PAELife

Élvio Manuel Teixeira Rodrigues

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Examination Committee

Chairperson: Prof. José Carlos Martins Delgado
Supervisor: Prof. Daniel Jorge Viegas Gonçalves
Member of the Committee: Prof. Tiago João Vieira Guerreiro

October 2013

Acknowledgments

I would like to thank Professor Daniel Gonçalves for guidance along the way.

I would like to thank Micael Carreira for the outstanding teamwork on the PAELife project. This thesis benefited highly with your input and observations.

I would like to thank everyone at room 529 (Micael Carreira, Gabriel Barata, Ricardo Dias and Pedro Pascoal) for the shared tips and knowledge. Being in that room somehow made the whole experience more enjoyable.

I would like to thank my parents, José Rodrigues and Maria de Lurdes, and my brother Gabriel Rodrigues, for the support and for always believing in me, even when I don't.

I would also like to thank several Institutions and the professionals that represent them, for enabling us to evaluate the developed keyboards with older adults. This includes Helena Sobral of Centro de Dia da Junta de Freguesia Olivais, Ana Antunes of Centro Social e Paroquial Nossa Senhora da Conceição, Nélia Ferreira of Centro de Dia da Fundação Cardeal Cerejeira and Mónica Santos of Centro Social Paroquial de São Jorge de Arroios. I would also like to thank all participants for their availability to perform the tests.

This work was supported by FCT (INESC-ID multiannual funding) under project PEst-OE/EEI/LA0021/2013 and the project PAELife, reference AAL/0014/2009.

Abstract

Touchscreen interfaces are increasingly more popular. However, they lack haptic feedback, making it harder to perform certain tasks. This is the case of text-entry, where users have to constantly select one of many small targets. This problem particularly affects older users, whose deteriorating physical and cognitive conditions, combined with their unfamiliarity with technology, can discourage them from using touch devices. On a first phase we developed a baseline QWERTY keyboard and five different variants. Two of the variants, *Color* and *Width*, used a letter prediction algorithm to highlight the four most probable keys by changing the color and width of the keys, respectively. The *Predict Words* variant used a prediction algorithm to suggest the four most probable words for the written prefix. The *Shifted* and *Size Invisible* variants aimed to correct neighbor substitutions by shifting the touch points to the top and the left, and by increasing the underlying area of the four most probable keys, respectively. These keyboards were tested on a baseline study with 20 regular participants in order to analyze their performance when inputting text on a tablet. Afterwards we performed a study with 20 older adults, with the most promising variants of the previous study. From the older adults study we learned more about their typing behavior, and therefore created four new variants to be used in a simulation study along with *Shifted* and *Size Invisible*. One of the variants downgrades the baseline QWERTY keyboard into a single touch keyboard, while the others reject interactions based on time features. Results show that visual changes should be kept to a minimum; touch points should be shifted upward and to the opposite side of the hand used to type; single touch keyboards perform better than multi-touch; and omitting keys below a certain time threshold minimizes accidental insertions.

Keywords

Older adults, Touchscreen, Text-entry, Tablet, Pre-Attentive Interface, Accessibility

Resumo

Os dispositivos multi-toque são cada vez mais populares. No entanto, a falta de *feedback* háptico, torna difícil a realização de determinadas tarefas. É o caso da tarefa de introdução de texto, na qual os utilizadores têm de seleccionar constantemente um de muitos alvos. Este problema afecta particularmente os idosos, uma vez que a deterioração da sua condição física e cognitiva em conjunção com a sua falta de familiaridade com a tecnologia, pode dissuadi-los de usar estes dispositivos. Numa primeira fase, desenvolvemos um teclado virtual QWERTY (base) e cinco variantes. Duas destas variantes, *Color* e *Width*, usam um algoritmo de predição de letra que destaca as quatro teclas mais prováveis, mudando a sua cor e largura, respectivamente. A variante *Predict Words* usa também um algoritmo de predição para sugerir as quatro palavras mais prováveis para o prefixo escrito. As variantes *Shifted* e *Size Invisible* têm como objectivo corrigir os erros de substituição de vizinhança através de um *shift* compensatório para o topo e para a esquerda, e através do aumento da área subjacente das quatro teclas mais prováveis, respectivamente. Estes teclados foram avaliados num estudo inicial que contou com 20 utilizadores, com o intuito de analisar a sua *performance* ao introduzir texto. Posteriormente, realizámos outro estudo com 20 idosos, com as variantes mais promissoras do estudo anterior. Neste estudo aprendemos mais sobre a forma de interacção dos idosos, o que nos permitiu criar quatro novas variantes a ser usadas num estudo de simulação, em conjunção com as variantes *Shifted* e *Size Invisible*. Uma das variantes comporta-se exactamente igual ao teclado QWERTY base, mas apenas reconhece um ponto de contacto, enquanto outras rejeitam interacções baseadas em características temporais. Os resultados mostram que as mudanças visuais devem ser o mais reduzidas possível; os pontos de contacto devem ser transladados para o topo e para o lado oposto da mão usada para interagir; teclados que apenas aceitam um ponto de contacto em simultâneo comportam-se melhor que os multi-toque; e, omitir interacções abaixo de um determinado limiar temporal reduz as inserções acidentais.

Palavras Chave

Idosos, Ecrã Táctil, Introdução de texto, Tablet, Interfaces Pré-Atentas, Acessibilidade

Contents

List of Figures

List of Tables

1

Introduction

Several studies have presented evidence indicating that the population is aging across the world [1]. This is due to the fact that, because of our healthier lifestyles, we live longer, and therefore are more likely to be physically and cognitively active until older ages. Still, as we age, we experience changes on several dimensions. This includes perceptual, psychomotor, cognitive, physical, psychological and social changes [2–6]. Besides these changes, older adults are also more likely to suffer from several diseases that also debilitate their capacities. Still, this dissertation is focused only on healthier older adults. Therefore, older adults with Parkinson, Alzheimer, Osteoporosis, among other diseases, are out of the scope of this thesis.

At the same time studies show that touchscreen devices are widely used worldwide, with an increased tendency to grow[1]. There is also some evidence that shows that, older users, in particular, benefit with the use of such devices, since it allows them to interact more easily and direct with digital content [7–13]. In this context it is expected that these kind of devices will be increasedly adopted by older adults. This is an opportunity to develop more inclusive interfaces for older adults, since these kind of devices rely much less on physical keys, to rely more on elements controlled by software. However, touch devices have also the disadvantage of lacking the haptic feedback of physical buttons, which makes it harder to accurately select targets.

This characteristic particularly hinders certain tasks, such as text-entry, on which the user has to constantly select one of many small targets. Furthermore, this task is of great importance since it is transversal to many applications such as, basic communications, managing contacts, editing documents, note taking, web browsing, searching, among others [14]. Although several studies focus on virtual keyboards for smartphones and tablets for young users, not much research has been performed regarding older adults. Since these new and updated technologies are often designed for the younger generations, who are familiar with using such technologies, there is a need to understand what are the special needs of the older adults. These needs include not only the fact that older adults are generally less experienced with technology, but also the fact that aging implies several changes that might limit their capacity to interact with such devices, if not taken into account. Furthermore, the literature is more focused on smartphones than tablet devices. Even though tablets can overcome some of the smartphone's problems, such as "buttons being too small", they are also touch devices, and therefore share several problems such as the lack of haptic feedback. This is why we decided to focus our research on understanding and developing more adequate solutions to improve typing performance of older adults on virtual keyboards for tablet devices.

---

[1] http://www.comscoredatamine.com/2012/03/exponential-tablet-adoption-in-2011-ushers-in-era-of-convergent-consumption/

1.1 Objectives

This dissertation is focused on the multi-touch modality, specifically on text-entry. Therefore, our main goal is to *thoroughly understand older adults touch typing behavior, in order to develop text-entry solutions more adequate to their needs, which will enhance their typing performance*.

This dissertation is a part of the PAELife project[2]. The project itself aims to fight older adults' isolation, exclusion and to allow them to be more productive, independent and to have a more social and fulfilling life, by empowering them with a Personal (Virtual) Life Assistant (PLA) - a virtual presence that supports social communication, learning and entertainment. The PLA will include three interaction modalities: gesture recognition (kinect), multi-touch (Tablet) and speech recognition (SR), which are expected to bring additional value to older adults, since the weaknesses of one modality can be counterbalanced by the strengths of another, resulting in an overall increased usability and accessibility.

1.2 Approach and Overview

We started by investigating the related work. We were particularly interested in studies that provided knowledge regarding users' touch patterns when acquiring targets in general, and on text-entry tasks in particular. We were also interested in virtual keyboard solutions that improved the typing performance of older adults. However, works focusing on older adults and virtual keyboards were limited. Therefore, we also took into account studies that aimed to improve typing performance for users in general.

As a result from the literature review, we developed a traditional QWERTY keyboard and five variants. Two of the variants, *Color* and *Width*, used a letter prediction algorithm to highlight the four most probable keys by changing the color and width of the keys, respectively. The *Predict Words* variant used a prediction algorithm to suggest the four most probable words for the written prefix. The *Shifted* and *Size Invisible* variants aimed to correct neighbor substitutions by shifting the touch points to the top and the left, and by increasing the underlying area of the four most probable keys, respectively.

On a first approach we tested these keyboards on a baseline study with 20 regular participants, since these were easier to find. Users were asked to enter sets of individual sentences using the different keyboards, while metrics such as *Words Per Minute* (WPM) and *Minimum String Distance* (MSD) were collected. Afterwards, on a second approach, we performed a similar study with 20 older adults, with the most promising variants of the previous study. From the older adults study we learned more about their typing behavior, and therefore created four new variants to be used in a simulation study along with *Shifted* and *Size Invisible* variants. One of the new variants downgrades the baseline QWERTY keyboard into a single touch keyboard (*Single Touch* variant), while the others reject interactions based on time features (*Intra-key*, *Inter-key* and *Combined Timed* variants).

---

[2] http://www.microsoft.com/portugal/mldc/paelife/

Results show that changing visual elements (color and size) of the keys to focus users' attention on the most probable keys, can have a negative impact on typing speed, and thus should be avoided. We verified the bottom-right touch typing pattern described by Nicolau [14] regarding older adults that only use their right hand. Taking advantage of this knowledge enabled us to reduce significantly *neighbor substitutions* (58.62%), *failed omissions* (100%) and *slide omissions* (39.39%) with the *Shifted* variant. Furthermore, we found that vertical shifts increase gradually from row to row until we reach the space bar. We also found that the horizontal shift pattern is closely related with the hand the user is using to type as hypothesized by Nicolau [14]; that is, the shift is more intense towards the side of the keyboard of the hand the user is using to type. A result that, to our knowledge, has never been reported by any other author is that, when users perform a vertical slide between keys of subsequent rows (up or down), 96.4% of the times, the user intends to tap the key from the row above. This knowledge can be used to reduce *slide omissions* even more. We also found that, in general, single touch keyboards are more adequate for older adult users (corrected 78.77% of *extra-finger insertions*). Finally, we also found that omitting keys below a certain time threshold (*Inter-key Timed* variant) can minimize *double insertion* errors (85.19%).

1.3 Contributions

Our main contribution is a thorough understanding of older adults' touch typing behavior on virtual keyboards for tablet devices. This knowledge will help on the development of virtual keyboards that are more accessible and more adequate for the older adults needs. Our research contribution includes:

- **Understanding the touch typing behavior of regular users.** Our baseline study allowed us to determine the most promising variants and gather knowledge to be used in the subsequent study. This knowledge can also be applied to develop text-entry solutions for regular users.
- **Understanding the touch typing behavior of older adult users.** Our main study allowed us to determine the most promising variants specifically for the older adults, and thoroughly understand their touch typing behavior.
- **Development of new variants based on the gathered knowledge from previous studies.** Our final study uses the gathered knowledge from previous studies to develop virtual keyboards that are more accessible and more adequate for older adults needs.
- **Understanding differences and similarities between the two user groups.** Although implicitly, it is easily extracted from this document.

1.4 Publications

At the time of this writing, our research has resulted in three papers, two of which were accepted for publication and one is currently undergoing peer review:

1. *Rodrigues, É., Carreira, M. and Gonçalves, D.* **Improving Text-entry Performance on Tablet Devices.** (accepted - Interacção 2013 [3])

---

[3] http://interacao2013.utad.pt/

2. *Rodrigues, É., Carreira, M. and Gonçalves, D.* **Developing a Multimodal Interface for the Elderly.** (accepted - DSAI 2013 [4])
3. *Rodrigues, É., Carreira, M. and Gonçalves, D.* **Improving Text-entry Experience for the Elderly on Tablets.** (under review - CHI 2014 [5])

A fourth paper is being written to be submitted to *Universal Access in the Information Society* journal.

1.5 Document Structure

The remainder of this dissertation is organized in seven chapters. Chapter 2 starts by presenting demographic facts regarding an aging population. It then presents several changes that come with aging.

Chapter 3 presents some works that try to understand how people interact with touch screen devices in general, and then particularly with virtual keyboards. A discussion is presented that leads to several design recommendations.

Chapter 4 presents and explains the proposed architecture for the keyboards to be used during the studies. It also presents an evaluation of the prediction system and a flowchart for each of the variants.

Chapter 5 presents a user study we conducted with 20 regular participants and 6 of the developed virtual keyboards. We discuss WPM, error rate and typing errors for each of the variants. We also investigate touch typing behavior for this kind of user.

Chapter 6 presents a user study and a simulation study. The user study was conducted with 20 older adults participants and three of the developed virtual keyboards. The simulation study uses the QWERTY input data from the user study to simulate 6 different variants. We discuss WPM, error rate, typing errors and typing behavior for each of the variants of the user study, and error rate and typing errors for the simulation study. Design implications are presented at the end of the chapter summing the gathered knowledge.

In Chapter 7, we conclude by presenting the major results and an overall discussion of the key findings of our research. Additionally, we present several possibilities regarding future work.

---

[4] http://dsai2013.utad.pt/

[5] http://chi2014.acm.org/

2

Understanding Older Adults

This chapter focuses on older adults, especially in understanding the perceptual, psychomotor, cognitive, physical, psychological and social changes that take place as we age. Most developed world countries have accepted the chronological age of 65 years as a definition of 'elderly' or older person [15]. That's the definition that is considered in this chapter.

Some studies reveal that the population is aging across the world. The number of people aged 65 or older is projected to grow from an estimated 524 million in 2010 to nearly 1.5 billion in 2050 [1]. At the same time technology usage has been increasing over the years, in several different areas, and will most-likely continue to grow. Therefore, it is important for interface designers and developers to take this data into account, and try to understand the capabilities and limitations of older adults for guidance to create more usable technologies.

2.1 Age-related Changes

To better understand the capabilities and limitations of older adults, we will take a closer look at the changes that occur to individuals when they age. These include perceptual, psychomotor, cognitive, physical, psychological and social changes.

2.1.1 Perceptual

Perceptual changes refer to changes associated with a decline in functional abilities, most commonly, vision and hearing.

2.1.1.1 Vision

All individuals will eventually experience some kind of visual impairment, varying in degree from person to person [2]. Additionally, as we get older there are a number of conditions whose presence causes a marked loss of vision in older adults. These include age-related macular degeneration (AMD), glaucoma, cataracts and diabetic retinopathy [16]. Firstly, AMD results in a loss of vision in the center of the visual field, which can make it difficult or impossible to read, recognize faces and drive, although enough peripheral vision remains to allow other activities of daily life. Next, glaucoma's most common consequence is the reduction of peripheral vision although it can also affect motion [17] and color perception, contrast sensitivity [18] and central vision acuity, which can make it difficult to walk, drive, avoid obstacles, etc. Cataracts are characterized by a clouding of the crystalline lens of the eye, which cause visual acuity and contrast sensitivity loss [19]. Finally, diabetic retinopathy can develop in anyone who has diabetes type I or II. It results in a decreased visual acuity, which can make it difficult to read and drive, especially at night [20].

2.1.1.2 Hearing

Approximately half of all men over the age of 65 and 30% of women suffer hearing loss [3]. There are three types of hearing impairment: conductive hearing loss, sensorineural hearing loss and mixed hearing loss [21]. First, conductive hearing loss is a reduction in the ability of sound to be transmitted to the middle ear. This can involve cerumen (wax) buildup or stiffening of the tiny ossicles in the middle ear [21]. Next, sensorineural hearing loss is a hearing loss resulting from damage to any part of the inner ear or the neural pathways to the brain. It can result from genetic causes or from systemic disease, ototoxic substances, or prolonged exposure to loud noise. The most common form of this type of hearing loss is called presbycusis and can be provoked by the continued use of drugs such as aspirin and antibiotics [22]. Patients with this type of hearing loss have an inability to distinguish high-frequency tones, which makes it difficult to comprehend speech. Finally, mixed hearing loss involves both conductive and sensorineural hearing loss [22, 23].

2.1.2 Psychomotor

Some changes occur that affect how older adults carry out movement. Research has shown that older adults take 30% to 70% longer than their younger counterparts to perform certain motor tasks [4], and are also less accurate in performing those movements [2]. Also, the sensation of touch is affected by changes to central and peripheral nervous systems. There's a reduced ability in detecting vibrotactile stimulation, perceiving difference in temperature [24], and noticing light pressure touches. Tactile acuity also suffers significant declines, with bodily extremities being the most affected body parts [25].

It is widely known that indirect input devices like a mouse are cognitively more demanding than direct input devices like touchscreens [26]. Indirect input devices also require fine motor skills [26]. Therefore, using direct input instead of indirect input might help older users to surpass some of the psychomotor difficulties related with aging.

2.1.3 Cognitive

Cognitive changes also have a significant influence for older users. Age-related differences in cognitive functioning can be seen to stem from the reduction of cognitive resources available, impairing older adults' ability to carry out cognitively demanding processes [27]. Particularly relevant to UI design are cognitive abilities such as memory, attention and spatial cognition.

2.1.3.1 Memory

There are several distinct types of memory, which are affected differently by age. First we have working memory which is especially affected by aging. It involves the active manipulation of information that is currently being maintained in focal attention, which is the reason why the decline of this type of memory can negatively impact several daily activities. Working memory differs from short-term memory in the sense that the latter only involves the simple maintenance of information over a short-period of time.

Opposed to this one, there's long-term memory which refers to a more permanent form of storing information. Into this category falls sematic memory which refers to one's general knowledge about the world. Such information is not tied to the space or time of learning and its retrieval is generally prefaced with ”I know”. Although access to information may be somewhat slower (particularly for words and names) the organization of the knowledge system seems unchanged with age. Prospective memory also falls into this category. Much of what we have to remember in everyday life involves prospective memory, and can be divided in two types: event-based and time-based prospective memory. Age-related declines in prospective memory are usually greater for time-based tasks rather than event-based ones [3]. Finally, we have Procedural memory which also falls into this category, and it refers to knowledge of skills and procedures such as riding a bicycle, playing the piano, or reading a book. Once acquired, procedural memories are expressed rather automatically in performance and are not amenable to description. In general, older adults have normal acquisition of procedural skills in both motor and cognitive domains and retain them across lifespan. However, learning new procedural tasks and developing new automatic processes is difficult for older adults [3], which is why it is advisable to make use of older adults crystallized knowledge when designing interfaces, by building upon previously learned mental-models and procedures [3].

2.1.3.2 Attention

Attention is a basic but complex cognitive process that has multiple sub-processes specialized for different aspects of attentional processing. First we have selective attention which refers to the ability to attend to some stimuli while disregarding others that are irrelevant to the task at hand. This is generally more difficult for older adults as they seem to have more trouble in concentrating on one factor while ignoring other distracting stimuli [2]. This is of high importance when designing interfaces, because selectivity is significantly tied to visual search capabilities as this consists of visually identifying a target among distracting stimuli [28]. Second, we have divided attention which requires the processing of two or more sources of information or the performance of two or more tasks at the same time. It is usually associated with significant age-related declines in performance [29]]. Similarly, the performance of older adults is slowed to a greater degree than that of young adults when attention must be switched from one task to another, requiring a change of mental set [29, 30]. Finally, sustained attention refers to the ability to maintain concentration on a task over an extended period of time. This can be an issue for older adults because research has shown that they are more susceptible to be distracted by surrounding stimuli that are irrelevant to the task at hand [3, 28].

Pak and McLaughlin [2] suggest that avoiding clutter and removing unnecessary information, while drawing attention to important items or frequently performed actions might be a form of avoiding attentional issues when designing interfaces for older adults.

2.1.3.3 Spatial Cognition

Spatial cognition is concerned with the acquisition, organization, utilization, and revision of knowledge about spatial environments. This skill is particularly important to human interaction with systems, given that it is responsible for the ability to construct mental models [2]. A mental model is a model of what users know (or think they know) about a system [31]. Ruotolo et al [32] found that aging doesn't seem to have a uniform effect on spatial abilities. Some abilities are well preserved, such as perceptual discrimination and retrieval of metric distance. Some others are instead impaired, such as the ability to mentally rotate visual images and retrieve spatio-temporal sequences [32], which means that older adults might have more difficulties in performing tasks that involve navigating information hierarchies [3].

2.1.4 Physical

Older adults show changes in their physical abilities, due to loss of muscle mass, flexibility and the emergence of age-related conditions such as arthritis and stroke [5]. These changes provoke an overall slowness of movement making it also harder to make precise selections on small interface targets [33]. To reduce these problems, designers should implement large targets for accurate cursor positioning, reduce scrolling when possible and allow for slower response times.

