

Qanti: A Software Tool for Quick Ambiguous Non-standard Text Input

Torsten Felzer¹, Ian Scott MacKenzie²,
Philipp Beckerle¹, and Stephan Rinderknecht¹

¹ Institute for Mechatronic Systems, Technische Universität Darmstadt,
Darmstadt, Germany

² Department of Computer Science and Engineering, York University,
Toronto, Canada M3J 1P3

{felzer, beckerle, rinderknecht}@ims.tu-darmstadt.de,
mack@cse.yorku.ca

Abstract. This paper introduces a single-key text entry application for users with severe physical impairments. The tool combines the idea of a *scanning ambiguous keyboard* (which promises unusually high entry rates) with *intentional muscle contractions* as input signals (which require much less physical effort compared to key presses). In addition to the program architecture, the paper presents the results of several evaluations with participants with and without disabilities. An entry speed of 6.59 wpm was achieved.

Keywords: Human-computer interaction, scanning, ambiguous keyboards, intentional muscle contractions.

1 Introduction

Persons with physical disabilities are often unable to use a standard keyboard when interacting with a computer. Therefore, text-based communication typically involves an alternative input method, often relying on *scanning*. With scanning, the computer “suggests” an action (e.g., the selection of a character) by cyclically highlighting through a set of options with a *scan delay* τ (typically in the range 0.5 s to 2.0 s). The user selects the highlighted option by issuing a *selection marker* (which means, for example, pressing a single key or actuating a physical switch).

Unfortunately, while demanding less physical effort, scanning systems are usually prohibitively slow – considerably slower than keyboard-based text entry. Therefore, entering text is often cumbersome for physically disabled users with the consequence that communication methods requiring fast typing (e.g., Internet chat) are not possible. A fast one-key input device could tremendously improve the quality of life for physically disabled users by empowering them to participate in a world of communication opportunities formerly denied to them.

This work builds the idea of a *scanning ambiguous keyboard* (SAK) – as presented in ref. [1] – into a fully-implemented system. The SAK approach considers

multiple selections in a single scan step, which promises unusually high entry rates. However, SAK designs generally work only for words in a known list – a *dictionary*. Our system addresses this using a spell mode to add new words to the dictionary. In addition, the scanning approach is combined with the input signals in [2], i.e., *intentional muscle contractions*, to further reduce the required physical effort.

The resulting software application – dubbed *Qanti* for “Quick Ambiguous Non-standard Text Input” – is a tool specifically for persons with severe physical disabilities, allowing the user to effortlessly enter text in an acceptable amount of time. After reviewing approaches to one-key text entry in the literature in section 2, the *Qanti* solution is described in section 3. The system’s practical effectiveness will be demonstrated in several small-scale usability studies in section 4. The paper concludes with a summary and a look at future work.

2 Known Approaches

Making computers accessible to physically disabled users is challenging for at least two reasons. First, using a computer is virtually impossible without entering text – and alternative text entry applications (with a small number of keys or input signals) are therefore needed (e.g., [3]). Second, reducing the number of keys to represent the 26 letters of the alphabet (plus certain “special” characters like space, dash, apostrophe or various punctuation) brings about special challenges [4,5].

To confront these challenges, numerous acceleration techniques supplementing the scanning idea [6] have been developed. A straightforward idea is to reduce the scan delay, as a shorter delay increases the speed. But there is a trade-off: very short delays cause erroneous or missed selections, *reducing* the entry rate. The approach of Simpson and Koester [7] tries to cope with this problem by dynamically adapting the scan delay (“on the fly”) based on user performance.

Another idea is to use *word completion* (e.g., [8]), where the computer tries to “guess” a word based on the characters entered so far. Displaying a list of candidate words and offering a mechanism to select the desired candidate early helps save “keystrokes” (i.e., selections). However, constantly having to inspect the candidate list also increases the *cognitive load* on the user.

