Use of DUCK Keyboard for a Daily Use

Mathieu Raynal[1] and Philippe Roussille[1]

[1] University of Toulouse – IRIT
Toulouse, France
{firstname.lastname}@irit.fr

Abstract. This article presents the new version of the DUCK keyboard. To evaluate this new prototype, we chose an evaluation with novice users, in two steps: A first step with a copy of sentences. Then we let them get familiarized with our system for 15 days. Finally, we re-evaluate their performance after this phase of familiarization. Results shows that, after a learning phase, the participants significantly improve their input performance with the DUCK keyboard. Text input speed and accuracy have been improved.

Keywords: Soft keyboard, touch screen, smartphone, visually impaired users

1 Introduction

Since the arrival of smartphones, several text entry systems have been proposed to allow blind people to enter text on these devices. Two major categories of systems are opposed: on the one hand, systems based on the Braille alphabet; on the other hand, standard soft keyboards adapted to non-visual interaction. The first Braille input systems were very time-consuming because they required a typing sequence to enter each point corresponding to the Braille character to be typed [1]. With touch screens and the use of multitouch interaction, the input of a character could be improved [2,3]. However, these systems are still accessible to a small number of blind people, because only 10% of them know the Braille alphabet.

The most common input system used by blind people is the standard soft keyboard augmented by a screen reader (well-known systems are Talkback and Voiceover). However, these systems have a major drawback: the user must accurately enter each character. To solve this problem, we have proposed the DUCK keyboard [4]. This keyboard allows to enter different characters of the word imprecisely, then to select the word in a list of deduced words. Our first evaluation showed two main problems [5]: on the one hand, the input of short words; and on the other hand, navigation and selection in the word list. To overcome these two problems, we have proposed different solutions [6,7].

This paper presents the new keyboard that integrates the selected solutions. Here, we have chosen for this new experiment to evaluate the performance of our system with novice users. Then we let them get familiarized with our system for 15 days. Finally, we re-evaluate their performance after this phase of familiarization. After presenting the improvements made to our system compared to the first study, we will present the results obtained with this new system during the two sessions.

2 Improvements to the DUCK Keyboard

Generally, the input of a word with the DUCK keyboard remains the same as initially [4]: the user enters the first character of his word precisely. Then, he continues with approximate strokes and validates the desired word using the list of deducted words.

2.1 Short Word Mode

For words of four letters or less, the user can enter the word with the short word mode: the user enters the first character of the word in the same way as for other words. Then he presses the smartphone screen with two fingers to access the list of short words associated with that first character. The list of short words can contain a maximum of 8 words. To establish the list associated with each character, we chose the 8 words most frequently used in the language. These lists are therefore static. Finally, the proposed interactions for using these lists of short words are the same as for the deduced word lists.

2.2 Interaction with Lists

For all the lists used in the DUCK keyboard (word deduction lists or static short word lists), we have taken the choice presented in [6]: the list presents the words in a linear layout. Selecting a word is done by sliding the finger over it. Lifting the finger off the screen validates the selection. Finally the voice synthesis pronounces the word that is flown over by the user’s finger, and announces the selected word at the time of the final validation.

3 Protocol

We recruited 4 visually impaired users (2 women and 2 men, aged 45 years on average), for this study. These four participants work or are students in a Specialized Education Center for the Visually Impaired. Participants are regular users of smartphones and daily use an onscreen keyboard along with VoiceOver. None of the participants had participated in the first experiment, nor in the study of lists or short words.

For this experiment, participants used a Samsung Galaxy SIII, with the same features and software tools used in the first experiment. Two keyboards were presented to the participants: on the one hand, the DUCK keyboard with improvements presented previously; and on the other hand, the VODKA keyboard as it was already used during the first experiment. VODKA is a VoiceOver like keyboard that proposes the same interactions. The characters layout is the same on both keyboards.

For each exercise, participants would type a set of sentences as quickly as possible. Before starting the first session, each participant could test the system. During this period the participant learns to use the keyboard, to type one or more words, then one or more sentences, according to his wish. When the participant felt comfortable with the system, he had to enter 20 sentences. The exercise stopped after 20 minutes even though he had not finished typing the 20 sentences. For each sentence to be entered, the sentence was read to the participant by speech synthesis. Then, the participant had to type this sentence as quickly as possible. The user could listen again to the sentence to copy, if he wished.

After each word entered, the system checked whether the word entered corresponded to the requested word. If the word was wrong, the user had to re-enter the word until it was correct. In order to simplify the task for the participants, we did not ask them to enter exactly each word: a "homophonic" tolerance of each word was enough (for example, “sear” would be accepted even if “seer” was requested).