2.1.5 Psychological and Social Changes

All of the changes discussed until now, especially the physical changes, can force older adults to participate less in social interactions. The perception of limited time can also hinder their interaction with other persons, because they might feel that they don't have enough time to work on new relationships [6]. Also, there's a change in the perception of themselves, which can be negative if they don't accept the changes that come with aging.

3

Understanding Touch Interaction

In this chapter we will present some works that try to understand how people interact with touch screen devices in general, and with virtual keyboards in particular. First we will analyze the kind of gestures non-technical users find adequate for certain actions. Then, we will analyze what are the restrictions regarding target size/position/technique that lead to better performance. Finally, we will also analyze current virtual keyboards and touch typing behavior of users.

Although the main focus was to understand the specificities regarding the older adults, we decided to broaden the research since there are not enough works focusing solely on them. Therefore, we also included works that focused on younger adults without any type of impairments, as well as works that focused on persons with motor-impairments. Although the needs of both groups of users are different than older adults' needs, some results can be extrapolated and applied to the older adults' contexts.

3.1 Eliciting Gestures

Wobbrock et al. [34] presented an approach to designing tabletop gestures that relies on eliciting gestures from non-technical users by first portraying the effects of a gesture, and then asking users to perform its cause. Some commands elicited little gestural agreement, suggesting the need of on-screen widgets. However, participants still exhibited a substantial level of agreement in making their gestures. The user-defined gesture set differs from sets proposed in the literature by allowing flexibility in the number of fingers that can be used, rather than binding specific number of fingers to specific actions. It further differs from prior surface systems by providing multiple gestures for the same commands, which enhances guessability. Users demonstrated preference for one handed gestures.

In the sequence of the previous study, Wobbrock et al. [35] conducted another study to compare the gesture set obtained in that study [34] (end-user elicitation method) with a set of gestures authored by three HCI researchers. Participants, who didn't know the gestures' authorship, evaluated 81 gestures presented and performed on a Microsoft Surface. Participants preferred gestures that were physically easier to perform and/or demanded less cognitive effort. For instance, one-handed gestures were preferred to two-handed, and gestures using only a single finger were preferred to those using multiple fingers or an entire hand. Conceptually simpler gestures (i.e., based on physical analogies rather than abstract mappings) were also preferred.

Stöbel [36] conducted a series of studies, one of which we will point out (Study 2B [36]). For each task, participants were shown four possible gestures, which they had to rate according to their perceived suitability for the task (Figure 3.1). The four gestures that were presented for each task had been selected according to the following criteria, based on the results of previous studies: the set must contain the two most frequently suggested gestures by the younger group, the two most frequently suggested gestures from the older group, and the overall top three most frequently suggested gestures. Older adults chose manipulative gestures over iconic and symbolic gestures. An example of this is the choice between an arrow gesture, and a swipe gesture. When given the choice between both gestures, the swipe gesture was preferred. Although the results of the final studies show that there's a strong preference for manipulative and indexical gestures across all age-groups, it's important to notice that older users are more willing to trade efficiency for familiarity, and as such are more likely to accept symbolic and iconic gestures than the younger group. Older adults are also less ready to perform multi-finger gestures, but, on the other hand, are more tolerant to gestures that are slightly more complex. Results also show that alphanumerical gestures were perceived as little suitable by older users. There are also considerable differences between age groups regarding the question which gestures are most suitable for a certain task (verified in 50% of the tasks). Moreover, it still proves difficult to establish a gesture set which is particularly suited to older users.

Figure 3.1: Experimental setup: Task descriptions are displayed on the upper monitor, gesture videos on the lower.

3.2 Performance of Gestures

Kobayashi et al. [7] conducted a study to assess standard mobile touchscreen interfaces for older adults. The tasks included: (1) performing basic gestures such as taps, drags, and pinching motions for which the authors measured task completion times, analyzed their behaviors while making the motions, and asked about the users' preferences; (2) using basic interactive components such as software keyboards and photo viewers, for which the authors simply observed their behaviors and asked for user comments. Participants performed the tasks in two devices: the ”large” device (iPad) and the ”small” device (iPod Touch). Results show that the larger screen outperformed the smaller screen for tapping, dragging and pinching, even though these last two required more than twice the amount of finger movement on the screen. It was found that mobile touchscreens were generally easy for older adults to use and a week's experience generally improved their proficiency. It was also found that dragging and pinching are more comfortable than tapping. However, older adults tend to miss their intended targets due to parallax and the large contact area of each finger.

This problem is also related with the fact that, when using touchscreens, users do not see where they touch and cannot feel the position of virtual keys and buttons. Taking this into account, Henze et al. [37], analyzed the touch behavior of smartphone users. Using a game published to the Android Market they collected more than 120 million touch events from 91,731 installations. The gameplay (Figure 3.2) consists of targets that are presented to the player and the task is to touch these targets. If a player successfully hits a target, it disappears. If a target has not successfully been hit in a certain time frame it is counted as a miss. The authors decided to use circles as targets as these have the same diameter in all directions and thus allow easier comparison of different target sizes. They used the data to determine the error rate for different target sizes and screen positions. After analyzing the data, the authors got to the conclusion that events were systematically skewed towards a position in the lower-right screen. Based on this finding they trained a function that shifts the touch events to reduce the number of errors. The function reduced the error-rate by 7.79%.

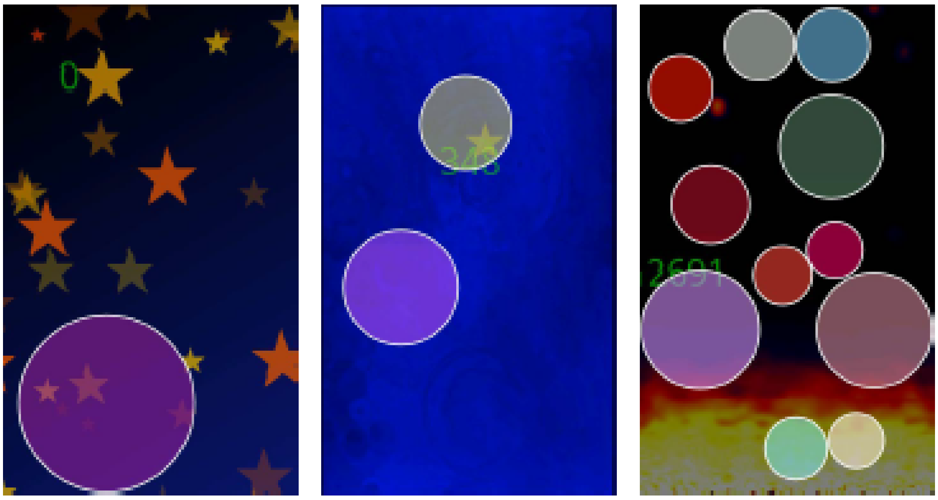


Figure 3.2: Screenshots of the game.

Wacharamanotham et al. [38] explored a new interaction technique (swabbing) for older adults who suffer from hand tremor, since this group has difficulties interacting with touchscreens because of finger oscillation. Swabbing is based on the fact that sliding one's finger across the screen may help reduce this oscillation. The technique works as follow: the user touches any area on the screen and slides his finger towards the target placed on the edge of the screen. After the finger moved beyond a distance threshold, a linear regression is calculated from recent touch coordinates to determine the intended target, which is then highlighted. To select the target, the user either lifts his finger or slides it across the target and beyond the screen (Figure 3.3). To cancel the highlighting, the user slides the finger backward. In the study, participants were asked to perform selections with tapping and swabbing techniques with 16, 25 and 36 possible targets on screen. The number of selection errors was recorded and touch movements (landing, sliding and lifting) were timed stamped. Results show that, although swabbing is inherently slower than tapping, participants were satisfied with this input method. Users with tremor prefer more accurate input methods to faster ones. The positive impact of swabbing appeared to be strongest for the 16 targets layout.

In a similar study, Wacharamanotham et al. [39] found that swabbing is able to reduce significantly the error rate when compared to tapping. Even though swabbing was a completely new technique to the participants, its acceptance was at least as high as tapping's.

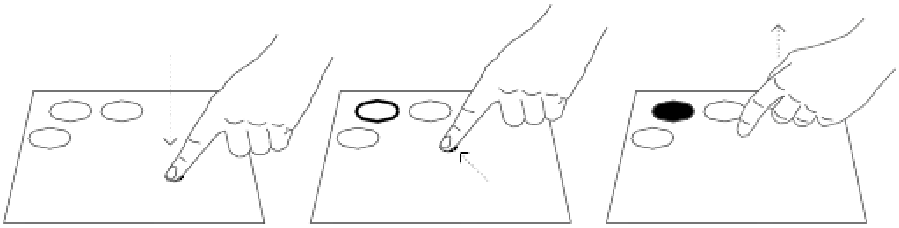


Figure 3.3: The three steps of swabbing: touch, slide towards the target, and lift.

3.2.1 Young vs Older Adults

Stöbel [36, 40] conducted a study to understand if gesture-based interaction facilitates technology interaction, especially with regard to older users, or whether it further decreases accessibility and usability of such devices for this user group. In the study the author compares younger and older users on a set of 42 simple gestures with varying complexity. The results indicate that older users might be slower, but not necessarily less accurate performing simple gestures on a touchscreen. Moreover, factors, such as for example the familiarity of certain gesture patterns, can influence performance, and older adults seem to benefit especially from familiar patterns.

Jin et al. [41] conducted a study that investigated the optimal button size and spacing for touchscreen-based user interfaces for older adults. Two experiments were performed: the first one consisted of a target, randomized between 9 different sizes, which appeared in a random position within a designated 160 mm x 160 mm area at the center of the screen; the second one consisted of a 3 x 3 matrix, from which participants had to select a button that matched the button that appeared at the top of the screen. For both tests, subjects were asked to touch the target button as quickly and as accurately as possible. The results showed that older adults have shorter reaction times with larger touch-sensitive buttons. It also showed that larger spaces between touch-sensitive buttons don't improve the touchscreen performance. Results also showed that manual dexterity will not significantly affect the performance of touching an isolated button on the touchscreen, but it has a significant effect on speed and a slight effect on the accuracy of selecting and touching a target button embedded in a row of adjacent buttons.

3.2.2 Motor-Impaired

Guerreiro et al. [42] conducted a study to provide empirical knowledge to be used in the design of accessible touch interfaces for motor-impaired people. Participants were asked to perform target selections with a set of interaction techniques (Tapping, Crossing, Exiting and Directional Gesturing) and a set of parameterizations regarding position (corner, edge, middle) (Figure 3.4), size (7, 12, 17 mm) and direction (e.g. north) of the target. Results show that the best sizes are the medium and larger ones, except for exiting which requires bigger targets. It also showed that support from screen barriers offer physical stability, which allowed users to tap more precisely on corners and edges, and perform directional gestures less erroneously. In the middle of the screen neither of the interaction techniques was revealed as significantly more accurate.

Nicolau et al. [43] conducted a similar study but instead of having only motor-impaired participants, they included also able-bodied users, so they could identify the main resemblances and differences

Figure 3.4: Screen areas (left): black - corners; gray - edges; white - middle; Target (right).

between them. Results show that tapping can be used by motor-impaired users, although with higher error rates than those obtained by able-bodied users. Crossing targets has also shown to be a suitable alternative to motor-impaired users. Inversely, directional gesturing proved to be quite inappropriate for motor-impaired users.

3.3 Virtual Keyboards

In this section we will present works that explore different virtual keyboards that try to enhance users' typing performance, as well as works that try to understand users' touch typing behavior. Although there are several works focusing on smaller-sized devices such as smartphones, not much research has been done regarding tablets. Still, some of the categories on which we will focus are orthogonal to that fact.

3.3.1 Patterns and Touch Behavior

Just as it was verified for touch interaction in general [37], on text-entry tasks users also generally touch towards the bottom-right of targets. For instance, Nicolau et al. [43] performed a text-entry experiment with older adults, using two devices – a smartphone and a tablet – mainly to understand touch behavior. Users interacted with their dominant-hand (all participants were right-handed) on both conditions. On the smartphone condition users held the device with their non-dominant hand, while on the tablet condition the device was static on a table. Results show that users generally touch on the bottom-right of targets. The author hypothesizes this is related with hand-dominance. Furthermore, results show that tablets allow higher input rates and lower error rates. Results also show that the most common error type was omission errors. This pattern occurred across device conditions, suggesting that it was due to cognitive faults (device independent). The author also found that errors were strongly correlated with participants' tremor profile. However, each error type was correlated with different measures of tremor. *Substitutions* were largely explained by a subjective measure – task-specific tremor, while *insertion* errors, particularly *bounces* and *accidental* touches were strongly correlated with Oscillation in the X axis (dominant hand). The non-dominant hand also played an important role in mobile errors: Hand Oscillation was strongly correlated with overall *MSD* error rate, *accidental* touches and *slips*. The author also found that omitting interactions with an inter-key interval below a defined threshold, can reduce insertion errors. Furthermore, he found that the user-independent solution performed worse than the user-dependent.

Henze et al. [37] developed a typing game that records how users touch on the standard Android keyboard. When visualizing the touch distribution, the authors identified a systematic skew to the bottom and derived a function to compensate it by shifting touch events. Following the discoveries, they performed a second experiment to test three different approaches: shifting touch positions, shifting key labels and showing the touched position using dots.

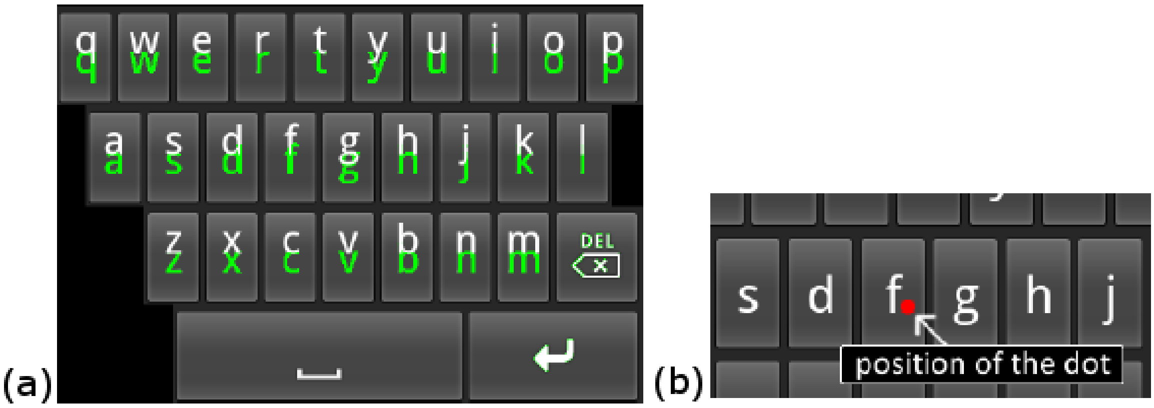


Figure 3.5: (a) The two different positions of the keys' labels. The green labels show the default Android keyboard and the white labels show the elevated labels; (b) keyboard that shows a red dot at the position where the user touches the screen after typing an "f".

On the first approach, the authors wanted to analyze how shifting the users' input would influence the touch behavior. They used three different ways to shift the users' taps. The first technique is "no shift", which does not shift the touch events. The second technique is "native shift" which is the standard Android keyboard that shifts the touch events by 10 density independent pixels towards the upper part of the screen. Finally, for the third technique ("adapted shift"), the authors derived a compensation function from the data collected in the first experiment. The technique follows the assumption that it is best to shift the users' input in a way that moves the centers of the touch distributions to the centers of the keys. The second approach is based on the assumption that users are influenced by the location of the keys' labels and, at least to some degree, try to hit the label. The labels are either shifted to the upper part of the keys ("elevated labels") or not ("default labels") (Figure 3.5(a)). Finally, the third approach informs the user about the touched position in an unobtrusive way. A small red dot appears as soon as the user's finger touches the screen (Figure 3.5(b)).

Results show that using the adapted shift results in 2.6% higher speed, 5.0% higher performance, and 7,7% lower error rate when compared to the control condition (it is also better than the native shift). For all conditions with a shift function, elevating the labels' position decreases the speed, decreases the performance, and increases the error rate for the Android keyboard. The only condition that improved by elevating the labels is without shift and with dot. Still, the overall results strongly suggest that elevating the labels' position to the upper part of the key does not improve the users' typing. The authors also found that when users get feedback from the touchscreen indicating the touched position, the error rate decreases for all conditions. However, for all conditions with the native or the adapted shift function, the dot decreases the speed up to 5.2% and also decreases the performance.

3.3.2 Adaptation/Personalization

Instead of using a function to compensate the bottom-right pattern found in previous studies, other authors decided to adapt the center of each key independently, according to the spatial distribution of keystrokes.

For instance, Himberg et al. [44] developed a method for on-line adaptation of a numerical touch pad keyboard layout. The method starts from an original layout and monitors the usage of the keyboard by recording and analyzing the keystrokes. An on-line learning algorithm subtly moves the keys according to the spatial distribution of keystrokes. In consequence, the keyboard matches better to the user's physical extensions and grasp of the device, and makes the physical trajectories more comfortable during typing. The authors present two implementations (batch and continuous adaptation) that apply different vector quantization algorithms to produce an adaptive keyboard with visual on-line feedback. On batch adaptation the coordinates of the keystrokes are recorded for a longer period, a training cycle. The key's centroid is only shifted after one batch adaptation cycle, and they are shifted to the average of keystroke locations in each key. For continuous adaptation, the key centroid is shifted instantly towards the location of the keystroke.

Results show that the keyboard size correlates slightly with the participant's thumb length. In general, with right-handed participants, the center of the keyboard is found to move towards left. However, there were a few exceptions. For some participants the center moves also clearly upwards, and for one, clearly downward. Also, the resulting keypad layouts significantly resemble each other in shape. The shape of the keyboard adapts due to natural thumb muscle trajectories so that near the upper corner stretches up and towards the palm and distant and lower corner extends. Figure 3.6 shows two different results for two different participants.

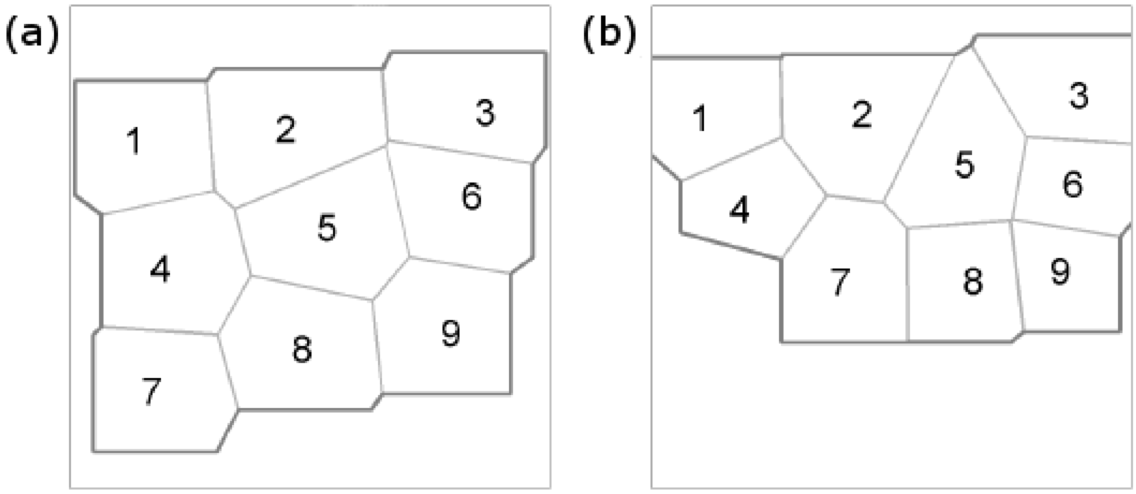


Figure 3.6: (a) Typical result; (b) untypical shape from continuous adaptation for a participant. This keypad is a result of disturbingly abrupt changes that may occur during fast adaptation.

Another work focusing on adaptation is the work of Findlater et al. [45] which introduces and evaluates two novel personalized keyboard interfaces, both of which adapt their underlying key-press classification models. The first keyboard also visually adapts the location of keys (Visual-Adaptive) while the second one always maintains a visually stable rectangular layout (NonVisual-Adaptive). The experiment consisted in 3 different sessions. All participants began typing 10 practice phrases and 20 test phrases with the Conventional keyboard. This data was used to seed the personalized keyboards. Participants then typed 5 practice phrases and 20 test phrases with each of the three keyboards. In Sessions 2 and 3, participants typed 5 practice phrases and 40 test phrases on each keyboard.

Results show that the NonVisual-Adaptive keyboard provided a typing speed improvement over Conventional, but Visual-Adaptive did not (visualizing adapted key layouts can negatively impact speed). NonVisual-Adaptive was ranked as most Efficient/Easy-to-use/Preferred, while Visual-Adaptive was considered the most Comfortable/Natural.

Azenkot et al. [46] were also interested in exploring the touch behavior of users when using soft QWERTY keyboards for smartphones. They were especially interested in studying if different touch behaviors would emerge when using two thumbs, an index finger, and one thumb. They collected text entry data from 32 participants in a lab study and described touch accuracy and precision for different keys. They found that distinct patterns exist for input among the three hand postures, suggesting that keyboards should adapt to different postures. They also discovered that participants' touch precision was relatively high given typical key dimensions, but there were pronounced and consistent touch offsets that could be leveraged by keyboard algorithms to correct errors.