As text entry progresses, selectable options might increase in likelihood compared to alternatives (according to syntactic or semantic considerations). The system described by Wandmacher et al. [9] takes advantage of this by dynamically reordering the scanned options, so that more likely options are selected faster.

The solution presented here reduces the number of steps in a scan cycle by associating more than one character to a single virtual key (also [10]). The actual words belonging to the key sequences are computed using dictionary-based disambiguation.

3 Software Tool

The text entry application *Qanti* is depicted in fig. 1. It divides the screen into four areas: the sequence selection area on the upper left, an output area on the

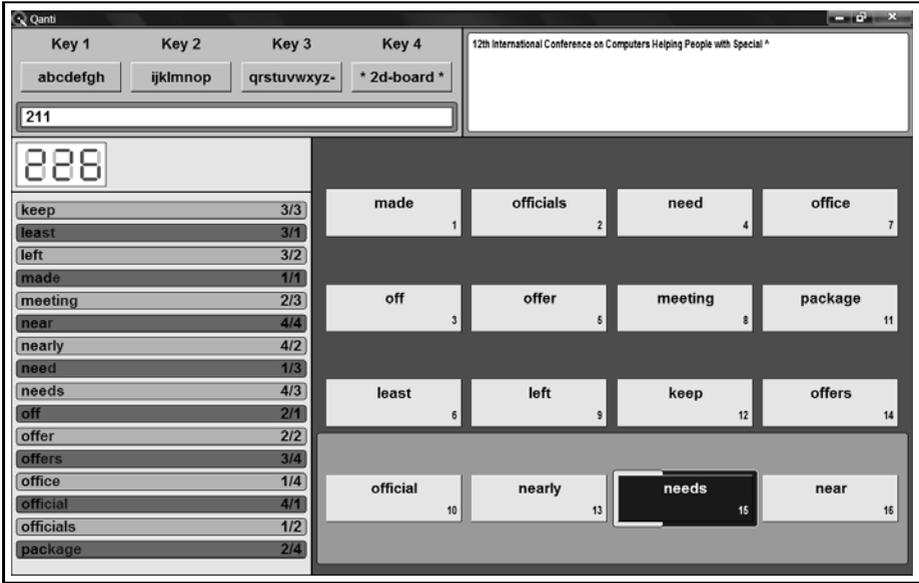


Fig. 1. Specialized writing tool *Qanti*

upper right showing the text entered so far, an information area on the lower left, and a large 4×4 scanning board on the lower right.

3.1 Typical Operation

Entering a word typically involves three phases: a sequence selection phase, a candidate selection phase, and a finalization phase. First, the user enters a sequence consisting of the linearly scanned virtual keys in the sequence selection area by issuing intentional contractions as *selection markers*. Each of the first three keys is associated with approximately one third of the alphabet (in analogy to a phone keypad where most keys are associated with three letters). An important property of this phase is that the scanning timer is reset after each selection with the highlight persisting on the current key. As a consequence, the user can make multiple selections during a single scan step – which makes the method enormously fast.

The program responds to each selection of a “character key” by determining all candidate words matching (and completing!) the current sequence in a frequency-ordered dictionary. The 16 most frequent candidates are displayed in the information area (in alphabetic order) *and* on the virtual keys of the two-dimensional board (ordered by frequency). If the desired word is among those presented, the user can switch to the next phase by selecting the fourth virtual key in the sequence selection area. After the selection of the intended candidate (using conventional row-column scanning), the user is prompted to finalize the

word by choosing a *modifier* (which adds a space, comma, or period at the end or capitalizes the first character).

Note that the candidates are *not* placed in a “row-oriented” but in a “diagonal-oriented” order on the 2d-keyboard (seen from the button indexes in fig. 1). The reason for this is the selection time required using row-column scanning – for instance, selecting the third button in row 1 (which would be the position of candidate 3 in “row-oriented” order) takes *longer* than selecting the first button in row 2 (the “row-oriented” position of candidate 5)!