In order to evaluate the evolution of the participants' performances during the first uses of the DUCK keyboard, we conducted two evaluation sessions. The two sessions were separated by two weeks. A session consisted of two exercises: one with each system to be tested. Two participants started with the DUCK keyboard before performing the second with the VODKA keyboard. The other two participants did the exercises in reverse order. After the first session, the DUCK keyboard was left for the participants for two weeks. We asked them to daily use the keyboard, to make a written production corresponding to a few lines of text, like an email.

During each exercise, we recorded all the actions performed by the user (presses, releases, and movements of the finger on the screen), the instructions given by the system to the user (instructions of the exercise, sentences to be copied) as well as the lists of words presented to the user during the entry (those produced by the deduction system as well as those used for short words).

4 Results

We detail here the results obtained during the two sessions of our study. We performed Wilcoxon tests as well as Friedmann analyzes to support our results if they do not follow a normal distribution. The significance level (α) of the statistical tests was always set to 0.05.

4.1 Evolution of Text Input Speed during the Two Sessions

First, we looked at the results of the two sessions of the study. At first, we chose to consider only the first session, without training, then the second session, after the user training phase for two weeks.

The graph shown in Fig. 1 gives the text input speeds per session and keyboard. The average text input speed of each keyboard are for the first session of 0.71 characters per second (cps) for DUCK against 0.56 cps for VODKA. The Wilcoxon test showed that these speeds were significantly different ($W=1113$, $p = 0.0478$). At the second session, the participants achieved a text input speed of 1.1 cps with the DUCK keyboard, whereas they only entered an average of 0.61 cps with the VODKA keyboard. The Wilcoxon test showed that these speeds were significantly different (W=859, p = 0. 0364).

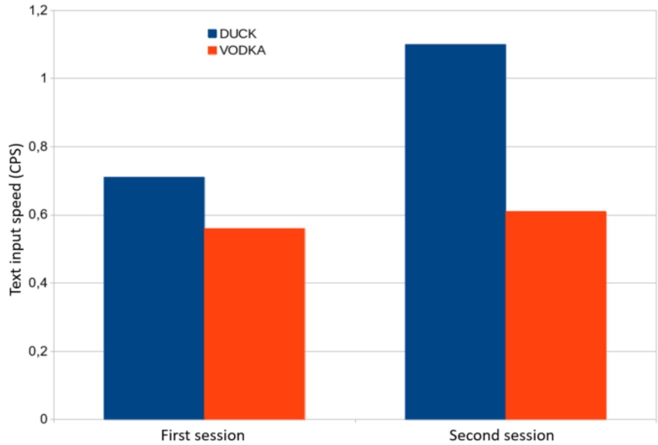

Fig. 1. : Text entry speed of both keyboards during the first and second sessions

On the other hand, we can see an evolution of the input speed with the DUCK system between the two sessions. There was a significant difference between the two sessions (W = 649, p = 0.024).

4.2 Text Entry Speed according to Word Length

Figures 2 and 3 show the mean text input speed according to the length of the word to be typed, respectively in the first and second sessions.

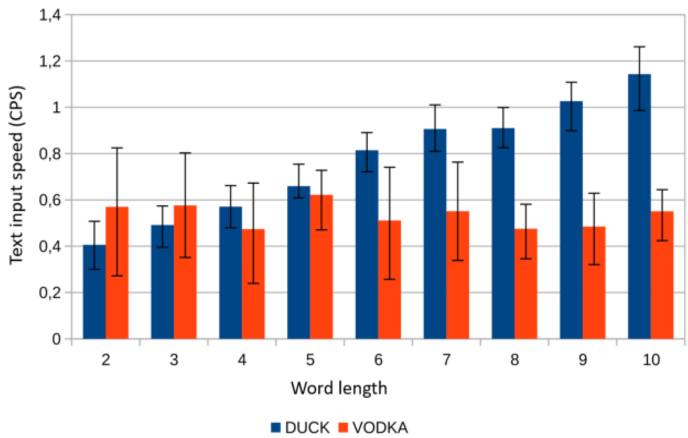

Fig. 2. : Text input speed according to the word length during the first session

In Fig. 2, we can see that DUCK is more efficient than VODKA for words of four or more characters. Moreover, the text input speed is higher for the long words than in the first experiment [5]. By analyzing the input time with DUCK, we can see that the time required to validate a word in the deduction list is only 2.85 seconds on average, whereas it was 3.6 seconds in the first study. This confirms that the changes made to the interaction with the list are beneficial to the validation time, and thus allow the user to enter the words faster.

After the learning phase, during the second session, participants improved their text input speed with the DUCK system, especially for short words (see Fig. 3). There are no significant differences in entering short words between the two keyboards. This can be explained by the good use, by the participants, of the list of short words. Moreover, the decomposition of the input time, with the DUCK keyboard, between the selection phase and the validation phase shows that the users validate the word in 2.07 seconds during the validation phase. Thus, with the learning phase, participants reduced the time needed to validate a word in the deduction list.