Cheng et al. [47] were more interested in understanding users' preferences for the layout and position of the keyboard, depending on the grasp condition (none, one-handed, and two-handed). Therefore, the authors propose iGrasp, a novel approach that uses implicit grasps of a device to automatically adapt the on-screen keyboard's layout and position to match users' preferences (Figure 3.7).

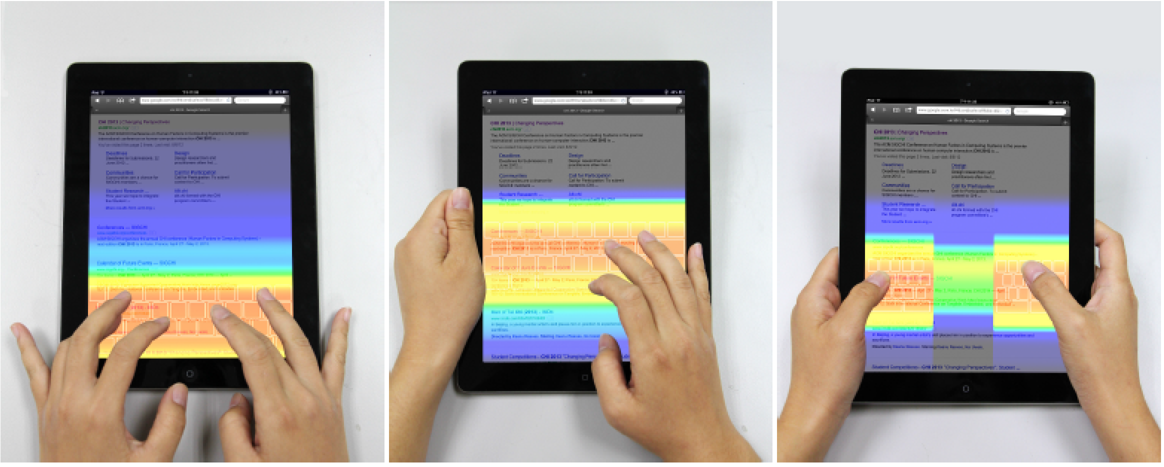


Figure 3.7: Heatmap visualization of preferred keyboard layouts and positions for three grasp conditions: (left) not grasping the devices, (center) grasping the devices with one hand, (right) grasping the devices with both hands. The most preferred keyboard modes are merged+docked, merged+undocked and split+undocked.

iGrasp supports two adaptation modes. *iGraspSwitch* senses the user's grasp condition (none, one-handed, and two-handed) by grouping 46 sensors into four sensor groups and adapts the keyboard to the same layout and position that was last used for the currently sensed grasp condition. *iGrasp-Position* additionally senses the current grasp position and continuously adapts the keyboard to the grasp location by utilizing all 46 sensors. To avoid constant movement of the keyboard while typing, the system stops re-positioning the keyboard once the user has started typing.

Their evaluation shows that participants are able to begin typing 42% earlier using the iGrasp's adaptive keyboard compared to manually adjustable keyboards. In addition, participants also rated iGrasp significantly easier to use (4.2 vs 2.9 on 5-point Likert scale). They also found that continuous position adaptation shows no statistically significant improvement over users' last-used positions.

Yin et al. [48] proposed a new approach for improving text entry accuracy on touchscreen keyboards by adapting the underlying spatial model to factors such as input hand postures, individuals, and target key positions. To combine these factors together, they introduce a hierarchical spatial backoff model (SBM) that consists of submodels with different levels of complexity. Considering that in practice people may switch hand postures (e.g., from two-thumb to one-finger) to better suit a situation, and that the submodels may take time to train for each user, a specific submodel should be applied only if its corresponding input posture can be identified with confidence, and if the submodel has enough training data from the user. The authors introduced the backoff mechanism to fall back to a simpler model if either of these conditions is not met. They implemented a prototype system capable of reducing the language-model-independent error rate by 13.2% using an online posture classifier with 86.4% accuracy.

Stone [49] was more concerned with the fact that devices like iPhone and iPod Touch are not so accessible for the elders, since buttons are too small. The author argues that this kind of devices should have a gesture that allows to choose multiple sizes for interface elements (fonts, buttons and icons). Furthermore, this should also be extended to virtual keyboards, giving the opportunity to the user to choose between different layouts (traditional QWERTY keyboard, 12 button mobile phone interface and binary interface), depending on the current situation and his capabilities. However, no implementation nor experimental evaluation was performed.

3.3.3 Language Models

Another way to significantly reduce the error rate of soft keyboard usage is through language models combined with models of pen placement, as emphasized by Goodman et al [50]. When a soft keyboard user hits a key near the boundary of a key position, both language model and key press model can be used to select the most probable key sequence, rather than the sequence dictated by strict key boundaries. Results show that this can lead to an overall error rate reduction by a factor of 1.67 to 1.87.

MacKenzie et al. [51] also made use of language models, but in their case for an eye typing system that uses not only word prediction, but also a letter prediction and a fixation algorithm. Similar to word prediction, letter prediction chooses three highly probable next letters and highlights them on an on-screen keyboard by changing the color of the button (Figure 3.8). The fixation algorithm chooses which button to select for eye-over highlighting. It often chooses the desired button even if another button is closer to the fixation location. Error rates were reduced when using the fixation algorithm combined with letter prediction.

Another approach to highlight keys involves making the rendered keys larger or smaller, depending on their likelihood [52]. The proposed solution helps to facilitate the selection task by expanding the next entry. Moreover, the prediction system reduces visual scanning time to find the letter the user is looking for. Results show that users were 25.14% faster and more accurate with BigKey virtual

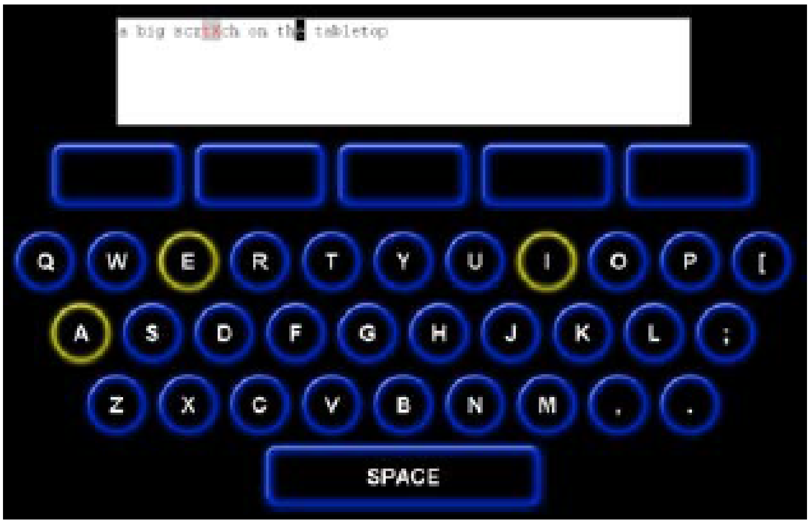


Figure 3.8: On-screen keyboard with letter prediction, highlighting ”a”, ”e” and ”i” keys.

keyboard than with normal virtual keyboard.

A different approach to highlight the most likely keys, involve labeling the corresponding keys in bold [53] (Figure 3.9). Since we do not have complete access to this paper, we are not aware of the results obtained. Their goal was to optimize the performance of novice users in different kinds of layouts (AZERTY, Metropolis-like), through the use of visual clues.

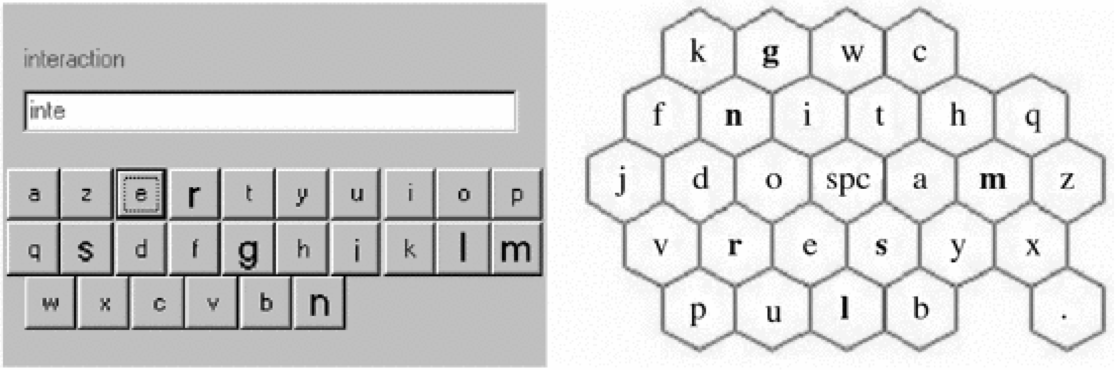


Figure 3.9: Use of visual clues with an AZERTY (left) and a Metropolis-like keyboard (REF) (right)

Lucas [54] developed two different keyboards for smartphone, that used a mock-up *prediction algorithm* to highlight the four ”most probable” keys; one of the variants changed the alpha value while the other changed the width. Each of these variants was tested with the mock-up algorithm suggesting the right key 100% and 20% of the times. Results show that the *alpha* and *width* variants with the mock-up algorithm that suggests the right key 100% of the times, reduced error rate significantly when compared to the QWERTY baseline condition. Results also show that the *alpha* variant with the mock-up algorithm that suggests the right key only 20% of the times, was also able to reduce error rate significantly when compared to the QWERTY baseline condition. This means that, even with a bad *prediction algorithm*, there are advantages in highlighting the keys by changing its alpha value. Regarding input rate, all variants performed similarly, except the *width* variant with a bad prediction algorithm which performed worse.

Still, some studies [44] report that users can find the dynamic rendering of keys distracting. In order to avoid the aforementioned distraction, Gunawardana et al. [55] developed a method that expands or contracts the keys' underlying area, based on a language model. Furthermore, their method differs from state-of-the-art methods because they preserve the area around the label, which gives the possibility to the user to select a key that is not considered by the language model as one of the most probable (Figure 3.10(b)).

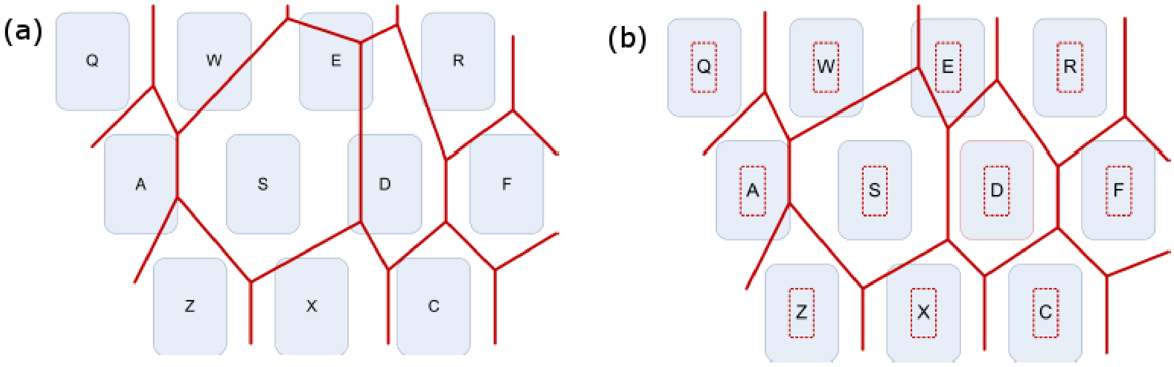


Figure 3.10: (a) A schematic example where key-target resizing has made it difficult for the user to type the key "e" because the language model predicts that it is very unlikely compared to the key "s". The key-target outlines are shown in heavy lines; (b) an schematic example where target areas respect each key's anchor. The target area outlines are shown in heavy lines, while the anchor outline are shown in broken lines.

The authors performed a user study with an unaltered virtual keyboard to gather data about touch positions, intended key and pressed key. Afterwards, a simulation study was performed in order to find the optimum size of the anchor that would minimize errors (including the inexistence of anchor which is represented in Figure 3.10(a)). Results show that maintaining an anchor yields better results than the normal keyboard and the keyboard with key-target resizing without anchors.

No studies regarding alternative keyboard layouts were presented here, because one of our goals is to develop virtual keyboards that aid new users to input text, without hindering older users who are already experienced with QWERTY keyboards. As stated in Chapter 2, learning new procedural tasks and developing new automatic processes is difficult for older adults. Therefore, making use of older users' crystallized knowledge (QWERTY layout) will yield better results.

3.4 Discussion

In this section we will discuss each subset of works, to better understand which are the relevant characteristics that gestures/interfaces/virtual keyboards should include, to enable us to develop more inclusive systems for older adults.

3.4.1 Eliciting Gestures

Most of the gestures used nowadays on most devices are gestures defined by system designers, who personally employ them or teach them to user-testers [34]. This means that these gestures do not necessarily take into account the preferences of the generality of the users, and even less the specific needs of older adults. We will try to understand the users' preferences through several dimensions such as number of fingers and hands employed, type of gestures and gesture direction.

Number of Fingers. Stöbel [36] concluded that older adults rely less on multi-finger gestures than younger adults. In other study (Study 1B [36]) the author observed that one-finger gestures can be performed reasonably well, if the device is held in one hand and the other hand performs the gesture. Regarding the use of two-finger gestures using the same posture, these are more erroneous, slower and less accurate when comparing with a stabilized posture (e.g.: using the device on a table).

Older users sometimes use one, two or three fingers to select objects, depending on its size [12]. Wobbrock [34] argues that gestures should not be distinguished by the number of fingers employed, because people generally do not regard the number of fingers they use in the real world.

Number of Hands. In the experiment conducted by Wobbrock et al. [34] participants preferred mainly 1-hand gestures. In other study [35] some participants stated that gestures using multiple hands would become tiring, and time consuming, if they were to use them with any frequency.

Lepicard and Vigoroux [56] tested the use of the two-hands in other context. Participants had to select targets with one or two hands, depending on the test. Results show that the use of two-hands increases the selection time and error rate for older people. The study shows it is important to reduce the cognitive overload in the interface, in terms of number of targets and number of blocks, so older people can achieve a higher performance.

Type of Gestures. Stöbel [36, 40] showed that there's a strong preference for manipulative and indexical gestures across all age-groups. Still, older users are more willing to trade efficiency for familiarity, and as such are more likely to accept symbolic and iconic gestures than the younger group. Results also show that alphanumerical gestures were perceived as little suitable by older users. There are also considerable differences between age groups regarding the question which gestures are most suitable for a certain task. Moreover, it still proves difficult to establish a gesture set which is particularly suited to older users.

Results from the studies conducted by Wobbrock et al. [34] show that simpler commands resulted in physical gestures (gestures that have the same effect on a table with physical objects), while more complex commands resulted in metaphorical (when a gesture acts on, with, or like something else) or symbolic gestures (visual depictions).

A follow up study conducted by Wobbrock et al. [35] showed that, in general, participants preferred simpler gestures - gestures that were physically easier to perform and/or demanded less cognitive effort - to more complex ones. Also, gestures with conceptually simpler natures (those based on analogy to the physical world, and those using common symbols) were preferred by the participants to those with more conceptually complex natures (those based on metaphorical or abstract mappings).

Gesture Direction. Stöbel [40] conducted an experiment on which he concluded that older adults seem to benefit especially from familiar patterns. They are able to perform familiar patterns faster and more accurate than unfamiliar patterns.

In another study, Stöbel [36] concluded that upward and leftward gestures are subjectively perceived as more difficult than downward and rightward gestures. Also, several older users mentioned an increase in difficulty with the number of corners and direction changes.

Stöbe [36]l also noticed a tendency for older participants to perform a swipe in the opposite direction compared with the younger group when trying to scroll a list up or down. Stöbel concluded that the preferred scrolling direction is habit-dependent and strongly depended on prior experience with similar or different technical systems, rather than a general preference of age group.

In a different context - thumb interaction and no older adults -, Yatani et al. [57] hypothesized that error rates would vary with direction. Although no significant effect of direction was found, some participants did dislike some directions (NW, W, and SW) because they involved stretching the thumb, whereas other participants disliked other directions (S and SE) because they involved contracting the thumb.

3.4.2 Targets

To allow users to achieve the highest performance when interacting with touchscreen devices, different characteristics of targets, such as size, position, and spacing must be taken into account. These characteristics vary from technique to technique, as well as from the characteristics of the user himself. Although, we are more interested in elders, not all works focus on this group of users. So we also take into account what happens with the younger group, as well as motor-impaired users which are somehow comparable with older adults, because of their limited motor abilities.

3.4.2.1 Target Size

Tapping. Kobayashi et al. [7] argues that interactive objects such as buttons, icons, and clickable text should at a minimum be larger than 8 mm, since in the experiment they conducted with elders, the touch locations were mostly distributed within 8 mm on the physical screen, regardless of the device and the target size.

Jin et al. [41] also conducted an experiment with older users and concluded that the target size depends on the reaction time needed, as well as on the fact that there's adjacent buttons or not. The target size should be 11.43 mm if there's no adjacent buttons and a reaction time of around 1400 ms is acceptable. If better performance is required the target size should be 19.05 mm. If screen space is limited and design uses rows of adjacent buttons the target size should be 16.51 mm.

Guerreiro et al. [42] conducted an experiment with motor-impaired users to assess the best target size for the tapping technique. Three sizes were tested: 7, 12 and 17 mm. Results show that the best sizes are the medium and larger ones (there's no significant effect between them on task error).

Nicolau et al. [43] also conducted an experiment with motor-impaired users, but in this case they also included able-bodied users in order to understand the resemblances and differences between them. Their performance on Tapping is similar. Both perform worse with smaller targets (7 mm), and error rates start to converge at 12 mm.

Crossing. The target size for the crossing technique is similar to the tapping one as concluded by Guerreiro et al [42]. The best sizes are the medium and larger ones (12 and 17 mm).

Nicolau et al. [43] concluded that crossing targets is a suitable alternative for motor-impaired users, although it wasn't clearly specified what's the best target size.

Exiting. Unlike Tapping and Crossing, Exiting requires bigger targets (17 mm) as concluded by Guerreiro et al [42].

Directional Gesturing. Directional gesturing has an unconstrained nature and as such does not require a target selection. Results of the experiment conducted by Nicolau et al. [43] showed that it proved to be an accurate interaction technique for able-bodied users, as opposed to motor-impaired users to whom it was quite inappropriate.

Swabbing. Swabbing had better results with the biggest target size (41 mm). When an interface requires targets smaller than 41 mm, swabbing is a better choice than tapping.

3.4.2.2 Target Position

Tapping. Guerreiro et al. [42] conducted an experiment with motor-impaired users and concluded that corners and edges offer higher stability towards a precise movement, although this is not reflected in higher accuracy. This means that tapping is suitable for any of the target positions as was also concluded by Nicolau et al [43]. For able-bodied users, Nicolau et al. [43] concluded that for small sizes, targets are easier to acquire on the middle of the screen. Also, regarding vertical distance, those small targets are harder to acquire near the bottom edge.

Crossing and Exiting. Crossing and exiting are essentially the same technique. The only difference is that the target of the former is positioned in the middle of the screen, whereas the target of the later is positioned in the corner or edge of the screen. The results of the experiment conducted by Guerreiro et al. [42] showed that acquiring mid-screen targets (Crossing) is easier than towards screen barriers (Exiting). The results also showed that, although users achieve similar accuracy on corners and edges, they produce more erroneous gestures when their movement is more restricted (in the corner, the direction of the movement is restricted to 90 degrees, while for an edge it is restricted to 180 degrees).

Nicolau et al. [43] only performed experiments with crossing (middle of the screen). Results show that in this area the accuracy for motor-impaired and able-bodied users using crossing is similar to Tapping.

Swabbing. As a restriction of this technique, targets are positioned in a radial layout. Authors [38, 39] did not report increase in performance depending on the position.

3.4.2.3 Target Spacing

Jin et al. [41] conducted an experiment with older adults to understand how spacing between buttons influences their performance. They preferred and were also more accurate with a spacing of 6.35 mm in rows of adjacent buttons. Large spacing only increases the time for searching the screen and moving to touch the target button. No space between buttons is associated with the lowest accuracy and the lowest preference ratings.

3.4.3 Virtual Keyboards

Regarding virtual keyboards we will present a discussion about: shifted touch, adaptation and language models.

3.4.3.1 Patterns and Touch Behavior

Overall, from the data gathered from current literature, it seems that users' touch points are mainly skewed to the bottom [37]. Other studies have also reported a bottom-right pattern [14]. If this pattern is also found for older adults, we will be able to reduce error rate by simply shifting all touch points to the top (or to the top-left). Nicolau [14] also reported that it is possible to reduce (accidental) insertion errors, by omitting interactions with an inter-key interval below a defined threshold. He reported that this characteristic is user-dependent. This gives us some guidance on how to reduce insertion errors.

3.4.3.2 Adaptation/Personalization

If a different pattern is found, or if no pattern is found at all, another choice is to adapt the center of each key independently, according to the spatial distribution of keystrokes. Although we believe this solution is promising, no study addresses the effects of adaptation during a long period of time. Will the keyboard converge to a specific layout, or will it continuously shift towards an unrecognizable keyboard? Regarding the context of the PAELife project, will this kind of solution be adequate for a shared tablet (husband and wife)?