Finally, a feature of the software is its multilingual support. The user may choose either English or German. This choice affects the output strings used throughout the program, the pre-loaded dictionary, and the selectable characters, for instance, “umlauts” when using German.

3.2 Menu Mode

In addition to typical word selection, the user can enter a menu mode by selecting the fourth “sequence key” *without* prior selection of a “character key” (i.e., without any other selections in the same scan cycle). The menu options include resetting the current sequence, configuring the scan delay, or invoking a conventional row-column scanning keyboard for entering “out of dictionary” words.

Another option launches an automated *test protocol* which randomly selects phrases from a phrase set supplied to the program, displays the phrases in the output area, and records the time for the user to copy the phrases.

After starting the test, an additional timer is displayed (not shown in fig. 1) together with the phrase to be copied. The user then has the opportunity to study the phrase and “plan ahead”, because the timer is halted until the next input signal. When the user is finished transcribing the phrase, he/she can add a newline character. *Qanti* responds to that by computing the current entry speed – in *words per minute* (wpm) – etc., writing the statistics to disk, and displaying the next phrase to be copied (with the timer again suspended until the user issues another input signal).

3.3 Input Signals

As already mentioned, the software supports intentional muscle contractions as input signals. The operation of the program relies on *scanning* through a number of options, and to make a selection in this context, the user just issues tiny contractions of a single muscle of choice.

To detect those contractions, the activity of the muscle in question – indirectly recorded with the help of a piezo element – is constantly compared to a user-dependent threshold. Exceeding this *sensitivity* threshold triggers a contraction event and, thus, a selection.

The prototypical configuration – utilized in the evaluation studies in the next section – is tailored to the brow muscle, i.e., it involves a piezo pressed against the eyebrow by means of an elastic sports headband. The study participants

therefore merely needed to raise the eyebrow (or to *frown*), in order to produce selection markers.

4 Empirical Data

The program is particularly powerful concerning long and frequently used words. For instance, *Qanti* allows the user to enter the 14-character string “International ” (including the trailing space) in less than 12 seconds. This is of course a non-typical, very advantageous example, yet it shows that peak velocities of more than 1 character per second (i.e., more than 12 wpm) are possible in theory.

The practical aspects of *Qanti* were tested through three studies: a pilot study with one subject diagnosed with *Friedreich’s Ataxia* (FA), an attempted (but not completed) test with two subjects with *Spinal Muscular Atrophy* (SMA), and a usability study involving five able-bodied users. Each participant was male and between 22 and 39 years old.

The task of all eight subjects was to copy phrases – as quickly and accurately as possible (correcting errors if noticed) – presented at random from a phrase set supplied to the program. It was ensured that each unique word in the phrase set was also in the program’s dictionary. The phrase set consisted of 56 common German sayings and proverbs with an average length of 56.2 characters (ranging from 31 to 81) – some typical examples are shown in fig. 2.

<p>Es wird nichts so heiß gegessen, wie es gekocht wird. Wächst das Gras im Januar, so wächst es schlecht im ganzen Jahr. Was du heute kannst besorgen, das verschiebe nicht auf morgen. Wer anderen eine Grube gräbt, fällt selbst hinein. Wovon man nicht sprechen kann, darüber muss man schweigen.</p>
--

Fig. 2. Sample phrases used in the evaluations

The FA patient in the pilot study has used a wheelchair for more than 20 years and has considerable motor problems. However, his usual method of entering text involves a standard keyboard at a typing rate of typically 3 – 6 wpm. He has participated in a number of empirical studies concerning alternative text entry and intentional muscle contractions. He was even familiar with *Qanti*, so he was a perfect choice to do some initial tests and verify that the software runs smoothly (the purpose of the pilot study).

The task was to enter five blocks consisting of three phrases each, using the participant’s preferred scan delay (1000 ms). To avoid over-exertion, rest intervals of at least five minutes were inserted between blocks, and the test protocol described in the previous section provided for additional pauses between individual phrases. The average entry speed achieved in the five blocks is illustrated in fig. 3a).