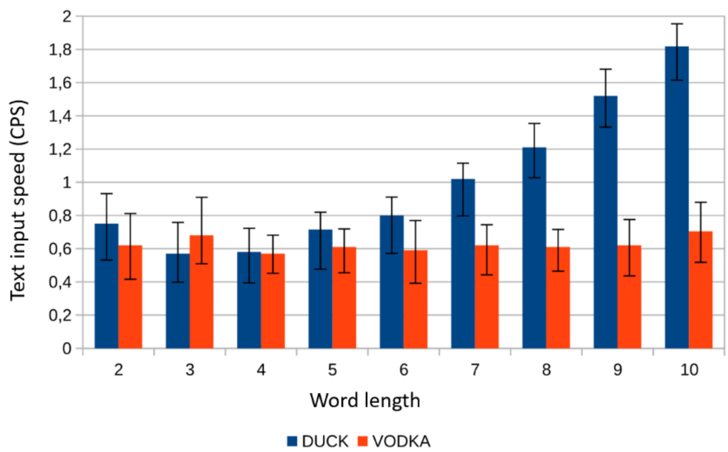

Fig. 3. Text input speed according to the word length during the second session

4.3 Success Rate

To compare the accuracy of each keyboard, it was necessary to introduce the notion of "correct word". A word is defined as a correct word if, and only if, the word phonetically corresponds to the word requested in the sentence but exists as a real word.

To be able to compare the two keyboards, we calculated a success rate by the formula:

$$Success\ Rate = \frac{\text{Number of correct words}}{\text{Number of words entered}}$$

For both keyboards, we obtain the results presented in Fig. 4: for the first session, DUCK has a success rate of 75% against 52% for VODKA. In the second session, the success rate is 91% for DUCK against 77% for VODKA.

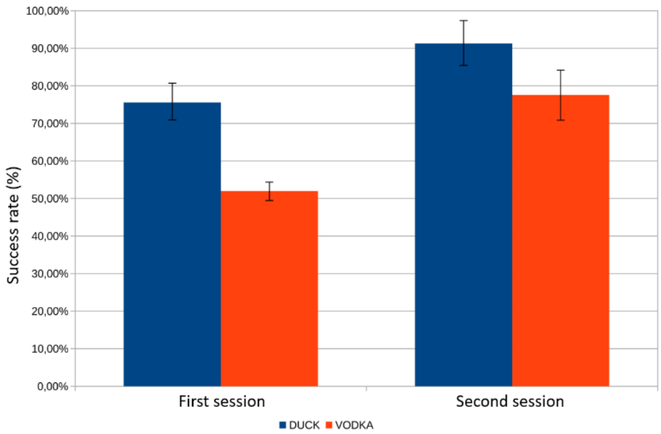

Fig. 4. Success rate of both keyboards during the first and second sessions

A Wilcoxon test indicates that there is a significant difference between precision with the DUCK keyboard and with the VODKA keyboard in the first session (W= 1070, p=0.036), as well as for the second session (W=2069, p=0.032). In addition, we can also observe that for both systems, the success rate increases between the two sessions. This observation is confirmed by Wilcoxon tests which confirm a significant difference between the two sessions and for both keyboards (W=1208, p=0.0487 for DUCK and W=276, p = 0.0124 for VODKA).

Given these observations, we can conclude that DUCK is more efficient than VODKA in word correction: the number of correctly entered words is greater with DUCK than with VODKA.

4.4 Use of Short Words

Finally, we were interested in using the short word mode throughout the experiment. Our first hypothesis was to consider that the activation of the short word mode allowed the user to accelerate his input.

In order to evaluate the impact of using the short word mode, we counted the number of uses of the short word mode. We split into two use cases: a) the "useful" use, when the user must really type a short word; b) the "useless" use, when the user activates this mode by mistake. In this case, we do not distinguish between cases when the user has consciously activated the short word mode by mistake (The user thinks to find the word in the list, but this one is not there) and when it was an involuntary movement on the screen.

To evaluate the use of the short word mode, we calculated the rate of useful use (RUF) of the short word mode by the formula:

$$\text{RUF} = \frac{\textit{number of useful uses of short word mode}}{\textit{number of short words}}$$

"Number of useful uses of short word mode" is the number of times the user uses this mode correctly, and "Number of short words" is the number of times the user should have used this mode.

Similarly, we also calculated the rate of useless use (RUL) of the short word mode by:

$$\text{RUL} = \frac{\textit{number of useless uses of short word mode}}{\textit{number of words}}$$

"Number of useless uses of short word mode" is the number of times the user uses this mode by mistake, and "Number of long words" is the number of words that are not in the short word lists.