One of the works also focused in understanding users' preferences for the layout and position of the keyboard, depending on the grasp condition (none, one-handed, and two-handed). Younger users grasp the tablet device differently in different situations; still, we do not know if this is true for older adults. We believe that older users will mainly prefer to interact with the tablet on a table or on their lap, in order to have more stability.

Stone [49] proposed a solution that attempts to make the text-entry task on touch devices more accessible for older adults. Still, it is a solution more adequate for smartphones, since tablets do not have the same size restriction as smartphones.

3.4.3.3 Language Models

Current literature shows that virtual keyboards based on language models can improve typing performance either by highlighting the next most probable keys or by increasing the underlying area of the keys, invisibly. The former can help users who are not acquainted with the QWERTY keyboard to focus their attention on the most probable keys, allowing a greater input rate, while the latter is less intrusive and is able to reduce error rate. The study performed by Lucas [54] also indicates that even a bad *prediction algorithm* is able to enhance users' typing performance (in the case of changing the alpha value of the key).

3.5 Design Recommendations

In this section we will present a list of features, infered by the previous discussion, that inclusive systems for older adults should comprise (also applicable for virtual keyboards in particular).

- The gesture set should all be focused on one-finger gestures;
- Gestures should not be distinguished by the number of fingers. E.g.: There shouldn't be a difference between Tapping with one or two fingers;
- If multi-finger gestures are necessary, these should have visual cues indicating that more than one-finger is required;
- The interface should not be overloaded with too many blocks and targets;
- Manipulative and indexical gestures should be included in the gesture set;
- Gestures with conceptually simpler natures should be included in the gesture set;
- Familiar patterns, which includes avoiding gestures with upward and leftward movements and using recognizable shapes, should be included in the gesture set;
- Target Size:
  - **Tapping:** 11.43 mm when there's no adjacent buttons and a reaction time of around 1400 ms is acceptable; 19.05 mm if better performance is needed; 16.51 mm if space is limited and design uses rows of adjacent buttons.
  - **Crossing:** Between 12 and 17 mm.
  - **Exiting:** 17 mm, although this technique is not adequate for older adults.
  - **Swabbing:** 41 mm yields best results.
- Target Position:
  - **Tapping:** Targets should be placed in the middle of the screen. Edges and corners adequacy should be assessed specifically for the older adults.
  - **Crossing and Exiting:** A good alternative for Tapping is Crossing targets (middle of the screen).
- Target Spacing: 6.35 mm spacing (in rows of adjacent buttons).
- Substitute Tapping by Swabbing if there is the need to include older adults who suffer from hand tremor;
- Touch points should be shifted to the top (or top-left) to compensate their skewness;
- Interactions with an inter-key interval below a certain threshold should be omitted;
- If different patterns are found, a model that adapts the center of each key independently, could be useful;
- Use language models to highlight the most probable keys or to (invisibly) increase the underlying area of the key. On the second case, the area around the label should be preserved (still to be assessed for older adults);
- Use bigger devices, like tablets, if possible.

4

Keyboard Alternatives

From the information gathered about the current literature, we are now able to propose several keyboards that aim to aid new users to input text, without hindering older users who are already experienced with the QWERTY keyboard. In this chapter we will present the proposed architecture for the keyboards to be used during the studies, and explain each of its modules. We will also present an evaluation of the prediction system and a flowchart for each of the developed variants.

4.1 Architecture

The proposed architecture is composed by two main modules (Figure 4.1): the keyboard module and the prediction system module. The keyboard module is further decomposed in two different sub-modules: the touch analyzer module and the visual representation module.

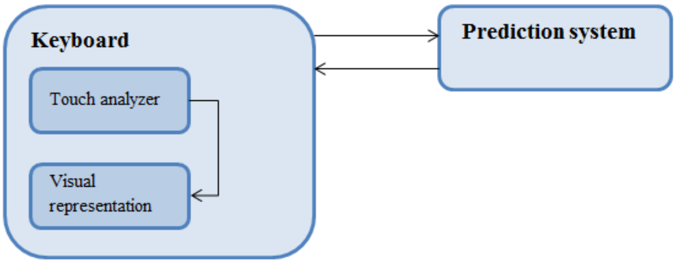


Figure 4.1: Proposed architecture.

The touch analyzer module is responsible for mapping the input touch coordinates into a specific key, while the visual representation module is responsible for changing keys' visual attributes when certain conditions are met. The prediction system module is responsible for finding the most probable words for the given input. Sections 4.2 and 4.3 further explore the important features of each of the main modules.

As a restriction of the PAELife project, the keyboard had to be developed as a Windows Store App for Windows 8. Since Windows' virtual keyboard is not extensible, we had to create our own traditional QWERTY keyboard from scratch, and develop the remaining variants based on this one.

In order to have complete control over the actions performed by the users, we had a visual representation of the keys, but users would actually interact with a canvas, placed over the keyboard. However, the canvas is completely transparent, so for the users it feels as if they are interacting directly with the keys of the keyboard (Figure 4.2 (a)). In Figure 4.2 (b) we show the area occupied by the canvas, in blue. Depending on the coordinate pressed (and released) by the user, the interaction is assigned to one of the 27 keys, unless the user taps on an empty space (area without key); in such case no key is assigned. Each key had 20 mm of width and 15 mm of height, which follows the design implication proposed by Nicolau [14] ("width rather than height"). Visually, there is a space of 2 mm between keys, horizontally and vertically. However, our implementation does not allow pressing between keys: each touch is always assigned to a key. This makes the keyboard more responsive, thus avoiding the frustration of performing a tap that does not produce a character.

The touch analyzer module is responsible for the mapping between coordinates and keys. As soon as the touch analyzer maps the coordinates to a key, it communicates this information to the visual representation module, so the appropriate visual feedback is given to the user; that is, if the key is currently pressed, the button is colored white, while the label is colored black. When a release occurs, it goes back to the normal visual state (black button, white label).

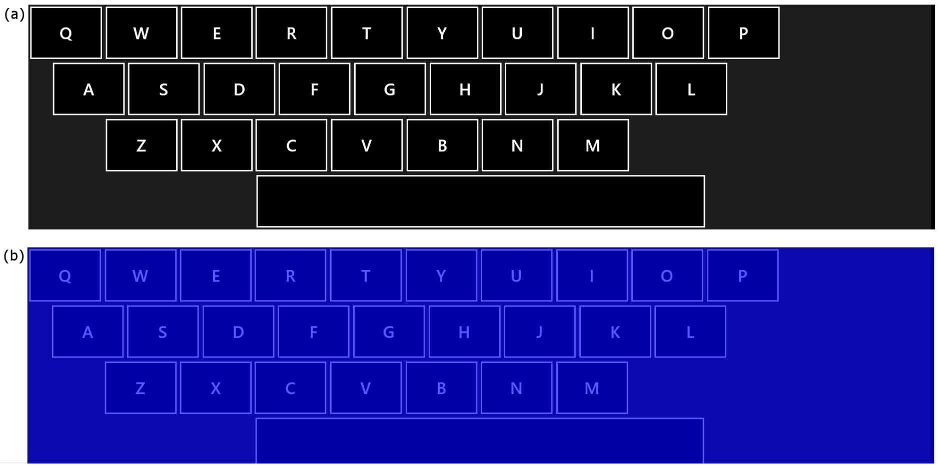


Figure 4.2: (a) The QWERTY keyboard as it is presented to the users (invisible canvas); (b) The QWERTY keyboard with the canvas opacity set to 50%.

4.1.1 Technology

The QWERTY keyboard and remaining variants were developed for Windows 8 (Windows Store App) using C# programming language. The UI elements (keys) were developed using the Extensible Application Markup Language (XAML).

4.2 Text Prediction

In order to develop more advanced variants of the virtual QWERTY keyboard, we used two types of prediction to anticipate what the user is going to type: *word prediction* and *next letter prediction*. If the *prediction system* is able to predict correctly, the number of keystrokes needed to write a sentence decreases. Thus, it can enhance the typing speed and reduce the physical effort required to compose messages. There are several techniques to predict the text the user is trying to input; some more complex than others. The most advanced prediction systems have learning features, are able to make inferences, are adaptable and are able to act independently [58]. However, since the aim of this work was not developing a novel and more efficient prediction algorithm, we opted for a simplistic one. Our *prediction system* only takes word frequencies into account. When the user types the beginning of a word, the system offers the most probable words beginning with the same character(s). This approach has achieved good results in previous studies [58].

To implement the word prediction system, we used the CETEMPúblico Portuguese text corpus[1], which contains approximately 180 million words. From that corpus we processed the word frequencies and then stored them in a dictionary structure that contains all the information about each word and its prefixes' frequencies, so that the information can be efficiently accessed. When the user is typing, the *predictor* shows an ordered list of the most frequent words that start with the typed prefix. After implementing the *word prediction system*, we decided that the *next letter prediction* should be based on the same algorithm in order to avoid the case of the *letter prediction algorithm* suggesting a letter that is not present in any of the suggested words. For instance, imagine the user wants to type ”home”, and at this point has already typed ”ho”. If the *letter prediction* algorithm suggests the letter ”t” (hot) and the *word prediction system* suggests the word ”home” it could be confusing for users. So we decided to implement the *letter prediction* algorithm through the *word prediction system*. What happens is, since the most probable word is ”home”, and the user has already typed ”ho”, the *letter prediction* algorithm will choose to highlight the ”m” key.

4.2.1 Results of the text prediction algorithm

To evaluate the results of the implemented *prediction system*, 88 sentences were extracted from a written language corpus [14] (sentences available in Appendix B.1). Each sentence had 5 words with an average size of 4.48 characters and a minimum correlation with the Portuguese language of 0.97. As we will see later, in Chapter 5 and 6, these are the same sentences used in the evaluations of the developed virtual keyboards. Figure 4.3 shows the result of the word prediction. Only words of length between 6 and 12 characters were considered, because any length lower than that does not represent a considerable save in key presses. For instance, if a user already typed two letters from a four letter word, the difference between tapping on the suggested word and typing the remaining letters will not be much, regarding keystrokes. Words with a length over 12 characters were not considered because they only represent 6% of the total words from the written language corpus. Therefore, they are not common. As expected, the more the suggested words, the greater chance of success. However, the success rate does not seem to increase much when presenting a list of more than 6 words (only an increase of 3% between suggesting 6 and 7 words). We must also take into account that the more words we suggest, the more cognitive effort is required for the users to process the suggestions' list. Therefore, there should be a balance between the number of words suggested (which affect directly the success rate) and cognitive effort required to process the suggestions list (which increases with the number of words).

We also performed the same evaluation for the *next letter prediction*. As we can see in Figure 4.4, it is much easier to correctly predict the next letter (space included) than to predict the full word the user is typing. Up to 4 letters, the success rate increases from 4-7% and after that, only an increase of 0-2% is found. Note that we never hit 100% success even if we highlight all the letters of the keyboard because one of the sentences had a surname that was not in our dictionary, so the *prediction system*

---

[1] http://www.linguateca.pt/cetempublico

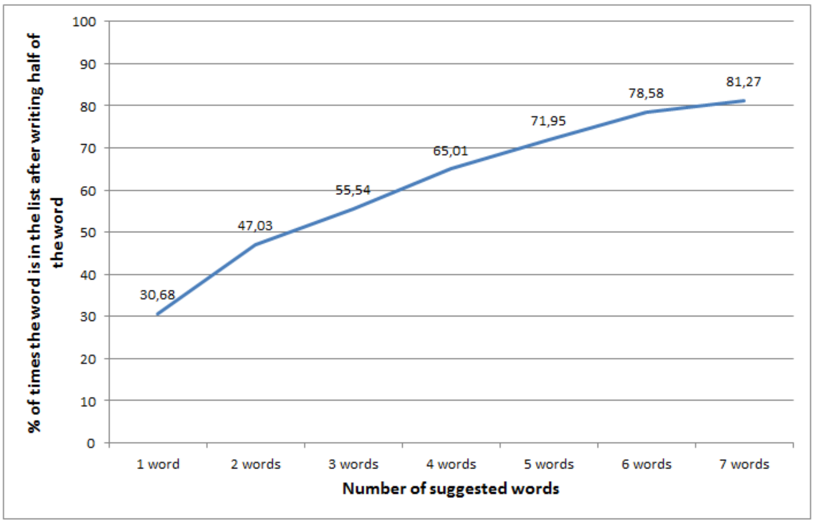


Figure 4.3: Performance of the word prediction algorithm.

is not able to predict it.

Since most text prediction methods are heterogeneous, and since the measurements offered by authors are based on heterogeneous parameters (not always clearly described) [58], its hard to assess how well our algorithm performs when compared with others.

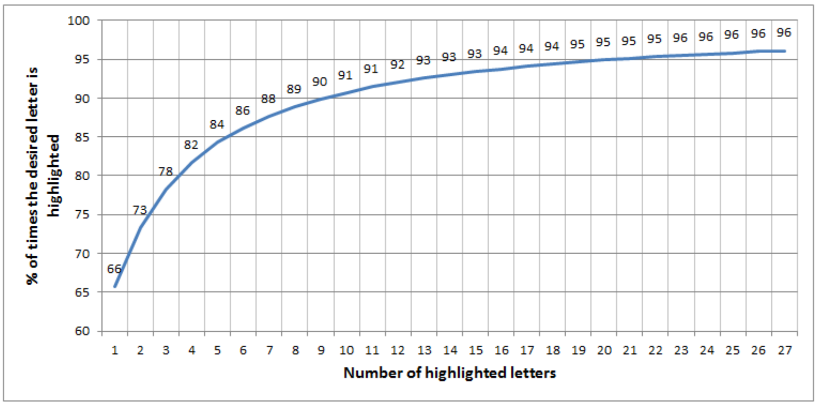


Figure 4.4: Results of the letter prediction algorithm.

4.3 Keyboards

In this section we will present all the developed keyboards. We explain the existence of each variant and the features that make them different from each other. The traditional QWERTY keyboard and, the *Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible* variants were used during the user studies, while the *Single Touch*, *Intra-key*, *Inter-key* and *Combined Timed* variants were only used as simulations. The simulations were used because it would be unrealistic to ask each older participant to perform tests with ten different variants. That would require at least three different sessions with each participant. It is important to note that simulations were only performed with variants that were visually similar to the QWERTY keyboard; the differences were only regarding the processing of the touch inputs (this will be further discussed in Chapter 6, Section 6.3).

4.3.1 Traditional QWERTY keyboard

The *traditional QWERTY* keyboard is the baseline keyboard we developed. It is similar to the other virtual keyboards existing on most touch devices, minus the fact that letters are entered using a lift-off strategy. This strategy avoids multiple insertions, since older users' key presses are usually long [14]. Also, a letter is only inserted if the released key is equal to the pressed key. In Figure A.1 we present a flowchart that shows the functioning of the *traditional QWERTY* keyboard.

4.3.2 Color variant

The *Color* variant uses the *prediction system* described in section 4.2 to highlight the next most likely letters of the current word. Regarding the number of keys to highlight, we decided to highlight four keys because Faraj et al. [52] have previously tested highlighting one, two and four keys, obtaining better results with the latter. Also, the results of the letter prediction evaluation showed that highlighting four letters has an increased success rate when compared to highlighting fewer letters. Therefore, this is the optimum number of letters to highlight. We decided to highlight the keys by changing its color from black to gray, which is a neutral color (Figure 4.5 (a)), to ensure that cultural connotations associated with particular colors are avoided (e.g., green and red colors have positive and negative connotations, respectively). We also increase the size of the key's label. The highlight is continuous: the more probable the letter, the brighter the color and bigger the label on the key. The biggest goal of this variant is to help users who are not completely familiarized with the QWERTY layout, to locate faster the key they want to type. We also expect users to commit fewer errors by noticing if they are about to press a key that is not highlighted, or by acknowledging they missed a key press. Several studies used similar approaches in other contexts [51, 54]. ***To our knowledge, it has never been tested with older adults.*** In Figure A.2 we present a flowchart that shows the functioning of the Color variant, which uses the letter prediction algorithm.

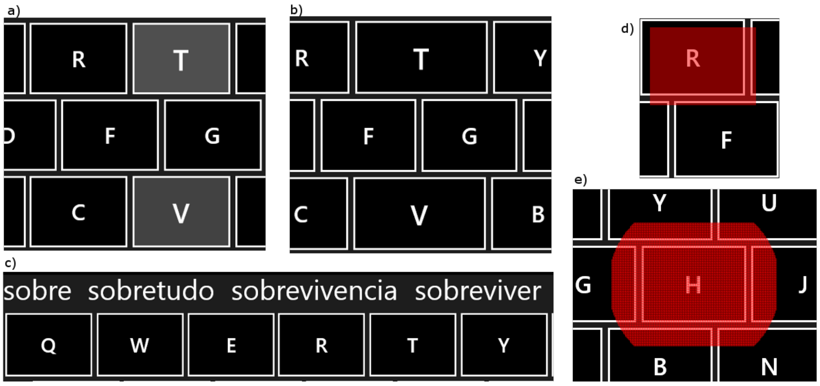


Figure 4.5: (a) Color variant; (b) Width variant; (c) Predicted Words variant; (d) Shifted variant; (e) Size Invisible variant.

4.3.3 Width variant

The *Width* variant uses the same principle as the *Color* variant. The difference is that, instead of highlighting the keys by changing its color, it highlights the keys by increasing their width by 30% (Figure 4.5 (b)). However, for this variant we did not use a continuous increase in size based on the probability of the letter, because it was much harder to tell which buttons were highlighted if the size increased was little. As happens with the *Color* variant, the label of the key increases in size proportionally to its probability. With this variant we expect users to commit less substitution errors by hitting the desired key instead of the neighbor keys, since the most likely keys are bigger. A previous study [52] has shown that this approach can improve both the input and error rates of the typed sentences on smartphones. ***To our knowledge, this has never been tested with older adults on tablet devices.*** The *Width* variant has the same functioning as the *Color* variant (Figure A.2), differing only on the applied visual change; instead of changing the color of the key, it changes the width of the key. It also uses the letter prediction algorithm.

4.3.4 Predict Words variant

The *Predict Words* variant is a common alternative that can be selected as typing method in most touch devices. While the user is typing, a list of the most likely words is shown in a horizontal ribbon above the keyboard (Figure 4.5 (e)). If the word the user wants to type is on the suggested list, he can save some key touches by tapping it so the full word along with a space character will be inserted. In the literature, there is no conclusive study about the optimum number of words to suggest [58]. Since there is a trade-off between the number of suggested words (that directly affects the success rate) and the cognitive effort required for the user to process the list, we opted to suggest 4 words. Although this is not a novel approach, we wanted to systematically confirm if this variant would possess any advantage over the normal QWERTY keyboard, either in typing speed or quality of the transcribed sentences (fewer errors). It is a fact that users save some time by tapping fewer keys, but they also waste time in the cognitive effort of continuously checking the suggestion list. ***To our knowledge, this has never been tested with older adults on tablet devices.*** The *Predict Words* variant has a similar functioning as the *Color* and *Width* variants (Figure A.2), differing only on the applied visual change; instead of changing the color or width of the key, it updates the horizontal ribbon above the keyboard with the new suggested words. This one uses the word prediction algorithm.

4.3.5 Shifted variant

The approach of shifting the real touch area of keys from its visual representation is also common in many virtual keyboards [37, 59]. In small touch devices, like smartphones, this approach has proven its benefits [37, 59]. However, no systematic studies have been performed for tablet devices. These devices vary from the former not only in screen size, but also in the typing posture users assume when using them; when using smartphones users usually type with the two thumbs, while with tablets they can type with all fingers. Previous studies have consistently shown that users' touch points are skewed to the bottom-right, on smartphone devices [37, 59]. Still, neither of those studies indicates the optimum shift we need to apply to compensate the users' tendency to touch in the bottom right of targets. Taking this into account, we choose to deviate the real touch area of the key 10% of the key's height to the bottom, and 10% of the key's width to the right in our implementation (Figure 4.5 (c)). Note that 10% was a value chosen by us, because it seemed to work well. The user studies will help us verify if this is the best value indeed. Visually for the user, this variant is exactly the same as the *QWERTY* keyboard. We expect users to commit less *neighbor substitution* errors with this variant. ***To our knowledge, this has never been tested with older adults.*** In Figure A.3 we present a flowchart that presents the functioning of the *Shifted* variant.

4.3.6 Size Invisible variant

Similar to the *Width* variant already described, this variant increases the size of the most likely keys. However, this variant does it only internally; to the users it remains visually the same as a regular *QWERTY* keyboard. This approach has also been the aim of previous studies [55]. In our implementation, we increased the likely keys' width in 50% (25% to the left and 25% to the right) and 50% in height (25% to the top and 25% to the bottom). Note that these values were chosen by us, because they seemed to work well. The user studies will help us verify if these are the best values indeed. We also imposed a maximum distance from the center of the key (125% of half of its original diagonal) so that the final shape of the touch area of the highlighted keys had rounded corners (Figure 4.5 (d)). The touch point is always assigned to the key that has the lowest Euclidean distance from its center to the touch point. With this variant we expect users to commit less *neighbor substitution* errors by hitting the desired key instead of the neighbor keys, since the most likely keys are internally bigger. ***To our knowledge, this has never been tested with older adults.*** In Figure A.4 we present a flowchart that presents the functioning of the *Size Invisible* variant.

