the ABC layout, we expect that the user searches through a sub-range of the alphabet and hence have to assume that the search is limited to a small, but non-zero size. Finally, we can observe that the similar argument breaks down partially for FITALY, as there are only 28 relevant buttons on the keyboard, which provides a natural maximum for the visual search.

Table 2: Number of elements searched through for different methods and estimated values for cognitive delays (see text).

Technique	"Visual Search Size"	Expected Search Size	Cognitive Delay, ms
QWERTY	2.34	2.34	~0
split-QWERTY	8.28–11.71	(5–7)?	49–246?
ABC	8.9	≤ 8.9	≥ 0
FITALY	56 (!)	≤28	≥202

Note that we cannot be sure that our explanations are exhaustive or 100% certain. Hence we choose to state our numbers as inequalities. However, we can assume that at least for the *split*-QWERTY and FITALY layouts the difference between the predicted and the measured time between key presses can be attributed at least partially to cognitive overhead. To give the reader an idea of the magnitude of the factors, we converted the difference between the observed search size and the expected search size into times, and show them in the last column of Table 2.

5.3.3 Cognitive Overhead

Cognitive overhead is usually described as the additional effort or concentration in keeping track of several things (such as positions of letters on keys) at a time. In this study, examples of such overhead would be trying to memorize the layout, trying to remember in which half of the *split*-QWERTY the required key is, and trying to recall the relative position of a letter in the alphabet.

One easily visible effect of cognitive overhead is the evidence of mental fatigue, as observed in our case by the users rating FITALY as the most tiring of all. Mental fatigue decreases the speed and correctness of memory recall (Anderson, 2000), which eventually results in a decline of performance, which indeed took place with FITALY, approximately after the 8th out of 10 phrases (see Figure 6). Although for *split*-QWERTY, the cognitive overhead could be just as large as for FITALY, its search size is considerably smaller, thus reducing the mental strain and causing users to like it significantly better in the questionnaires as well as producing no evidence of a performance drop-off.

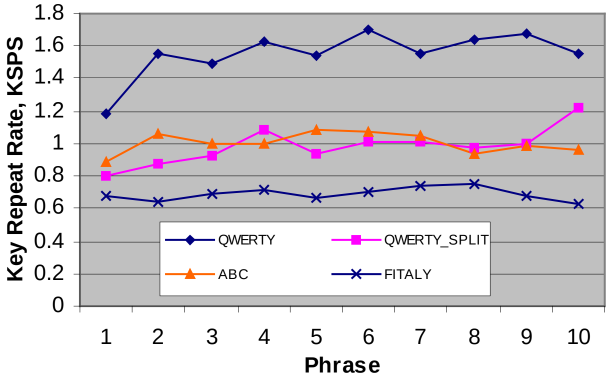


Figure 6: Change of Key repeat rate for different layouts over time.

Based on this reasoning, we offer a potential explanation behind the fact that participants always required significantly longer for a keystroke with the FITALY layout compared to the three other methods. We hypothesize that this layout has a higher cognitive overhead because its letter ordering looks almost random to a novice user, and, for that reason, requires much longer visual search times. In contrast, QWERTY does very well here, as our participants were very familiar with this layout. The ABC layout also did much better than FITALY, as the knowledge of the order of the alphabet seems to reduce the size of the visual search.

However, based on the fact that many public kiosk-systems use the ABC layout, we need to point out that our results are biased as we utilized a population of QWERTY users. We made this choice as it allowed us to investigate the cognitive overhead through comparison with other layouts. We add that for systems designed for an audience not familiar with the QWERTY layout we expect that the ABC layout perform best.

Overall, text entry speed is affected not only by the average movement time, but also by the familiarity of the arrangement (affecting the size of the neighbourhood visually searched) and the mental strain a layout induces on a user. The study shows that even the slightest deviation from the common QWERTY layout, as evidenced from the results of the *split*-QWERTY condition, leads to a dramatic drop in performance. Note that, although the results for *split*-QWERTY and ABC are almost identical, we attribute these results to different factors (see Table 2 and the corresponding text).

5.4 Comparison with Other Techniques

It is interesting to point out that the performance figures obtained by the three slower layouts can also be achieved with other text entry approaches. In particular, it has been shown that 8 wpm is a reasonable speed to expect for *novices* on a phone keypad (see e.g. *Less-Tap*, Pavlovych & Stuerzlinger, 2003), and slightly higher speeds are achievable with the use of prediction (James & Reischel, 2001; Pavlovych & Stuerzlinger, 2004). This effectively means that, if the standard QWERTY layout is not viable for some reason, then other text entry paradigms should also be considered by a designer of a text entry system.