We present the results we obtained in Fig. 5. The RUF is 43% for the first session, and 68% for the second session. A Wilcoxon test confirms the significant difference between the two sessions of the useful use of the short word mode (W=511, p=0.036). This is explained by a better knowledge by the users of the contents of the lists of short words during the second session.

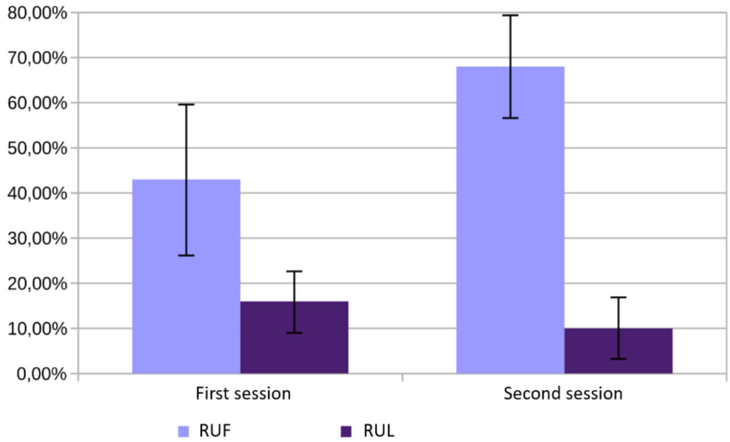

Fig. 5. Use of short word mode during the first and second sessions

Regarding the RUL, participants activated by mistake the short word mode for 16% of the words in the first session and 10% in the second session. However, this decrease is not significant (W=421, p=0.280). The participants explained at the end of the second session that the useless use of the short word mode were largely due to a problem of interaction, most often due to an uncontrolled finger movement.

5 Discussion

The aim of this experiment was to evaluate the DUCK keyboard integrating the improvements made to the use of short words and the validation of words in the deduction lists. We chose to carry out this experimentation in two sessions separated by a learning phase. This allowed us to see the results of novice users, and then once they took control of the system for two weeks.

Overall, we can see that improvements to DUCK have been beneficial. Indeed, the system is more efficient than VODKA from the first use. The participants thus obtained a significantly higher input speed with DUCK than with VODKA in the first session. Similarly, they made fewer text entry errors with DUCK.

Through this evaluation, we were able to observe the improvement of the performances concerning the validation of the word in the deduction list. Indeed, during the first evaluation, this time was 3.5s to validate a word [5]. In the first session of this experiment, the mean validation time was only 2.85s to validate a word in the deduction list. This time was further reduced during the second session to 2.07s. This reduction in validation time has made it possible to increase the user input speed with DUCK. Participants thus have higher input speeds with DUCK for words of 4 or more characters in the first session, compared to 6 characters or more in the first experiment.

In addition, the participants also assimilated the short word mode. If they only used it in 40% of the cases during the first session, they used it in two-thirds of the cases during the second session. This good use of the short word mode allows users to enter faster with DUCK, for all words regardless of their length.

Finally, we can see that after a learning phase, the participants significantly improve their input performance with the DUCK keyboard. Text input speed and accuracy have been improved. With a longer use, participants thus find advantages in using this system rather than the ones they already use.

References

1. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: Brailletype: Unleashing braille over touch screen mobile phones. INTERACT 2011, Springer-Verlag, pp. 100–107 (2011).
2. Mascetti, S., Bernareggi, C., Belotti, M.: Typeinbraille: Quick eyes-free typing on smartphones. ICCHP 2012, Springer-Verlag, pp. 615–622 (2012).
3. Nicolau, H., Montague, K., Guerreiro, T., Guerreiro, J., Hanson, V. L.: B#: chord-based correction for multitouch braille input. CHI 2014, ACM, pp. 1705-1708 (2014).
4. Roussille, P., Raynal, M., Kammoun, S., Dubois, E., Jouffrais, C.: DUCK: a deDUCtive Keyboard. In 3rd International Workshop on Mobile Accessibility, pp. 1-5 (2013)
5. Raynal, M., Roussille, P.: DUCK: A DeDUCtive Soft Keyboard for Visually Impaired Users. In Studies in health technology and informatics (242), pp. 902-909, (2017)
6. Roussille, P., Raynal, M.: LOVIE: A Word List Optimized for Visually Impaired UsErs on Smartphones. UAHCI 2016, pp. 185-197 (2016).
7. Roussille, P., Raynal, M.: SWIFT: A Short Word Solution for Fast Typing. ICCHP 2014, Springer International Publishing, pp. 464-471 (2016).