4.3.7 Single Touch variant (simulated)

The *Single Touch* variant behaves exactly as the baseline keyboard, except it is single touch; that is, instead of allowing more than one touch point at a time, it only allows one. So, if a user presses a second key before releasing the first one, the second touch interaction will be discarded, thus only inserting the first character. Authors that performed studies focusing on touch devices have reported that in general users prefer to interact mainly with one hand and only one finger [35, 56]. This variant will help us understand if single touch is indeed a more accessible choice for older adults, or if multi-touch is more adequate. In Figure A.5 we present a flowchart that presents the functioning of the *Single Touch* variant.

4.3.8 Intra-key Timed variant (simulated)

This variant emerged mainly to correct *accidental insertion* errors. This kind of error is characterized by a reduced time interval between the press and release of a key (intra-key). In order to correct this kind of error, this variant assumes the existence of a threshold that indicates which interactions are considered *accidental insertions* and which are not. Since we do not know what the best threshold for all users is, or if different users will require different thresholds, we will perform several simulations in order to find the best threshold that maximizes the correction of *accidental insertion* errors and minimizes the creation of new errors. A similar approach has been executed by Nicolau [14]. This variant is built upon *Single Touch* variant; i.e., instead of allowing more than one touch point at a time, it only allows one. Regarding all other aspects, this variant behaves just like the *baseline QWERTY* keyboard. In Figure A.6 we present a flowchart that presents the functioning of the *Intra-key Timed* variant. The condition ”elapsed time less than threshold” (Figure A.6) is between the press and release of the same interaction (intra-key).

4.3.9 Inter-key Timed variant (simulated)

This variant emerged mainly to correct *double insertion* errors. This kind of error is characterized by the insertion of a second character with a reduced time interval between the release of the first key, and the press of the second key. Just like the previous variant, in order to correct this kind of error, this variant assumes the existence of a threshold that indicates which interactions are considered *double insertions* and which are not. Since we do not know what the best threshold for all users is, or if different users will require different thresholds, we will perform several simulations in order to find the best threshold that maximizes the correction of *double insertion* errors and minimizes the creation of new ones. A similar approach has been executed by Nicolau [14]. Just like the previous variant, this variant is built upon *Single Touch* variant; i.e., instead of allowing more than one touch point at a time, it only allows one. Regarding all other aspects, this variant behaves just like the *baseline QWERTY* keyboard. In Figure A.7 we present a flowchart that presents the functioning of the *Inter-key Timed* variant. The condition ”elapsed time less than threshold” (Figure A.7) is between the release of one key and the press of a second key (inter-key).

4.3.10 Combined Timed variant (simulated)

The *Combined Timed* variant is the combination of *Intra-key* and *Inter-key Timed* variants. Therefore, its main goal is to correct *accidental* and *double insertion* errors. Both variants keep track of time in order to operate. However, the variants are independent of each other, which mean that each one will have its threshold and operate independently. In Figure A.8 we present a flow chart that presents the functioning of the *Combined Timed* variant. The upper condition ”elapsed time less than threshold” (Figure A.8) is between the release of one key and the press of a second key (inter-key), while the one below is between the press and release of the same interaction (intra-key).

5

Baseline Study

In this chapter we will focus on a user study we conducted with 20 regular users and six of the virtual keyboards presented in Chapter 4 - *traditional QWERTY* keyboard, and *Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible* variants. This first study was not focused on older users because in this phase we were interested in assessing which were the most promising variants. Therefore, as a baseline study, we opted to perform the test with regular users, since they were easier to find. This Chapter aims to provide the knowledge needed to design text entry solutions that help improve users' performance. We describe users' typing behaviors and performance errors, as well as their comments.

5.1 User Study

Touchscreen devices are increasingly replacing their button-based counterparts. However, touchscreen devices lack the haptic feedback of physical buttons, making it harder to accurately select targets. This characteristic hinders certain tasks, such as text-entry, on which the user has to constantly select one of many small targets. Our goal is to better understand users touch typing behavior and assess if any of the five variants enhances users' performance. We also want to assess if our application is robust enough to be used in the older adults user test.

5.1.1 Research Questions

We aim to answer the following research questions:

1. Will users perform better with the Color variant?
2. Will users perform better with the Width variant?
3. Will users perform better with the Predict Words variant?
4. Will the Shifted variant help reduce neighbor substitution errors?
5. Will the Size Invisible variant help reduce neighbor substitution errors?

5.1.2 Participants

Twenty participants, 13 males and 7 females, took part in the user study. All of the users' ages were between 19-30 years, except for one user that was 52 years old. Only 2 participants were left handed. All participants had a college degree, except one that had a high school degree. Every single participant had previous experience with QWERTY keyboards and uses it every day. Most participants (13) also use virtual QWERTY keyboards on a daily basis, 1 weekly, 4 rarely, and only

2 had never used them at all. Only 6 participants use a tablet at least weekly, while 13 use virtual keyboards on smartphones daily. Table 5.1 summarizes all demographic data of participants.

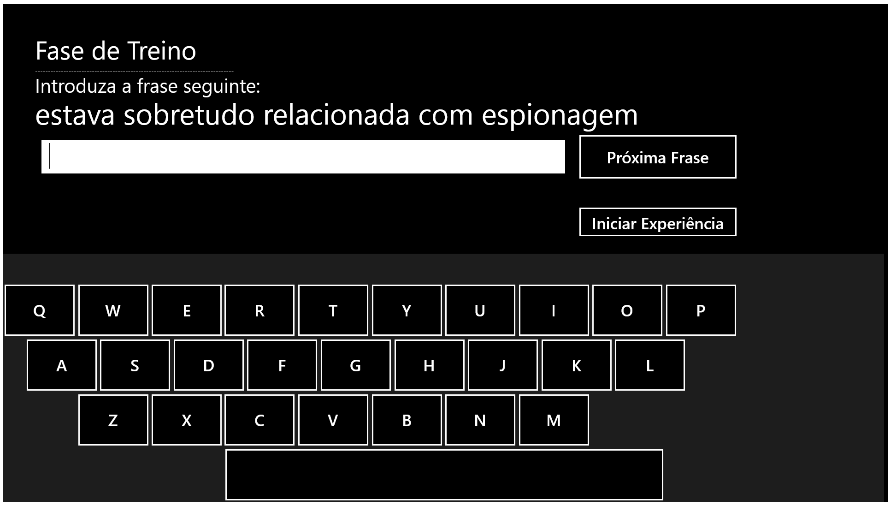
Participants	Age	Gender	QWERTY experience	Tablet experience	Smartphone experience
#1	[19, 30]	Female	high	none	none
#2	[19, 30]	Male	high	none	high
#3	[19, 30]	Male	high	none	none
#4	[19, 30]	Female	high	low	high
#5	[19, 30]	Male	high	low	high
#6	[19, 30]	Male	high	high	high
#7	[19, 30]	Male	high	none	high
#8	[19, 30]	Female	high	none	high
#9	[19, 30]	Male	high	none	low
#10	[19, 30]	Male	high	low	low
#11	[19, 30]	Male	high	none	high
#12	[19, 30]	Female	high	low	high
#13	[19, 30]	Female	high	high	high
#14	[19, 30]	Male	high	none	low
#15	[19, 30]	Male	high	high	high
#16	[19, 30]	Female	high	mid	none
#17	[19, 30]	Male	high	mid	high
#18	[51, 60]	Male	high	none	low
#19	[19, 30]	Female	high	none	high
#20	[19, 30]	Male	high	none	high

Table 5.1: Participants' profile.

5.1.3 Procedure

The user study had two main phases: training and evaluation. At the beginning of the first phase, we explained to each participant that the aim of the study was to evaluate each variant of the virtual QWERTY keyboard, and not the users themselves. Users were free to type in the position they found more comfortable: with one or two hands, with the tablet supported on the table, on the lap or on the free hand. Since participants were not familiar with the keyboard variants we developed, they were allowed to try each keyboard variant for two minutes, except *Shifted* and *Size Invisible* variants. These variants behaved visually just like the *QWERTY* condition, so users were not aware about their existence at this point.

The task in both phases consisted in copying a sentence that was displayed at the top of the screen Figure 5.1. After entering the sentence, the user could proceed to the next sentence by pressing the ”Próxima Frase” (”Next Sentence”) button. Copy typing was used to reduce the opportunity for spelling and language errors, and to make error identification easier. Both required and transcribed sentences were always visible. Sentences were randomly chosen from a set of 88 sentences (Appendix B.1) extracted from a Portuguese language corpus such that no sentence was written twice per participant. These were the same sentences used to perform the text prediction evaluation, which were extracted from another study [14]. Each sentence had five words with an average size of 4.48 characters and a minimum correlation with the language of 0.97. In order to avoid different correction strategies by the users, the backspace/delete keys were removed. Users were instructed to continue typing if an error occurred.



Fase de Treino

Introduza a frase seguinte:

estava sobretudo relacionada com espionagem

Figure 5.1: Screen shot of the evaluation application.

On the evaluation phase, participants were instructed to type the sentences as quickly and accurately as possible. Each user was asked to type 5 sentences for each variant, the first one being a practice trial. Before the test, users were informed that they would perform tests on 2 more variants that were only slightly different from QWERTY. During the evaluation users did not know whether they were using the *Shifted* or the *Size Invisible* variants, or even the traditional QWERTY. This way, we ensured that their typing pattern was not influenced by that knowledge. The order of conditions was counter balanced to avoid bias associated with experience. In the end, users were asked to answer a survey with some demographic data, as well as satisfaction regarding each variant. The whole process took approximately thirty minutes per participant.

5.1.4 Apparatus

A Samsung ATIV Smart PC Pro 11.6" was used in the user study. Each key had 20mm of width and 15mm of height. Visually, there is a space of 2mm between keys, horizontally and vertically. However, our implementation does not allow pressing between keys: each touch is always assigned to a key. This makes the keyboard more responsive, thus avoiding the frustration of performing a touch that does not produce a character. A letter was entered when the user lifted his finger from the key. All participants' actions were logged through the evaluation application, for posterior analysis.

5.1.5 Dependent Measures

Performance during the text-entry task was measured by several quantitative variables: *Words Per Minute* (*WPM*), *Minimum String Distance* (*MSD*) error rate, and character-level errors (substitutions - incorrect characters, insertions - added characters, and omissions - omitted characters) [60]. Qualitative measures were also gathered in the end of the experiment by debriefing each participant.

5.1.6 Design and Analysis

We used a within-subjects design where each user tested all conditions. For each keyboard condition each user entered 5 sentences (1 practice $+$ 4 test), resulting in a total of 30 sentences per user. In summary the study design was: 20 users $\times$ 5 sentences $\times$ 6 keyboards.

We performed Shapiro-Wilkinson tests of the observed values for *WPM*, *MSD* error rate, types of errors to assess if they were normally distributed. If they were, we applied parametric statistical tests, such as repeated measures ANOVA, t-test, and Pearson correlations. On the other hand, if measures were not normally distributed, we used non-parametric tests: Friedman, Wilcoxon, and Spearman correlations. Bonferroni corrections were used for post-hoc tests.

5.2 Results

In this section we analyze input speed and accuracy for the six conditions (*QWERTY* keyboard and *Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible* variants), focusing on type of errors.

5.2.1 Input Speed

In this subsection we analyze input performance regarding speed for each keyboard condition. To assess speed, we used the *Words Per Minute* (*WPM*) [61] text input measure calculated as:

$$(transcribed\, text - 1) \times (60\, seconds \div time\, in\, seconds) \div 5\, characters\, per\, word \qquad (5.1)$$

Figure 5.2 (a) illustrates WPM for each variant, while Table A.1 shows the participants' average *WPM* for *QWERTY*, *Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible* conditions.

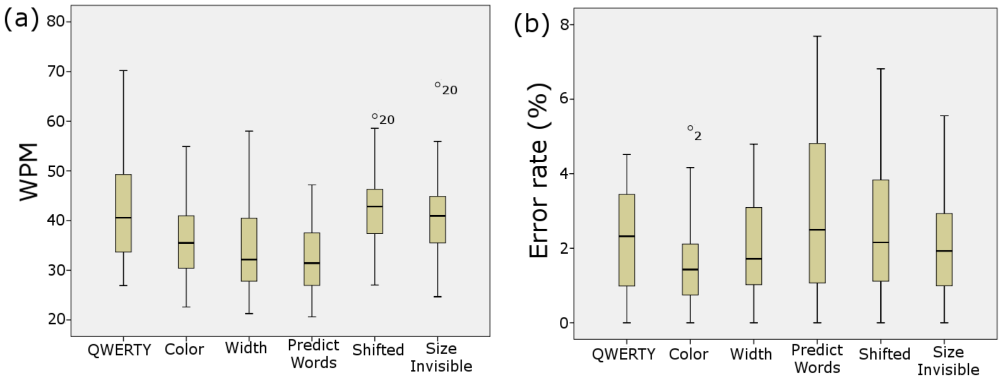
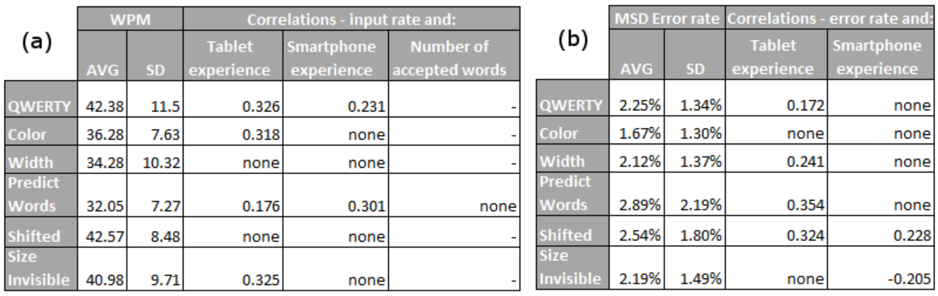


Figure 5.2: (a) Typing speed and (b) error rate for each variant with outliers.

Table 5.2 (a) shows the average and standard deviation of WPM for each variant, as well as the result of the (Spearman) correlations between input rate and: tablet experience, smartphone experience and number of accepted words (only relevant for *Predict Words*). Correlations were mostly moderate-to-weak and weak-to-low. In the stronger case it did not go beyond a moderate-to-weak correlation of 0.326, meaning that Tablet and Smartphone experience are not able to explain input speed.

Effect of the virtual keyboard on typing speed. A repeated measures ANOVA revealed significant differences between keyboard variants on text-entry speed $(F(5,\ 90) = 18.787\, p < 0.001)$. Bonferroni post-hoc tests showed significant differences between *QWERTY* and *Color*, *Width* and *Predict Words* variants, meaning that participants typed significantly slower in these 3 variants. This result was somewhat expected, since the visual changes can be distracting, which may reduce the typing speed for younger users acquainted with the QWERTY layout. The *Predict Words* variant is also slower than



(a)

	WPM		Correlations - input rate and:		
	AVG	SD	Tablet experience	Smartphone experience	Number of accepted words
QWERTY	42.38	11.5	0.326	0.231	-
Color	36.28	7.63	0.318	none	-
Width	34.28	10.32	none	none	-
Predict Words	32.05	7.27	0.176	0.301	none
Shifted	42.57	8.48	none	none	-
Size Invisible	40.98	9.71	0.325	none	-

(b)

	MSD Error rate		Correlations - error rate and:	
	AVG	SD	Tablet experience	Smartphone experience
QWERTY	2.25%	1.34%	0.172	none
Color	1.67%	1.30%	none	none
Width	2.12%	1.37%	0.241	none
Predict Words	2.89%	2.19%	0.354	none
Shifted	2.54%	1.80%	0.324	0.228
Size Invisible	2.19%	1.49%	none	-0.205

Table 5.2: Average and standard deviation of (a) WPM and (b) Error Rate for each variant, as well as correlations (Spearman, with n=20) between them and several dimensions.

the *traditional QWERTY*, which indicates that the saved keystrokes does not make up for the time and cognitive effort required to constantly check the suggestions list. Having said that, we expect these variants to help older users improve speed, since most of them might not be acquainted with the QWERTY layout. Therefore, if the *prediction system* suggests the right letter, the user will not need to waste time scanning all the keys. As expected, there were no significant differences between the input rate of the *traditional QWERTY* and the *Shifted* and the *Size Invisible* variants.

5.2.2 Quality of Transcribed Sentences

The quality of the transcribed sentences was measured using the *Minimum String Distance* (*MSD*) error rate, calculated as:

$$MSD(required\,text, transcribed\,text) \div Max(\mid required\,text \mid, \mid transcribed\,text \mid) \times 100 \qquad (5.2)$$

Figure 5.2 (b) illustrates *MSD* error rate for each variant, while Table A.2 illustrates participants' average *MSD* error rate for the *QWERTY* keyboard and its variants.

Table 5.2 (b) shows the average and standard deviation of error rate for each variant, as well as the result of the correlations between error rate and: tablet and smartphone experience. Correlations were mostly moderate-to-weak and weak-to-low. In the stronger case it did not go beyond a moderate-to-weak correlation of 0.354, meaning that tablet and smartphone experience are not able to explain error rate.

Effect of the virtual keyboard on quality of transcribed sentences. All variants slightly improved the overall quality of the typed sentences, since the error average was highest on *QWERTY*. However, a Friedman test did not reveal significant differences between keyboard conditions on error rate ($\chi^2(5) = 2.933,\ p = 0.710$). But, we must not forget that these results regard all types of errors. And, for instance, the *Shifted* and *Size Invisible* variants only aim to correct neighbor substitution errors. We will further analyze these variants in subsections 5.2.4 and 5.2.5, respectively. Furthermore, we performed a t-test between the *QWERTY* and the *Color* variant, since this was the variant with least errors. The t-test confirmed that there are significant differences between these variants on error rate ($t(17) = 3.151,\ p = 0.006$). This means that, despite the fact that participants were already familiarized with the QWERTY layout, they were committing fewer errors with this variant, especially omission errors. We will further analyze this variant in subsection 5.2.6.

5.2.3 Typing Errors

This section presents a fine grain analysis by categorizing the types of input errors: insertions - added characters; substitutions - incorrect characters; and omissions - omitted characters [60]). Figure 5.3 (a) shows the contribution of each type of error for the total amount of errors, on each variant. *Insertion* errors are the least common type of error on the *QWERTY* keyboard, *Width* and *Size Invisible* Variants. On the *Color* and *Shifted* variants this slot belongs to the omission errors. *Substitution* errors are consistently the most common error, across variants. This result differs from the result reported by Nicolau [14]; that is, the most common error among older adults were omission errors, which further emphasizes the differences between populations.

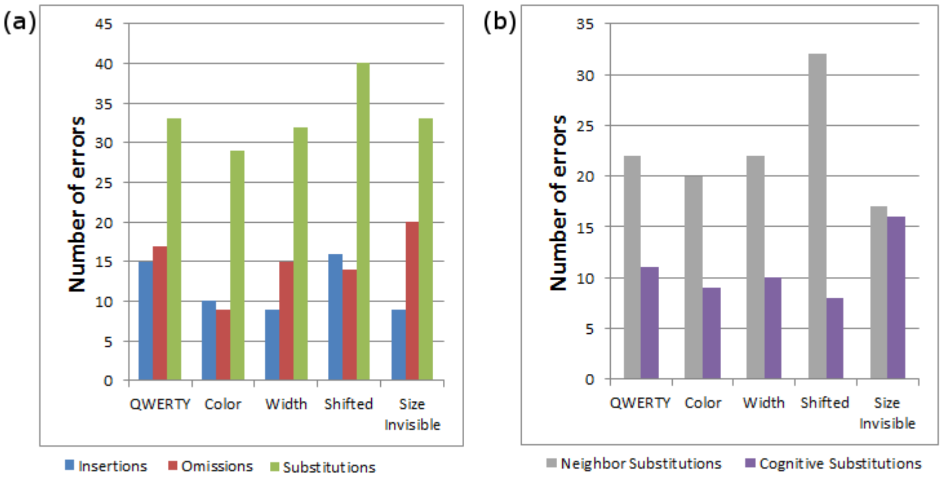


Figure 5.3: (a) Contribution of each type of error for the total amount of errors on each variant; (b) Number of neighbor and cognitive substitution errors on each variant.

5.2.4 Shifted variant

In this section we will present the results obtained with the Normal *Shifted* variant (10% shift) and the Improved *Shifted* variant (optimum shift). The latter obtained better results.

Normal Shifted variant. As we said previously (Subsection 4.3.5), the Shifted variant only aims to correct *neighbor substitution* errors, which occur when the user touches a key immediately adjacent to the expected key. Still, as we can see in Figure 5.3 (b) *neighbor substitutions* were more frequent on the *Shifted* variant than on the others. This is due to the vertical and horizontal shifts we applied to the touch points. Although other authors [14, 37, 59] reported that users generally touch on the bottom-right of targets, they never clarified what was the optimum shift to be applied in order to compensate the skewness. Therefore, these poor results were obtained, because we used a non-optimum value. Still, this variant was able to intervene correctly 56.41% of the times, correcting 13.51% of *neighbor substitutions*, when compared with the same input as if participants were typing on a QWERTY keyboard.