However, we need to point out that one of the advantages of the general QWERTY layout is ignored in this study: its capability of easy entry of numbers, punctuation marks, as well as numerous special symbols – something that is harder to implement in smaller form factors.

5.5 Future Directions

In this study, we limited ourselves with to consider only *text* entry. However, in real-world applications, and especially on whiteboards, entry of numbers may be nearly as important. Of course this has a significant impact on the design decisions for the layout. However, we expect that the major design guidelines of familiarity, etc. are still the same.

Another issue ignored in this study was editing facilities and navigation keys. It has been shown before that these “other” key presses are very important (Soukoreff & MacKenzie, 2003). However, currently we cannot see how one can seamlessly combine the existing knowledge about text *entry* and text *editing* and subsequently incorporate the result into future designs. On the other hand we would like to point out that on an interactive whiteboard, at least navigation could be done directly with the pointing device, which makes this less of an issue.

6 Summary

We presented a comparative study of four text entry techniques for interactive whiteboards. The results of the user study show that, for novice users familiar with the QWERTY layout, this layout works best and achieves more than 50% better performance than other layouts. The alphabetical and *split*-QWERTY layouts are competitive, but FITALY (a representative method from a class of theoretically optimal layouts) performed worst. Thus, we demonstrated that even the *smallest* deviations from the familiar layout might significantly decrease the performance.

Our analysis of the data shows that novice performance is not limited by the raw movement time, but rather by other factors. To get a better insight into this fact, we presented a breakdown of the times involved and separately estimated the visual search time and the cognitive overhead. Finally, we derived guidelines for future work in design of text entry systems for whiteboards.

7 Acknowledgements

The authors would like to thank Scott MacKenzie for providing us with software for the experiments, as well as National Sciences and Engineering Research Council of Canada for funding.

8 References

Anderson, J. R. (2000). Learning and Memory (2nd ed.). New York: Wiley.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, 47(6), 381-391.

Fitaly from Textware Solutions. Verified February 05, 2005, http://www.fitaly.com/.

Gmate Yopy YP-3000. Verified February 05, 2005, http://www.gmate.com/english/product/yp3000.htm.

Guimbretière, F., Stone, M., & Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. *Proceedings of UIST 2001*, 21-30.

Hick, W. E. (1952). On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4, 11-26.

Hunter, M., Zhai, S., & Smith, B. A. (2000). Physics-based graphical keyboard design. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2000*, 157-158.

James, C. L., & Reischel, K. M. (2001). Text input for mobile devices: Comparing model predictions to actual performance, *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI 2001*, 365-371.

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

MacKenzie, I. S., Kober, H., Smith, D., Jones, T., & Skepner, E. (2001). LetterWise: Prefix-based disambiguation for mobile text input. *Proceedings of UIST 2001*, 111-120.

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2003*, 754-755.

MacKenzie, I. S., & Soukoreff, R. W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17, 147-198.

MacKenzie, I. S., & Zhang, S. Z. (1999). The design and evaluation of a high performance soft keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems - CHI '99*, 25-31.

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, 18, 235-244.

Magerkurth, C., & Stenzel, R. (2003). A Pervasive Keyboard — Separating Input from Display. *Proceedings of the first IEEE Conference on Pervasive Computing and Communications (PerCom'03)*, 388-395.

Matias, E., MacKenzie, I. S., & Buxton, W. (1996). One-handed touch typing on a QWERTY keyboard. *Human-Computer Interaction*, 11(1), 1-27.

Pavlovych, A., & Stuerzlinger, W. (2003). Less-Tap: A Fast and Easy-to-learn Text Input Technique for Phones. *Graphics Interface 2003*, 97-104.

Pavlovych, A., & Stuerzlinger, W. (2004). Model for non-Expert Text Entry Speed on 12-Button Phone Keypads. *Proceedings of the 2004 conference on Human factors in computing systems – CHI 2004*, 351-358.

Perlin, K. (1998). Quikwriting: Continuous Stylus-based Text Entry, *Proceedings of the 11th annual ACM symposium on User interface software and technology – UIST 1998*, 215-216.

Smith, B.A., & Zhai, S. (2001). Optimised Virtual Keyboards with and without Alphabetical Ordering – A Novice User Study. *Proceedings of INTERACT'2001*, 92-99.

Soukoreff, R. W., & MacKenzie, I. S. (2003). Input-based language modelling in the design of high performance text input techniques, *Graphics Interface 2003*, 89-96.

Stewart, J., Bederson, B. & Druin, A. (2003). Single display groupware: a model for co-present collaboration, *CHI'99 Conference Proceedings*, 286-293.