The participant indicated that he was quite satisfied with this result (around 2.5 wpm), which was close to the lower bound of his usual typing speed – yet

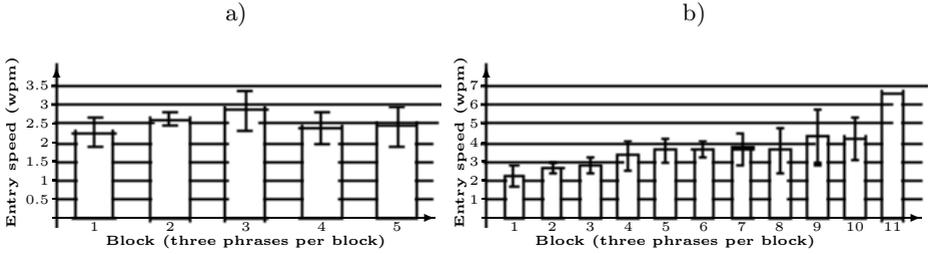


Fig. 3. Entry speed (in wpm) achieved in the evaluation studies: *a)* Pilot study with one participant with FA (using a scan delay of 1000 ms); *b)* usability study with five able-bodied participants (the scan delay was 1000 ms in block 1 and decreased by 50 ms per block, finishing at 500 ms in block 11)

requiring only a fraction of the physical effort (raising the eyebrow versus manual typing). In addition, he proceeded to achieve even higher rates with extended training.

After the successful pilot study, the program was considered ready for evaluation by members of the target population. Two participants with SMA (both not able to enter text manually at all) volunteered for this evaluation. The plan was to explain the program and to have them try the input concept in two initial sessions (one for each) and to have them copy an indetermined number of phrases (as many as they would like) in timed test sessions.

The participants were both able to produce intentional muscle contractions and to use those to operate the program in the individual initial session. Each of them proved this by spelling his first name with the row-column scanning keyboard as well as adding it to the program's dictionary and subsequently retrieving it using the three-phase word selection process detailed above. Unfortunately, due to medical problems which emerged on that day (not caused by the use of *Qanti*), the actual tests had to be canceled.

The participants for the third study were recruited among the students of the Technische Universität Darmstadt. Again, the testing was divided into two parts for each subject. At first, the participants were introduced to the software. Afterwards, they engaged in a session containing up to 11 blocks. Each block consisted of three sentences taken from the phrase set mentioned above. The difference between the successive blocks was the scan delay decreasing by 50 ms per block, starting at 1000 ms. The users' task was to copy the presented sentences and proceed from one block to the next until they were not able to keep up with the scan delay.

Fig. 3b) shows the entry speeds of the participants plotted against the blocks, which corresponds to a decreasing scan delay. The differences between the single sentences within the blocks of the individual participants are not shown – the values in the chart represent an overall average. It is evident in the chart that the entry speed is increasing with decreasing scan delay. Starting at 2.22 wpm

at 1000 ms, the average speed increases to 6.59 wpm at 500 ms (block 11) due to the shorter waiting intervals. At the same time, the standard deviation (represented by the error bars in the figure) is rising with the blocks. This is caused by the increased concentration required. Some participants had problems in concentrating at lower scan delays, while others were able to raise their entry speed even more, because they were able to retain their concentration. In addition to that, the participants aborted the test when they were not able to keep up with the delay. The first participant aborted the test after accomplishing the 700 ms block, the next two after the 650 ms block. The remaining two aborted after 550 ms and 500 ms. So the last block of the chart does not have an error bar, because it relies on the results of one participant only.

Another interesting result is the number of sentences entered correctly. This number increased with the decreasing scan delay at the beginning of the study due to learning effects of the participants (increasing from 2.4 correct sentences at 1000 ms to 3.0 correct sentences at 850 ms). It decreased to 2.0 correct sentences at 650 ms, because the participants had to concentrate more to keep up with the scan delay. At delays between 600 ms and 500 ms, it was increasing again due to the fact that only the most proficient participants were able to continue with short delays.