Improved Shifted variant. In order to find the optimum horizontal and vertical shifts, we calculated the necessary shift to transform each bad touch point into a good one, and the necessary shift that avoids transforming good touch points into bad ones. We were able to verify that sometimes shifts were contradictory; that is, good assigns happened when the touch point was skewed to the bottom-right, thus being corrected by the shift, if the real touch point was on a neighbor key. Bad assignments happened when the touch point was skewed to the top-left; for instance, several bad assignments happened when the real touch point was on the leftmost limit of the ”o” key (intended key), but when shifted, it was assigned to the ”i” key, wrongly. Therefore, the optimum value was found by maximizing good assignments and minimizing bad ones, which was shifting the touch points to the top 6% of the height of the key, and to the left 7% of the width of the key. This allowed the *Shifted* variant to intervene correctly 87.18% of the times, correcting 48.65% of *neighbor substitution* errors.

5.2.5 Size Invisible variant

In this section we will present the results obtained with the Normal *Size Invisible* variant (5% size increase) and the Improved *Size Invisible* variant (optimum size increase). The latter obtained better results.

Normal Size Invisible variant. Just like the *Shifted* variant, the *Size Invisible* variant only aims to correct *neighbor substitution* errors. As we can see in Figure 5.3 (b), *neighbor substitutions* were lower on the *Size Invisible* variant, suggesting that the variant was able to intervene correctly. As stated previously , this variant increases the height and width of the underlying area of the four most probable keys. Since this was our first study the values were set by experimentation; we did not know what would be the optimum size increase, since that was what we wanted to find. Therefore, we decided to increase 50% of the height of the key vertically, and 50% of the width of the key horizontally. Results show that this variant was able to intervene correctly 68.97% of the times, correcting 37.04% of *neighbor substitution* errors, when compared with the same input as if participants were typing on a QWERTY keyboard.

Improved Size Invisible variant. In order to find the optimum size increase, we calculated the necessary size increase to transform each bad touch point into a good one, and the necessary size increase that avoids transforming good touch points into bad ones. We were able to verify that sometimes, the size increase desired is contradictory, depending if the key the user wants to tap is in the four most probable keys or not. For instance, if the user wants to tap on the ”d” key (which is highlighted), but taps on the ”e” key, we need a high size increase, to accept the touch input as a ”d” key. But, for instance if the user taps on the ”e” key (intended key), but the ”d” key is in the 4 most probable keys (and ”e” is not), we need a low increase size, in order to accept the touch input as an ”e” key. Therefore, the optimum value was found by maximizing good assigns and minimizing bad assigns, which was increasing the size of the key 37% of the height of the key vertically, and 21% of the width of the key horizontally, maintaining the rounded corners. This allowed the *Shifted* variant to intervene correctly 93.10% of the times, correcting 62.96% of *neighbor substitution* errors.

The remaining substitution errors were not corrected because: (1) it was at the beginning of a word (11.11%). In such case the *Size Invisible* variant has no key highlighted; (2) the user had already committed a mistake (14.81%), therefore the *prediction system* was not able to highlight any key (or at least, not the right key); (3) the variant intervened wrongly by assigning the touch point to other key (highlighted) than the intended (7.41%); and, (4) the variant did not made a correction because the intended letter was not in the most likely list (3.7%).

5.2.6 Color variant

When looking at Figure 5.3 (a) we can verify that omission errors were lower on the *Color* variant. Omissions are most likely to occur when users miss a key or when their finger slips (they press one key and release on another, generating no output). We verified that this type of error is most frequent on the space bar (47% of all the omissions are spaces, on the *QWERTY* condition). This happened because the space bar is located at the bottom of the touchscreen, and sometimes users completely missed the touch area captured by the tablet, hitting its bevel instead. On the *Color* variant, when participants missed the space bar, they were able to detect it because the key remained highlighted, indicating that the key was not correctly pressed. As a matter of fact, space *omissions* were lowered to only 33% of all *omissions* on the *Color* variant.

5.2.7 Width variant

The *Width* variant was not so popular between the users. Their performance on the *Width* variant regarding error rate was comparable to *QWERTY's*. This was achieved at the cost of reducing significantly the typing speed. Still, participants' comments were mostly negative, because the keys were always changing position which highly increased the cognitive effort to not commit mistakes.

5.2.8 Predict Words variant

Since this variant behaved mainly as the *QWERTY* keyboard, we will only focus on the different characteristics and types of errors that emerged from using this variant. Participants accepted 32.50% of the words they could have accepted. As previously stated, this variant was slower than the *traditional QWERTY*, which indicates that the saved keystrokes do not make up for the time and cognitive effort required to constantly check the suggestion's list. Regarding errors, there were 18 errors that were specific of the use of the *Predict Words* variant. Although participants were instructed at the beginning of the test that, after accepting a suggested word a space would be automatically inserted, participants forgot this several times, and inserted another one (11 times). Also, when trying to accept the suggested word at the top of the keyboard, two participants tapped on the ”q” key instead. Therefore, the resulting word would contain the beginning of the word they had written and the letter ”q” attached to the end. The second problem can be easily solved by increasing the size of the area that allows accepting the suggested words.

5.2.9 Touch Typing Patterns

Even though the *Shifted* variant was able to correct 48.65% of the substitution errors in the optimized version, we did not verify the bottom-right touch pattern described by Nicolau [14]. The author hypothesized that the bottom-right pattern was related with hand dominance, since in his study users only interacted with their right hand. In our study, 18 users used both hands, while 2 users used only their right hand.

Even when analyzing the patterns individually for each group, no clear pattern seems to emerge. Figure 5.4 (a) and 5.4 (b) shows the key deviation for participants who interacted only with their right hand and participants who used both hands, respectively. In order to have more data (and thus more precise results), we considered the data from *QWERTY*, *Shifted* and *Size Invisible* variants, not taking into account the corrections performed by the latter two (i.e., all the data was treated like typing on a *traditional QWERTY*).

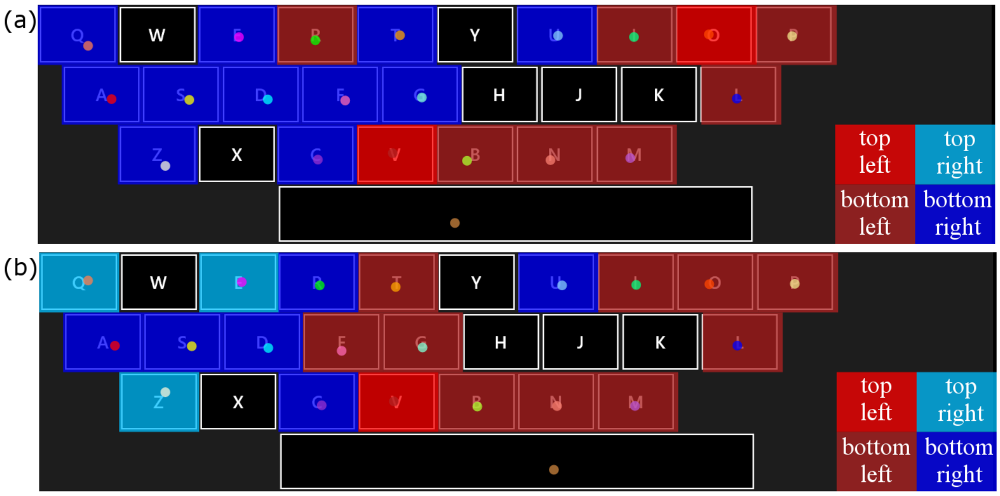


Figure 5.4: Key deviation from the center of the key for (a) participants who interacted only with their right hand and (b) participants who interacted with both hands.

When analyzing the individual key deviation of some participants who used only their right hand (Figure 5.5), we can see that their pattern is contradictory; while the pattern of participant #12 is in accordance with the bottom-right pattern reported by Nicolau [14] (except for the "o" key), participant's #16 is not. We do not know if participant #16 is a special case, or if indeed, every user has their own typing behavior, since only these two participants interacted only with their right hand.

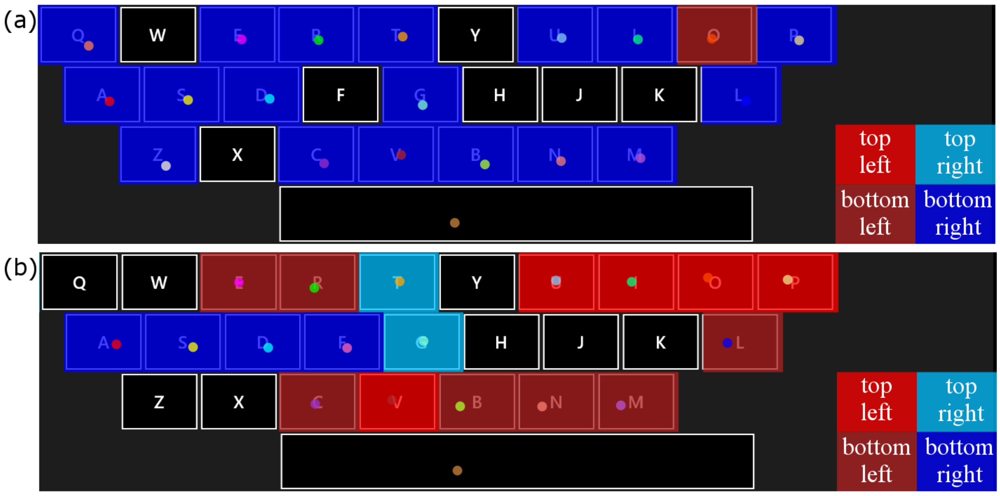


Figure 5.5: Key deviation from the center of the key for participants (a) #12 and (b) #16 who interacted only with their right hand.

Regarding participants' who used both hands to type (Figure 5.4 (b)), there seems to be an overall tendency to touch on the bottom-right side of the keys on the left side of the keyboard, and touch on the bottom-left side of the keys on the right half of the keyboard. But, when analyzing the average deviation from the center of the key of each user, we found that this deviation is strongly user-dependent. For instance, the key deviation for participants #3 (Figure 5.6 (a)) and #20 (Figure 5.6 (b)) are completely different, even though both used both hands to type.

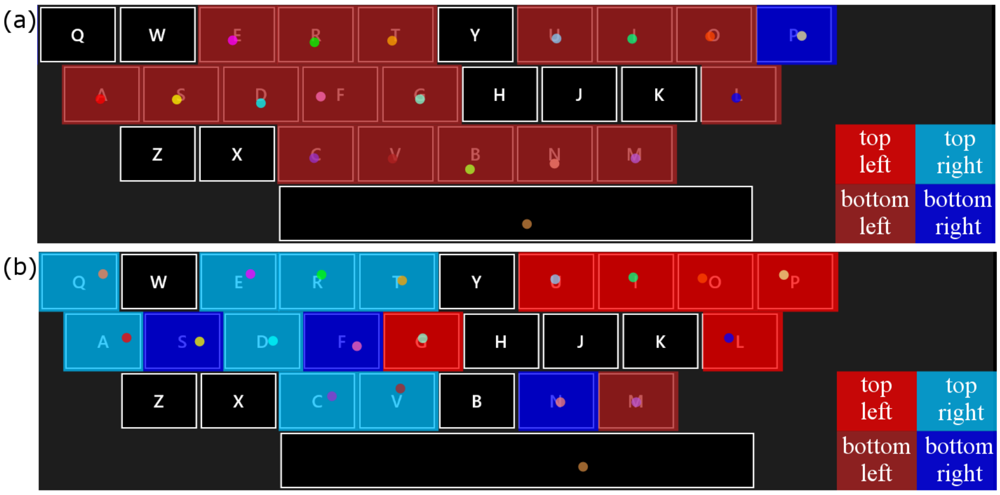


Figure 5.6: Key deviation from the center of the key for participants (a) #3 and (b) #20 who interacted with both hands.

Therefore, an adaptive model that constantly updates the center of each individual key seems to be the best solution to correct the *neighbor substitution* errors, without resorting to a predictive system.

5.2.10 Participants' Preferences, Comments and Observations

At the end of the user study participants were debriefed and asked to answer a satisfaction survey. We also collected comments during and after the test about their opinion regarding the several keyboard variants. The questions were only regarding the *QWERTY*, *Color*, *Width* and *Predict Words* variants, since users were not aware of when they were using the *Shifted* and *Size Invisible* variants. In general, users were satisfied and found easy to use the *QWERTY*, *Color* and *Predict Words* variants. Regarding the *Width* variant, users said it was difficult to use and were not happy using it. They commented that the fact that the keys were constantly changing width was visually confusing, and due to this they found harder to locate, aim and press a particular key. Some users reported that it was better not to look at the keyboard while typing, which made it harder to aim properly.

When comparing each variant to *QWERTY*, users said that, on average, the *Color* and *Predict Words* variants were useful. The *Width* variant obtained very scattered results in this question. However, on average, users felt it was unhelpful.

In a 1-5 scale, where 5 is the lowest cognitive effort, the variant that required the lowest cognitive effort for users was *QWERTY* ($Mean = 4.15;\ SD = 0.81$). The *Color* ($Mean = 3.6;\ SD = 0.88$) and *Predict Words* ($Mean = 3.25;\ SD = 1.16$) variants were also rated as somewhat low. The *Width* variant showed worse results ($Mean = 2.5;\ SD = 1.15$). When asked about the easiness of finding a particular letter, users found it easy on the *QWERTY* and *Color* variant, averaging the same. The *Width* variant had the worst results again; users said it was relatively difficult to find a particular letter.

Despite the fact that participants were slower and made the same amount of errors on the *QWERTY* keyboard as when using the *Predict Words* variant, they classified it as useful and easy to use. However, the *QWERTY* averaged better than other variants in satisfaction and easiness to use, which indicates that users prefer a visually static keyboard, as similar as possible to the physical ones.

Several participants expressed their dislike for the *Width* variant, which is in conformity with the results of the survey. Also, some users emphasized that the *Color* variant acted as a positive reinforcement when they were tapping a key that was highlighted.

5.3 Discussion

Our goal was to investigate how users inputted text in tablet devices in order to improve their performance. In this section we discuss the obtained results by: 1) answering the previously proposed research questions; and 2) identifying implications for design.

5.3.1 Answering Research Questions

1. *Will users perform better with the Color variant?*
   The *Color* variant was able to reduce significantly the error rate, when compared with the QWERTY keyboard. This result was obtained at the cost of reducing the input rate significantly. Still, participants were satisfied and found the *Color* variant easy to use.

2. *Will users perform better with the Width variant?*
   Young users performed roughly the same, regarding error rate, when comparing the *Width* variant with the *QWERTY* keyboard. Input rate was reduced significantly, even more than the *Color* variant. Furthermore, users found it difficult to use and were not happy using it.

3. *Will users perform better with the Predict Words variant?*
   Predict Words had the worst results regarding error rate and input rate. Still, participants were satisfied and found it easy to use.

4. *Will the Shifted variant help reduce neighbor substitution errors?*
   The optimized *Shifted* variant was able to reduce neighbor *substitution* errors, although not as much as it would be expected. This happened because the *Shifted* variant assumes a bottom-right touch pattern that was not verified for most of the young users. This means that users would benefit more with an adaptive model that constantly updates the center of each individual key. As it was expected the input rate was maintained unaltered.

5. *Will the Size Invisible variant help reduce neighbor substitution errors?*
   The improved *Size Invisible* variant was able to reduce a high number of *neighbor substitution* errors. This variant is able to overcome the fact that young users do not have a bottom-right touch pattern, since the four most probable keys increase their size in all directions. This means that, even if the touch point is at the top left of the intended key and the key is highlighted, the touch point will be assigned to that key.

5.3.2 Design Implications

Allow personalization. Since touch typing behavior is completely user-dependent, the best approach is to have an adaptive model that constantly updates the center of each key.

Use a language model to increase the underlying area of the most probable keys. Since no touch typing pattern emerged, a model that increases the underlying area of the four most probable keys will help to decrease *neighbor substitution* errors. The model will be able to ensure that, independently of the position of the touch point (top left, top right, bottom left or bottom right), it will be assigned to the intended key, as long as the intended key is highlighted.

5.3.3 Limitations

The user study reported in this chapter does not contemplate error correction. While this was necessary to assess natural typing patterns, understanding users' correcting strategies is also needed. Further research should focus on reporting error correction effect on touch-based devices.

Participants interacted mainly with both hands (18). Following studies should also focus in users that only use their left or right hand, in order to verify if different touch typing patterns emerge.

5.4 Conclusion

We investigated text-entry performance of 20 young adults on a touch-based device (tablet). Our user study featured six virtual keyboard conditions (*traditional QWERTY*, *Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible*). Users typed significantly slower with the *Color*, *Width* and *Predict Words* variants, which indicate that young users are faster with keyboards that are visually static. Regarding the quality of transcribed sentences, significant differences were found between the *QWERTY* keyboard and *Color* variant, meaning that fewer errors were committed on the *Color* variant. No significant differences were found on the remaining conditions. Still, after finding the optimum shift for the *Shifted* variant, and the optimum increase size for the *Size Invisible* variants, these variants were able to reduce the *neighbor substitution* errors substantially.

The most common type of error for the younger users are the *substitution* errors, followed by *omissions* and *insertions*, except on the *Color* and *Shifted* variants, where the *substitutions* were followed by *insertions* and then *omissions*.

We found that touch typing patterns are completely user-dependent, regarding young users. This means that users will benefit more with an adaptive model that constantly updates the center of each individual key.

Lastly, we identify some design implications that should improve typing accuracy and encourage researchers to create more effective solutions for young adults. Future research should apply the design implications described here and investigate their effect on text-entry performance.

6

Study with Older Adults

The results obtained in the baseline study described in the previous Chapter, show that the *Color*, *Shifted* and *Size Invisible* variants are the most promising variants. Therefore, one of the aims of this study is to verify if these variants are indeed advantageous for the older users.

In this Chapter we will describe the user study we conducted with 20 older users and three different virtual keyboards - *QWERTY traditional* keyboard, and *Color* and *Predict Words* variants. After the evaluation we also performed six different simulations (using the data from the QWERTY keyboard condition) that aimed to correct different kinds of errors. This Chapter aims to provide the knowledge needed to design text entry solutions more adequate for older adults. We describe users' typing behaviors and performance errors, as well as their comments. We also analyze thoroughly the errors that were corrected by each type of simulation.

6.1 User Study

Accurately selecting targets on current touchscreen devices can be a hard task to accomplish for older adults. In this user study we evaluate the performance of the *Color* and *Predict Words* variants when compared with the *QWERTY* keyboard. We dropped the *Width* variant tested in the younger users study because we did not want to overload the older users with too many tasks. Since that variant performed worse on the previous study, we thought it was the right one to omit. That is the same reason why we decided to leave the *Shifted* and *Size Invisible* variants out of the test. But, since these variants only operate in the background, that is, without performing changes visually, we were able to simulate the usage of those variants using the data from the QWERTY keyboard condition.

6.1.1 Research Questions

1. Will inexperienced older users who are not acquainted with the QWERTY keyboard layout perform better with the *Color* variant?
2. Will older users perform better with the *Predict Words* variant?
3. How do older adults type on touchscreens regarding accuracy, speed and typing strategies?
4. What are the most common types of errors committed by older adults?
5. Does tremor affect text-entry performance? If yes, how is users' performance correlated with hand tremor?

6.1.2 Participants

Twenty participants, 15 females and 5 males, took part in the user study. Their age ranged from 61 to 92 years old, with the most prevalent age group being from 81 to 90 years old. All participants were right handed, although two actually interacted only with their left hand. They were recruited from several local social institutions. None of the participants had severe visual impairments; all participants reported that they were able to read the screen content without difficulties.

None of the participants had ever experienced touchscreen devices. Regarding QWERTY familiarity, several participants had used this type of keyboard whether in typewriters (8 participants) or personal computers (14 participants). Table 6.1 summarizes all demographic data of participants. The task-specific tremor column (Table 6.1) ranges from 0 to 4. The lowest value indicates absence of tremor (0) while the highest value indicates marked tremor (4).

Participants	Age	Gender	QWERTY experience	Touchscreen experience	Visual characteristics	Task-specific tremor (right/left hand)
#1	[81, 90]	Male	high	none	normal	2/2.33
#2	[81, 90]	Female	mid	none	normal	0.33/1
#3	[81, 90]	Female	mid	none	normal	2/3
#4	[71, 80]	Female	none	none	normal	0.67/0.67
#5	[81, 90]	Female	high	none	cataracts	0/1
#6	[61, 70]	Female	mid	none	normal	2/1.33
#7	[71, 80]	Male	high	none	normal	1/2
#8	[81, 90]	Female	low	none	normal	0.33/1.67
#9	[61, 70]	Female	low	none	cataracts	1.33/2.33
#10	[61, 70]	Male	low	none	normal	0/1
#11	>91	Female	low	none	normal	1.33/1.33
#12	[71, 80]	Female	low	none	normal	2.33/2.33
#13	[81, 90]	Female	low	none	normal	1.33/2.33
#14	[71, 80]	Female	low	none	no glasses	1.33/1.33
#15	[81, 90]	Female	none	none	cataracts and strabismus	1.67/2.33
#16	[71, 80]	Female	low	none	normal	1.33/1.33
#17	[81, 90]	Female	none	none	normal	0.67/1
#18	[71, 80]	Female	low	none	normal	0.67/1.67
#19	[61, 70]	Male	high	none	normal	0.33/1
#20	[71, 80]	Male	high	none	normal	0.33/1.33

Table 6.1: Older participants' profile.

6.1.3 Procedure

The user study had two main phases: training and evaluation. At the beginning of the first phase, we explained to each participant that the aim of the study was to evaluate each variant, and not the users themselves. We then explained and exemplified to them how to use a virtual keyboard and its variants. Still, users were free to type in the position they found more comfortable: with one or two hands, with the tablet supported on the table, on the lap or on the free hand. Since participants were not familiar with touch devices and the QWERTY variants we developed, they were allowed to type two sentences per keyboard variant during this phase. If by the end of the two trial sentences participants did not fully understand the keyboard variant, we would let them type another sentence.

The task in both phases consisted in copying a sentence that was displayed at the top of the screen. After entering the sentence, the user could proceed to the next sentence by pressing the ”Próxima Frase” (”Next Sentence”) button. Copy typing was used to reduce the opportunity for spelling and language errors, and to make error identification easier. Both required and transcribed sentences were always visible. The sentences were chosen randomly from a set of 88 sentences (Appendix B.1) extracted from a Portuguese language corpus, such that no sentence was written twice per participant. This was the same set of sentences [14] used to perform the text prediction evaluation described in Section 4.2. Each sentence had five words with an average size of 4.48 characters and a minimum correlation with the language of 0.97. In order to avoid different correction strategies by the users, the delete key was removed. Participants were instructed to continue typing if an error occurred.

On the evaluation phase, participants were instructed to type the sentences as quickly and accurately as possible. Each user was asked to type 5 sentences for each variant, the first one being a practice trial. The order of conditions was counter balanced to avoid bias associated with experience. In the end, users were asked to answer a survey with some demographic data, as well as satisfaction regarding each variant. We also assessed users’ capabilities regarding task-specific tremor, by asking them to draw a spiral with each hand without leaning hand or arm on the table [62]. The whole process took approximately 1 hour per participant.

6.1.4 Apparatus

A Samsung ATIV Smart PC Pro 11.6” was used in the user study. Each key had 20mm of width and 15mm of height. Visually, there is a space of 2mm between keys, horizontally and vertically. However, our implementation does not allow pressing between keys: each touch is always assigned to a key. This makes the keyboard more responsive, thus avoiding the frustration of performing a touch that does not produce a character. A letter was entered when the user lifted his finger from the key. All participants’ actions were logged through the evaluation application, for posterior analysis.

6.1.5 Dependent Measures

Performance during the text-entry task was measured by several quantitative variables: *Words Per Minute* (*WPM*), *Minimum String Distance* (*MSD*) error rate, and character-level errors (substitutions - incorrect characters, insertions - added characters, and omissions - omitted characters) [60]. Qualitative measures were also gathered in the end of the experiment by debriefing each participant. We also gathered one tremor-related measure from each participant after the text-entry task: Archimedes spiral test (action tremor) [62].

6.1.6 Design and Analysis

We used a within-subjects design where each user tested all conditions. For each keyboard condition each user entered 5 sentences (1 practice + 4 test), resulting in a total of 15 sentences per user. In summary the study design was: 20 users × 5 sentences × 3 keyboards.

We performed Shapiro-Wilkinson tests of the observed values for *WPM*, *MSD* error rate, types of errors and tremor measures to assess if they were normally distributed. If they were, we applied parametric statistical tests, such as repeated measures ANOVA, t-test, and Pearson correlations. On the other hand, if measures were not normally distributed, we used non-parametric tests: Friedman, Wilcoxon, and Spearman correlations. Bonferroni corrections were used for post-hoc tests.

6.2 Results

In this subsection we describe and characterize each user's tremor profile and relate it with text-entry performance. Moreover, we analyze input speed and accuracy for the three conditions (*QWERTY* keyboard, *Color* and *Predict Words*), focusing on type of errors and main causes.

6.2.1 Tremor Profile

Regarding tremor, we measured task-specific tremor (a type of action tremor) in both hands, using the Archimedes spiral test [62]. Since we did not find an expert who could classify the drawings for us, we opted to ask three different observers to classify the drawings, by visually comparing the drawings performed by the participants with examples of spiral drawings from other study [14] (Figure 6.1). For instance, if to the eyes of the classifier, the drawn spiral was similar to a spiral classified as ”slight” on Nicolau's [14] study, the same classification should be attributed by the classifier. While this is not the best way to assess user tremor, we are confident that by having different classifiers corroborating each other's scores we have reached trustworthy results.

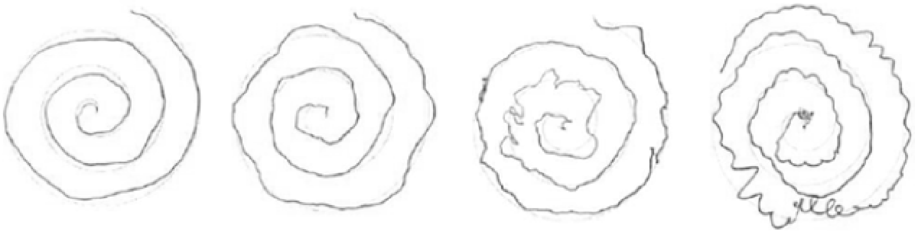


Figure 6.1: Examples of Archimedes spiral drawings classified as Absent, Slight, Severe and Marked, respectively.

Classifications could be one of five: *Absent* (0), *Slight* (1), *Moderate* (2), *Severe* (3) and *Marked* (4). After the classification, we proceeded with a *Cronbach's alpha* test to verify if the scores from the different classifiers were consistent. The *Cronbach's alpha* for the right and left hand is 0.890 and 0.901, respectively, which indicates a high level of internal consistency (the highest value of internal consistency is 1.0). To obtain the final score we calculated the average of the three observations. For the right-hand drawings, 45% of the participants had a score in the [0, 1[interval, 35% in the [1, 2[, 20% in the [2, 3[, and 0% in both [3, 4[and 4 ([4]). Regarding the left hand drawings, 5% of the participants had a score in the [0, 1[interval, 60% in the [1, 2[, 30% in the [2, 3[, 5% in [3, 4[and 0% in the 4 ([4]). Column ”Task Specific Tremor” from Table 6.1) summarizes the subjective tremor assessment.

6.2.2 Input Speed

In this section we thoroughly analyze input performance regarding speed for each keyboard condition (*QWERTY*, *Color* and *Predict Words*). To assess speed, we used the *Words Per Minute* (*WPM*) [61] text input measure calculated as described in Chapter 5 (equation 5.1).

Figure 6.2 (a) and Figure A.9 (a) illustrate WPM for each variant without and with outliers, respectively, while Table A.3 (a) shows the participants' average *WPM* for *QWERTY*, *Color* and *Predict Words* conditions. Outliers (#2, #5 and #17) were found using the *labeling rule* [63]. They were removed to perform the ANOVA analysis.

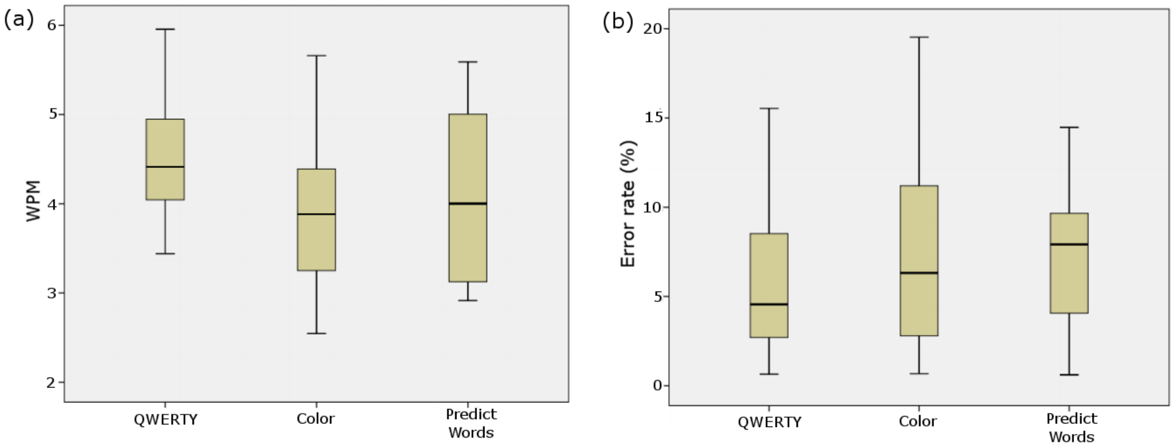


Figure 6.2: Participants' (a) WPM and (b) error rate for each variant without outliers.

QWERTY keyboard. Participants obtained a mean 6.19 $(SD = 3.92)$ wpm using the QWERTY keyboard. It is important to notice that participants #1, #5, #7 and #19 are well above the average (Table A.3 (a)). All these participants were experienced with QWERTY keyboards, either because they used typing machines, computer keyboards or both, in the past. Also, these participants used both hands to type during the tests. These two characteristics combined can explain the high input rate when compared to other users. Indeed there was a moderate positive correlation between QWERTY experience and input rate $[Spearman\ rho = .672,\ n = 20,\ p < .01]$, as well as between number of hands used and input rate $[Spearman\ rho = .651,\ n = 20,\ p < .01]$; that is, users that used a QWERTY keyboard before, and used two hands to type, inputted text faster with the QWERTY keyboard. We found a moderate-to-weak negative correlation between input rate and task-specific tremor for the right hand $[Spearman\ rho = .307,\ n = 18,\ p = .215]$, and no correlation for the left hand.

Color variant. Participants obtained a mean 5.42 $(SD = 3.60)$ wpm using the *Color* variant. The overall pattern is similar to the *QWERTY* keyboard; that is, participants with high input rate (#1, #5, #7 and #19) were experienced with QWERTY keyboards and used both hands to type. But, in this specific case, we found a moderate-to-low positive correlation between QWERTY experience and input rate $[Spearman\ rho = .349,\ n = 20,\ p = .131]$, and a strong positive correlation between number of hands used and input rate $[Spearman\ rho = .711,\ n = 20,\ p < .01]$; that is, participants that used a QWERTY keyboard before, and used two hands to type, inputted text faster with the *Color* variant. No correlation was found between input rate and task-specific tremor for the both hands.

Predict Words variant. Participants obtained a mean 5.51 $(SD = 3.29)$ wpm using the *Predict Words* variant. The overall pattern is similar to the *QWERTY* keyboard; that is, experienced participants who used both hands (#1, #5, #7 and #19) obtained higher input rates. In fact, we found a moderate positive correlation between QWERTY experience and input rate $[Spearman\ rho = .541,\ n = 20,\ p < .05]$, as well as a strong positive correlation between number of hands used and input rate $[Spearman\ rho = .731,\ n = 20,\ p < .01]$; that is, participants that used a QWERTY keyboard before, and used two hands to type, inputted text faster with the *Predict Words* variant. No correlation was found between input rate and number of words accepted from the suggested list; that is, there is no evidence that users that accepted words from the suggested list were faster than users that did not. Furthermore, no correlation was found between input rate and task-specific tremor for both hands.

Overall Results. A repeated measures ANOVA revealed significant differences between keyboard variants on text-entry speed $(F(2,\ 30) = 3.835,\ p < 0.033)$. Bonferroni post-hoc tests showed significant differences between *QWERTY* and *Color* variant, meaning that users type significantly slower with the *Color* variant. This result was not expected since our hypothesis was that, inexperienced users who are not acquainted with the QWERTY layout would benefit with the *Color* variant. We believe that the main reason for the low input rate in the *Color* variant is because the highlighting of the keys was distracting. However, no user reported this. We also noted that, in some cases, despite the correct letter being the only one highlighted by the *Color* variant, some older adults took a long time to find it. This means that some older adults were not paying enough attention to the highlighted keys, which excluded them from the benefits of letter suggestion. Overall, input rate was positively correlated with QWERTY experience and number of hands used.

Regarding the *Predict Words* variant there was no significant difference when compared with the *QWERTY* keyboard. Still, we must take into account that only 7 of the 20 participants accepted at least one suggested word from the list during evaluation. This means that the remaining 13 participants used the *Predict Words* variant as a normal QWERTY keyboard. Still, we did not find a correlation between text-entry speed on *Predict Words* variant and interaction methodology, i.e., if the participant accepted suggested words or typed as a normal QWERTY keyboard.

6.2.3 Quality of Transcribed Sentences

The quality of the transcribed sentences was measured using the *Minimum String Distance* (MSD) error rate, calculated as described in Chapter 5 (equation 5.2).

Figure 6.2 (b) illustrates Error Rate for each variant, while Table A.3 (b) illustrates participants' average *MSD* error rate for the *QWERTY*, *Color* and *Predict Words* conditions.

QWERTY keyboard. Participants achieved an average MSD error rate of 9.76% $(SD = 11.05\%)$ with the *QWERTY* keyboard. The error rate varied greatly from participant to participant (Table A.3 (b)). Fifteen participants (#1, #3, #4, #6-#11, #13-#16, #18 and #19) achieved a mean MSD error rate between 0% and 10%, 2 participants (#12 and #20) obtained results between 10% and 20%, one participant (#5) between 20% and 30% and two participants (#2 and #17) between 30% and 45%. Opposed to the results obtained on input speed, no correlation was found between MSD error rate and QWERTY experience, as well as between MSD error rate and number of hands used. Which means that, contrary to input speed, the quality of transcribed text cannot be explained by previous experience with keyboards and number of hands used. Moreover, no correlation was found between the MSD error rate and task-specific tremor for the right and left hands.

Color variant. Participants achieved an average MSD error rate of 10.30% $(SD = 9.16\%)$ with the *Color* variant. The error rate varied greatly from participant to participant. Twelve participants (#4, #6-#11, #13, #15, #16, #18 and #19) achieved a mean MSD error rate between 0% and 10%, 5 participants (#1, #3, #12, #14 and #20) obtained results between 10% and 20%, one participant (#2) between 20% and 30% and two participants (#5 and #17) between 30% and 40%. As in the *QWERTY* keyboard condition no correlation was found between MSD error rate and QWERTY experience, between MSD error rate and number of hands used, and between MSD error rate and task specific tremor for both hands.

Predict Words variant. Participants achieved an average MSD error rate of 9.61% $(SD = 7.68\%)$ with the *Predict Words* variant. The error rate varied greatly from participant to participant. Fourteen participants (#3, #4, #6, #7, #9, #10, #11, #13-#16 and #18-#20) achieved a mean MSD error rate between 0% and 10%, 3 participants (#1, #8 and #12) obtained results between 10% and 20% and three participants (#2, #5 and #17) between 20% and 30%. As in the *QWERTY* and *Color* conditions no correlation was found between MSD error rate and QWERTY experience, between MSD error rate and number of hands used, between MSD error rate and interaction methodology (i.e., if the participant accepted suggested words or typed as a normal QWERTY keyboard) and between MSD error rate and task specific tremor for both hands.

Overall Results. A repeated measures ANOVA did not reveal significant differences between keyboard variants on MSD error rate $(F(2,\ 32) = 1.044,\ p = 0.364)$. We expected both *Color* and *Predict Words* variants to outperform the *QWERTY* keyboard regarding MSD error rate. Although, we are not sure why the *Color* variant did not outperform the *QWERTY* keyboard, several situations occurred that can justify the obtained results. For instance, participant #20 was expected to type ”cooperantes” but ended up typing ”cooperacao”. This happened because the *Color* variant suggested the sequence of letters that lead to ”cooperacao”. The participant tapped the suggested letters without thinking too much; so he ended up writing an undesired word. This is an issue related with the *prediction algorithm*. Since the *prediction system* does not always suggest the right letter, the user still has to pay attention to the suggested letters. Also related with this, is a trust issue. Do users think that the system is correct most of the times? Are they completely capable of ignoring the suggested letters and select one of their own? Sometimes it seemed as if participants were afraid of tapping a certain key if the system was not suggesting it, especially after tapping a sequence of keys suggested by the system. The performance of the *Color* variant can also be explained by the fact that older users committed more errors than younger users, in general. This means that the *Color* variant cannot suggest as much correct letters as it suggested to the young users, because once there is an error, the *prediction system* is not able to correctly predict the sequence of letters intended by the user.

The *Predict Words* variant did not outperform the *QWERTY* keyboard regarding MSD error rate mainly because most participants (13) used it as a *QWERTY* keyboard. From the remaining seven, only three (#9, #10 and #19) accepted a high number of suggested words. From these three, participants #9 and #19 had worst results in the *Predict Words* variant when compared with *QWERTY*. This happened because sometimes when accepting a suggested word (located at the top of the keyboard) users tapped below the intended area, selecting a key from the top row of the keyboard instead. Another common error is to tap the space bar after accepting a suggested word. This counts as an insertion error because after accepting the suggested word a space is automatically inserted. Therefore, the use of the *Predict Words* backfired because participants ended up making mistakes they would not make in other situations.

6.2.4 Typing Errors

This section presents a fine grain analysis by categorizing the types of input errors. Our categorization is based on MacKenzie's et al. [60] categorization (*insertion*, *substitution* and *omission*), but with added subcategories in order to suit our needs.

First we have *insertion* errors, which are added characters. Depending on the nature of the insertion it can be classified as one of the next four different subcategories:

Accidental Insertion - an *accidental insertion* occurs when the user presses a key accidentally. This type of insertion is characterized by a reduced time interval between the press and release of the key;

Double Insertion - a *double insertion* occurs when the user inserts a repeated character with a reduced time interval between the release of the first key and the press of the second key;

Cognitive Insertion - a *cognitive insertion* occurs when the user inserts a character other than the expected and the elapsed time is unlike *accidental* and *double insertion* errors;

Extra-finger Insertion - an *extra-finger insertion* occurs when the user uses more than one finger at the same time to interact with the tablet. It is characterized by interleaved presses/releases of different touch points, i.e., when the press of a second key occurs without the release of the first key;

Then, we have *substitution* errors, which are incorrect characters. Depending on the nature of the substitution it can be classified as one of the next two different subcategories:

Neighbor Substitution - a *neighbor substitution* is an incorrect character immediately adjacent to the expected character. It was only considered a neighbor substitution error, if the touch point was in the nearest half of the adjacent key;

Cognitive Substitution - a *cognitive substitution* is an incorrect character the user inserts instead of the expected one. Sometimes it is related with similar representations of the letter (e.g.: p and q);

We also have *omission* errors, which are omitted characters. Depending on the nature of the omission it can be classified as one of the next three different subcategories:

Failed Omission - Our keyboard only contained keys important to our text entry task. Therefore, keys such as Tab, Shift and CTRL were not present, leaving empty spaces. A *failed omission* occurs when users try to tap a key at the edge of the keyboard (Q, A, Z, M, L, P and $SPACE$), but tap an empty space instead.

Slide Omission - a *slide omission* occurs when no output is generated, because the release action occurred in a different key when compared with the press action;

Cognitive Omission - a *cognitive omission* occurs when the user simply forgets to insert a given wanted character;

The last type of error is not specifically associated with any of the other three categories:

Empty - this type of "error" occurs when the user presses an empty area, i.e., an area without an associated key. In our implementation, this is not problematic since a press on an empty area does not generate output. However, in the final implementation of the keyboard, those areas will be assigned to the respective keys. Therefore, in this context, an *empty* error is a potential error.

QWERTY keyboard. In Figure 6.3 (a) we can verify that *insertions* were the most common errors committed by older adults, accounting for 54.50% of the total errors. This kind of error is unevenly distributed through all the users. For instance, users #2 and #17 are responsible for more than half (62%) of the *insertions*. No correlation was found between *insertion* errors and task-specific tremor for both hands.

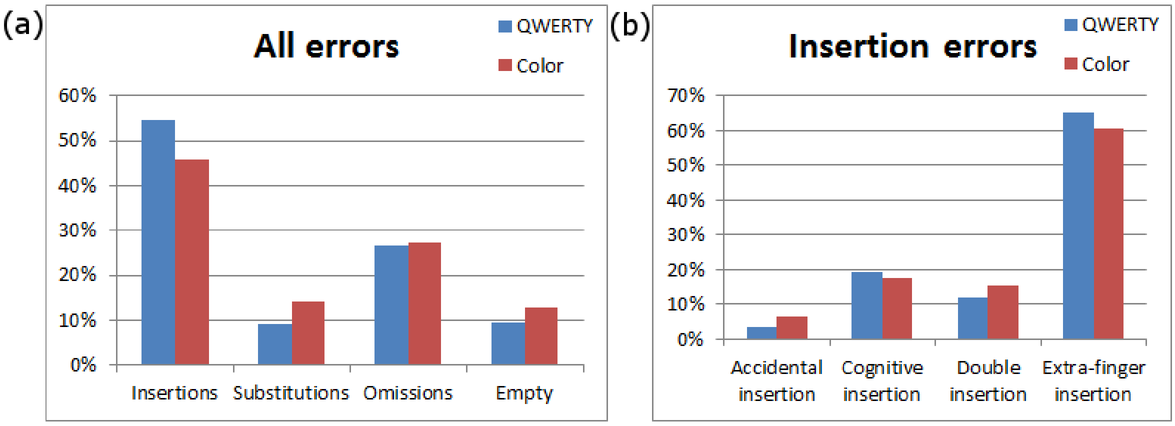


Figure 6.3: (a)The contribution of each type of error to the total amount in each condition; (b) The contribution of each type of insertion error to the total amount in each condition.

Substitution errors were the least common errors committed by older adults, accounting for 9.25% of the total errors. This type of error was more evenly distributed when compared with *insertion* errors. Still, some participants (#4, #9, #10, #14, #16, #17 and #18) did not commit *substitution* errors at all. No correlation was found between *substitution* errors and task-specific tremor for both hands.