5 Conclusion and Future Work

A tool for fast but effortless text entry designed for persons with severe physical disabilities has been introduced. The software makes use of *scanning* and thus allows users to enter text with a single input signal – an *intentional muscle contraction*. Entering a word typically involves producing a sequence of three “character keys” – each ambiguously representing about one-third of the alphabet – and selecting the desired candidate from a list of words matching or extending the entered sequence within the program’s *dictionary*.

Empirical data collected in several evaluation studies confirm the practical usefulness of the approach. The resulting entry rates ranged from 2.5 to 6.5 wpm, which is quite competitive for a scanning program, given that *Qanti* is a ready-to-use system supporting error correction and out-of-dictionary words.

The achievable entry rate heavily depends on the dictionary used by the system, so updating this dictionary yields a huge potential for future work. In this respect, it is planned to automatically extract a personalized corpus from text files already written by the user. Another idea for future work is to integrate *Qanti* into higher-level software – for example, the scanning-based telephony application of ref. [11] – as a universal text entry sub-system.

Acknowledgments. This work is partially supported by DFG grant FE 936/3-2 “The AID package – An Alternative Input Device based on intentional muscle contractions”.

References

1. MacKenzie, I.S., Felzer, T.: SAK: Scanning ambiguous keyboard for efficient one-key text entry. *ACM Transactions on Computer-Human Interaction (TOCHI)* (in Press)
2. Felzer, T., Nordmann, R.: Evaluating the hands-free mouse control system: An initial case study. In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) *ICCHP 2008*. LNCS, vol. 5105, pp. 1188–1195. Springer, Heidelberg (2008)
3. Baljko, M., Tam, A.: Motor input assistance: Indirect text entry using one or two keys. In: *Proc. ASSETS 2006*, pp. 18–25. ACM Press, New York (2006)
4. Cardinal, J., Langerman, S.: Designing small keyboards is hard. *Theor. Comput. Sci.* 332(1-3), 405–415 (2005)
5. MacKenzie, I.S., Tanaka-Ishii, K.: Text entry with a small number of buttons. In: MacKenzie, I.S., Tanaka-Ishii, K. (eds.) *Text entry systems: Mobility, accessibility, universality*, pp. 105–121. Morgan Kaufmann, San Francisco (2007)
6. Felzer, T., Strah, B., Nordmann, R.: Automatic and self-paced scanning for alternative text entry. In: *Proc. IASTED Telehealth/AT 2008*, pp. 1–6. IASTED (2008)
7. Simpson, R.C., Koester, H.H.: Adaptive one-switch row-column scanning. *IEEE Transactions on Rehabilitation Engineering* 7(4), 464–473 (1999)
8. Felzer, T., Nordmann, R.: Speeding up hands-free text entry. In: *Proc. CWUAAT 2006*, pp. 27–36. Cambridge University Press, Cambridge (2006)
9. Wandmacher, T., Antoine, J.Y., Poirier, F., Départe, J.P.: Sibylle, an assistive communication system adapting to the context and its user. *ACM Trans. Access. Comput.* 1(1), 1–30 (2008)
10. Belatar, M., Poirier, F.: Text entry for mobile devices and users with severe motor impairments: handiglyph, a primitive shapes based onscreen keyboard. In: *Proc. ASSETS 2008*, pp. 209–216. ACM Press, New York (2008)
11. Felzer, T., Beckerle, P., Rinderknecht, S.: How to make a telephone call when you cannot operate a telephone. In: Langdon, P., Clarkson, J., Robinson, P. (eds.) *DESIGNING INCLUSIVE INTERACTIONS: Inclusive Interactions Between People and Products In Their Contexts Of Use*, pp. 177–186. Springer, Heidelberg (2010)