All participants committed at least one omission error except users #13 and #19. This type of error was the second most common among the older adults, accounting for 26.76% of the total errors. No correlation was found between omission errors and task-specific tremor for both hands.

Finally, the remaining type of error (Empty) accounted for 9.49% of the total errors. This type of error is surprisingly high, indicating that users sometimes press outside the keyboard unintentionally. As said previously, this type of error is more of a potential error, since it does not produce any output. We do not know for sure if users would still commit this kind of error if those empty areas were filled with buttons. But, in a real situation, if they pressed those same areas, worse errors could occur. No correlation was found between empty errors and task-specific tremor for both hands.

Color variant. Results show that *insertion* errors were also the most common errors committed by older adults on the *Color variant*, accounting for 45.73% of the total errors. This kind of error is unevenly distributed through all the participants. For instance, participants #2, #17 and #20 are responsible for 55.44% of the *insertion* errors. No correlation was found between *insertion* errors and task-specific tremor for both hands.

Substitution errors were the third most common errors committed by older adults, accounting for 13.98% of the total errors. Almost half of the participants (#6, #10, #11, #12, #13, #14, #15, #16 and #17) did not commit *substitution* errors at all. No correlation was found between *substitution* errors and task-specific tremor for both hands.

All participants committed at least one *omission* error except users #4 and #10. This type of error was the second most common among older adults, accounting for 27.49% of the total errors. No correlation was found between omission errors and task-specific tremor for both hands.

Finally, the remaining type of error (Empty) accounted for 12.80% of the total errors. Just like on *QWERTY* keyboard, this type of error is surprisingly high, indicating that users sometimes press outside the keyboard unintentionally. No correlation was found between empty errors and task-specific tremor for both hands.

Predict Words variant. This variant was not analyzed thoroughly regarding typing errors, because most of the participants used it as a *QWERTY* keyboard. Therefore, we opted not to report this data.

Overall Results. We performed Wilcoxon tests between each type of error and we only found significant differences between *cognitive substitution* errors of the *QWERTY* keyboard and *Color* variant ($Z = -1.845,\ p = 0.065$). This suggests that the increase in *cognitive substitution* errors is not a coincidence. This issue will be further discussed in Section 6.2.6.

6.2.5 Insertion Errors

As stated in the previous subsection, we considered four types of *insertion* errors: *accidental*, *cognitive*, *double*, and *extra-finger* insertions.

QWERTY keyboard. In Figure 6.3 (b) it is depicted the contribution of each type of *insertion* error for the total amount of *insertion* errors. We can clearly see that the *extra-finger insertion* error is the most common type of *insertion* error. No correlation was found between *extra-finger insertions* and QWERTY experience, and between *extra-finger insertion* errors and task-specific tremor for both hands. Since the *extra-finger insertion* error exists because the keyboard is multi-touch (see *extra-finger* definition 6.2.4), it is also relevant to assess if this kind of error is mostly committed by participants who used both hands to interact with the keyboard. But, no correlation was found between number of hands used and *extra-finger insertions*. In fact, from the total 5 participants that used both hands to interact, only one participant (#5) committed *extra-finger insertion* errors. This participant only accounted for 4.79% of *extra-finger insertion* errors. Contrary to this, participants #2 and #17, who only used one hand to interact with the device, accounted for 82.19% of the *extra-finger insertion* errors.

Participant #2 interacted with the index finger of her left hand (intentionally), and with the thumb of the same hand (unintentionally). This means that, every time she would tap a key with her index finger, she would also tap the space bar key, or a key in the Z's row, unintentionally. This is the main reason why this user performed so many *extra-finger insertions*. Participant #17 was a completely different case; in fact, as strange as it may sound, this participant only used her index finger to interact with the device, and still committed *extra-finger insertions*. During the test, we thought she was committing *double insertion* errors, since she was only interacting with one finger, and the output would always double the keys she pressed; if she pressed the ”a” key, a double ”a” would appear in the output. When we analyzed the logs, after the test, we realized that something completely different had happened. While the actual cause is still unclear, the logs showed that at least two contact areas were recognized, since touches were interleaved; meaning that, a second press was recognized, before releasing the first press. The only way to correct this kind of error is to disable interleaved touches, i.e., transforming the multi-touch keyboard into a single touch keyboard. This will be further discussed in Section 6.3(Simulation Study).

Cognitive insertion errors were the second most common type of *insertion* errors, accounting for 19.20% of the total *insertion* errors. No correlation was found between *cognitive insertions* and QWERTY experience, and between *cognitive insertions* and task-specific tremor for both hands. When further analyzing this type of error, we realized that it had several causes. For instance, participant #17 sometimes repeated syllables. It seems as if the participant had forgotten that she had already written that specific syllable. Sometimes she would even repeat the same syllable twice in a row. Other participants, like participant #2, seemed to misread the word and then obviously typed a wrong word or character (e.g.: the user typed ”assembleias” when she was supposed to type ”assembleia”). Some participants inserted more than one space between words. These were not double insertions, since the time elapsed between interactions was higher than one second. It seemed as if participants inserted a space character after typing a word, read the next word from the sentence to transcribe, and re-inserted another space, probably because they were not sure if they had already done it. Because of this, the space bar was responsible for 39.53% of the *cognitive insertion* errors. No other patterns were found regarding keys or rows of the keyboard, associated with this kind of error.

The third most common *insertion* error is the *double insertion* error, accounting for 12.05% of *insertion* errors. All participants committed at least one *double insertion* error, except participants #1, #9, #12 and #19. No correlation was found between *double insertions* and QWERTY experience, and between *double insertions* and task-specific tremor for both hands. When analyzing the logs we realized that, even though we always classified the second inserted character as being the *double insertion* error (”aa”), the error itself can be either of the two inserted characters. The user can (1) insert the first character and accidentally insert a second one; (2) before inserting the intentional character, accidentally touch the keyboard and insert a character unintentionally before inserting the desired one. This classification, although similar to the *accidental insertion* error, is different because the *accidental insertion* error is an isolated error, with only one interaction. This will be further discussed in the Intra-key corrected errors subsection (6.4.3).

The least common *insertion* error is the *accidental insertion* error, accounting for 3.57% of *insertion* errors. Only five participants committed this type of error (#1, #2, #3, #16 and #20). No correlation was found between accidental insertions and QWERTY experience, and between *accidental inser-*

tions and task-specific tremor for both hands.

Color variant. Although participants committed slightly more errors on the *Color* variant in general, the contribution of each type of error is very similar. In fact, when looking at Figure 6.3 (b) we can verify that the *extra-finger insertion* error is still the most common type of *insertion* error, even though it was less common on the *Color* variant. As on the *QWERTY* keyboard, only participant #5, from the participants who interacted with both hands, committed *extra-finger insertions* (6.84% of total *extra-finger insertions*). Participants #2 and #17, which used only one hand, are responsible for more than half (67.52%) of the *extra-finger insertions* (similar to the *QWERTY* keyboard).

Cognitive insertion errors were the second most common type of *insertion* errors, accounting for 17.61% of the total *insertion* errors. Although participant #17 committed less *cognitive insertion* errors when compared with the *QWERTY* keyboard, it is probably keyboard unrelated, since this participant committed *extra-finger insertion* errors in all the sentences; this means that most of the time, the *prediction system* was unable to suggest letters, behaving mostly like the *QWERTY* keyboard.

The third most common *insertion* error is the *double insertion* error, accounting for 15.54% of *insertion* errors, followed by *accidental insertions* (6.22%).

Overall Results. The contribution of each type of *insertion* error is similar across variants. This was expected since the *Color* variant does not aim to correct *insertion* errors in general. It could somehow avoid *cognitive insertions* since it draws the attention of the user to the four most probable keys. Still, no significant differences were found between variants.

6.2.6 Substitution Errors

As stated previously in subsection 6.2.4, we considered two types of *substitution* errors: *neighbor* and *cognitive substitution* errors.

QWERTY keyboard. In Figure 6.4 it is depicted the contribution of each type of *substitution* error to the total amount of *substitution* errors. *Neighbor substitutions* account for 76.32% of the total *substitution* errors. No correlation was found between *neighbor substitutions* and QWERTY experience, *neighbor substitutions* and number of hands used and *neighbor substitutions* and task-specific tremor for both hands.

Regarding *neighbor substitution* errors, some of the most frequent were: $D \to X$ (20.69% of *neighbor substitution* errors), $S \to Z$ (20.69%), $c \to SPACE$ (17.24%), $A \to Z$ (6.90%) and $T \to G$ (6.90%). As we can see, there is a clear predominance of bottom key substitutions in the data, which suggests that participants found it easier to hit keys in the bottom (south) direction. Although there were no right key substitutions in the data, we can see in Figure 6.5 (a), that touch points are not only skewed to the bottom but also slightly to the right. Nicolau [14] also reported this result, and hypothesized that these findings could be related to hand dominance. Since in our study participants were free to interact the way they preferred, we filtered the results by participants that used only their *right hand*, only their *left hand* and *both*. Looking at Figure 6.5, we confirm that Nicolaus's [14] hypothesis was right; that is, ***touch points are generally skewed towards the bottom and to the side of the hand the participant is using to type.***

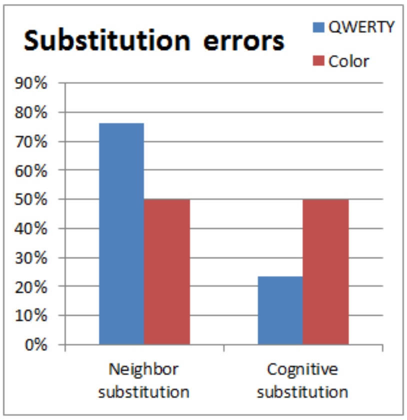


Figure 6.4: The contribution of each type of substitution error to the total amount in each condition.

When looking at Figure 6.5 (a), the one with the *right hand* pattern, we can verify that the only key that does not follow the pattern is the key ”Q”. The left hand pattern has mixed results (Figure 6.5 (b)). Still, we have to take into account that our data might not be enough, since we only had two participants (#2 and #10) typing only with their *left hand*. Although the left pattern was not always verified, the bottom pattern was verified for all keys, except for key ”G”. Regarding the pattern that emerged from the participants that used *both hands* (Figure 6.5 (c)), we can verify that the left side of the keyboard (Q, E, R, A, S, D, F, G, C and V), has its touch points skewed towards the bottom-left while the right side of the keyboard (U, I, O, P, L, N and M) has its touch points skewed towards the bottom-right. The only key that does not follow this pattern is the key ”T”. Instead of its touch points being skewed to the bottom-left, they are skewed to the bottom-right. The remaining keys (W, Z, X, Y, H, B, J and K) do not belong to either pattern because they were never used during the test. Still, we hypothesize that W, Z and $X's$ touch points would be skewed towards the bottom-left, while the remaining would be skewed towards the bottom-right, in the case of participants using both hands.

Figure 6.6 (a) shows the new center of the keys for the participants that only used their right hand. Visually inspecting the Figure, we can verify that shifts are more intense on the vertical direction (y axis) than on the horizontal direction (x axis). Also the vertical shift seems to increase gradually, from row to row, until we reach the space bar’s row. Something similar seems to happen with the horizontal shift; that is, the horizontal shift seems to increase gradually as we move from the left to the right side of the keyboard. Indeed, when looking at the Table A.6 (b), we can verify that the ***vertical shift increases gradually from row to row.*** Regarding the horizontal shift, the pattern is not completely verified, but it is clear that ***columns on the right side of the keyboard have a more accentuated shift when compared with columns on the left side of the keyboard*** (Table A.6 (a)).

Regarding participants that only used their *left hand* (Figure 6.6 (b)), we can also verify that shifts are more intense on the vertical direction (y axis) than on the horizontal direction (x axis). However, the ”gradually increasing vertical shift” pattern is harder to verify (visually) since some keys (Q, G, N and M) seem to contradict the pattern. But, this might be due to the lack of data regarding participants that typed only with their left hand. Regarding the horizontal direction, we were expecting a symmetrical result when compared with participants that only used their right hand. But, as reported previously, not all keys followed the bottom-left pattern, so we cannot verify it. We think it might be due to lack of

Figure 6.5: Touch points of participants who used only their (a) right, (b) left and (c) both hands to interact in the QWERTY condition.

data, since only two participants (#2 and #10) typed only with their left hand.

Regarding participants that used both hands (Figure 6.6 (c)), we can also verify that shifts are more intense on the vertical direction (y axis) than on the horizontal direction (x axis). However, it is not completely clear if the vertical shift increases gradually, from row to row. Concerning the horizontal direction, there seems to be an increasing shift to the right, from the middle of the keyboard to the right edge of the keyboard, and an increasing shift to the left, from the middle of the keyboard to the left edge of the keyboard.

The other type of *substitution* errors, *cognitive substitution* errors, accounted for 23.68% of the total substitution errors. No correlation was found between *cognitive substitutions* and QWERTY experience, *cognitive substitutions* and number of hands used, and *cognitive substitutions* and task-specific tremor for both hands. Contrary to *neighbor substitutions*, no pattern was found for *cognitive substitution* errors; all *cognitive substitutions* were isolated cases (e.g.: $E \rightarrow I$, $T \rightarrow L$, $A \rightarrow O$). Nicolau [14] described a substitution pattern on which participants mistake symmetrical or ”similar” characters such as $p \rightarrow q$, $b \rightarrow d$, $i \rightarrow l$, $i \rightarrow j$. During the evaluation phase we did not find this kind of substitution pattern, but that is probably because during the training phase we informed them that, for instance, when they used the letter ”q” instead of ”p” that they probably wanted to use ”p” instead of ”q”. None of them repeated this substitution pattern during the *QWERTY* keyboard test.

Color variant. Regarding *neighbor substitution* errors, some of the most frequent were: $C \rightarrow SPACE$ (30% of the *neighbor substitution* errors), $S \rightarrow Z$ (23.33%), $D \rightarrow X$ (16.67%), $N \rightarrow SPACE$ (6.67%) and $M \rightarrow SPACE$ (6.67%). Just as in *QWERTY* keyboard, there is a clear predominance

Figure 6.6: New center of keys for participants who only used their (a) right, (b) left and (c) both hands to interact in the QWERTY condition.

of bottom key substitutions in the data, which suggests that participants found it easier to hit keys in the bottom (south) direction. Looking at Figure 6.7, we can also verify that touch points are generally skewed towards the bottom and to the side of the hand the participant is using to type. When looking at Figure 6.7 (a), the one with the *right hand* pattern, we can verify that the only key that does not follow the pattern is the key ”Q”. The *left hand* pattern has mixed results (Figure 6.7 (b)). Still, we have to take into account that our data might not be enough, since we only had two participants (#2 and #10) typing only with their left hand. Although the left pattern was not always verified, the bottom pattern was verified for all keys.

Regarding the pattern that emerged from the participants that used *both hands* (6.7 (c)), we can verify that the right side of the keyboard (U, I, O, P, L, B, N and M) has its touch points skewed towards the bottom-right while almost all the keys from the left side of the keyboard (Q, E, A, S, D, F, G and C), has its touch points skewed towards the bottom-left. Only ”R”, ”T” and ”V” keys do not follow the pattern. Instead of the touch points being skewed to the bottom-left, they are skewed to the bottom-right. The remaining keys (W, Z, X, Y, H, J and K) do not belong to either pattern because they were never used during the test. Still, we hypothesize that W, Z and $X's$ touch points would be skewed towards the bottom-left, while the remaining would be skewed towards the bottom-right, in the case of participants using both hands.

Just like on the *QWERTY* keyboard, on the *Color* variant, shifts are more intense on the vertical direction (y axis) than on the horizontal direction (x axis). Also the vertical shift increases gradually, from row to row, until we reach the space bar's row (Figure 6.8 and Table A.7 (b)). For users who only used their *right hand*, the gradual shift is not completely verified, but it is clear that shifts are more

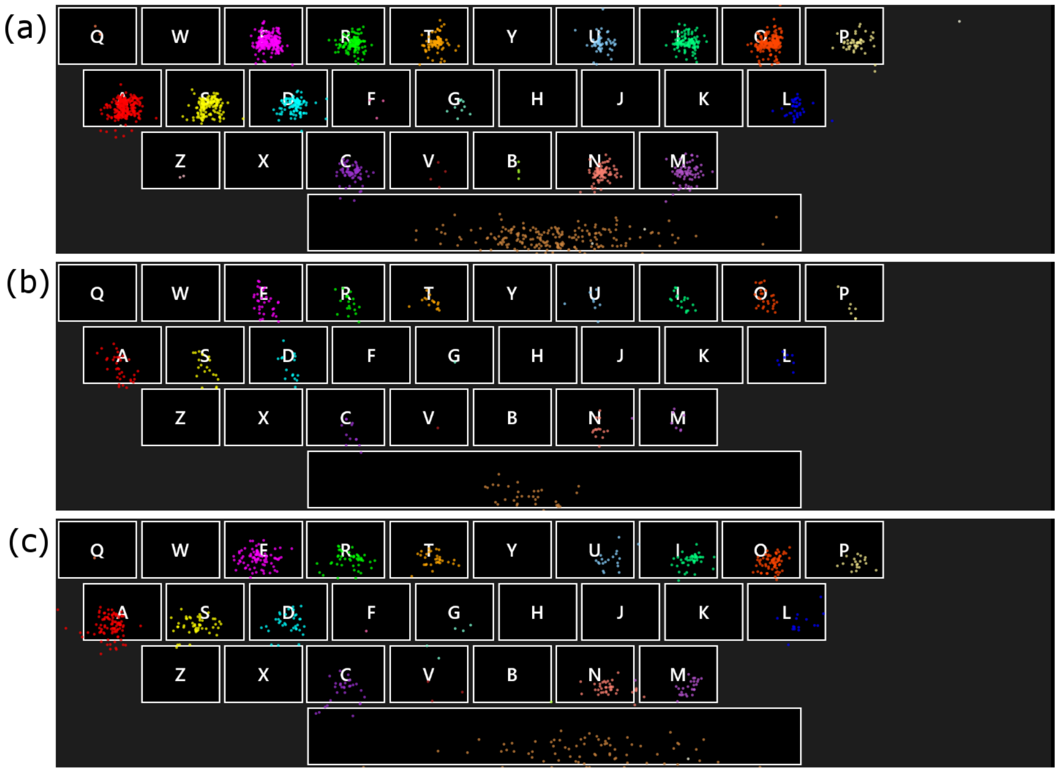


Figure 6.7: Touch points of participants who used only their (a) right, (b) left and (c) both hands to interact in the Color condition.

intense on the right side of the keyboard, than on the left side (Figure 6.8 (a) and Table A.7 (a)). For users who only used their *left hand* we cannot verify the expected opposite shift on the x-axis, since not all keys followed the bottom-left pattern (Figure 6.8 (b)). For users who used *both hands*, there seems to be an increasing shift to the right, from the middle of the keyboard to the right edge of the keyboard, and an increasing shift to the left, from the middle of the keyboard to the left edge of the keyboard (Figure 6.8 (c)).

The percentage of *cognitive substitution* errors doubled in comparison with *QWERTY's* keyboard. When analyzing in detail, we verified that in 65.52% of *cognitive substitutions* the user inserted a character that was highlighted by the *Color* variant (Table A.4 (b)). Also, in 20.69% of *cognitive substitution* errors, the expected key and the one the user inserted were both highlighted. And, in 34.48% of those errors the expected key was highlighted, which did not prevent the user from inserting another character. Although we cannot tell for sure what this means, and since there is a significant difference between the cognitive substitution errors of the *QWERTY* keyboard and *Color* variant $(Z = -1.845,\ p = 0.065)$, we think this is a good hint that the increase of *cognitive substitutions* in the *Color* variant might not be a coincidence. Although we are confident that participant #20 was influenced by the *Color* variant to type ”cooperacao” instead of ”cooperantes” (section 6.2.3), we do not know for sure if the same happened to participant #1, who typed ”internacional” when he was expected to type ”intermediario”. Although, the keys that lead to ”internacional” were highlighted, there is always the possibility that the user simply misread the word. Other cause for this kind of error seems to be the already referred substitution of symmetrical/similar characters. In two different cases, participants confused the ”q” and ”p” characters and the ”l” and ”i” characters, although the correct characters were highlighted (”p” and ”i”) and the wrong characters were not (”q” and ”l”).

Figure 6.8: New center of keys for participants who only used their (a) right, (b) left and (c) both hands to interact in the Color condition.

Overall Results. Although the keys involved in the *neighbor substitution* errors were not exactly the same across variants, the most frequent were, in general, the same. It was also verified a clear predominance of bottom key substitutions in the data, which suggests that participants found it easier to hit keys in the bottom (south) direction. We also verified that touch points are generally skewed towards the bottom and to the side of the hand the participant is using to type. We also found that shifts are more intense on the vertical direction (y axis) than on the horizontal direction (x axis). The vertical shift increases gradually, from row to row, until we reach the space bar's row. The pattern is not completely verified for the horizontal shift. On both QWERTY and Color keyboards the pattern was not verified for columns 4 (T, G and V), 5 (Y, H and B), 6 (U, J and N) and 8 (O and L). Still, it is clear that columns on the right side of the keyboard have a more accentuated shift when compared with the columns on the left side of the keyboard.

Regarding *cognitive substitution* errors, we found common patterns during the training phase of the *QWERTY* keyboard and the evaluation phase of the *Color* variant. Some participants mistake symmetrical or "similar" characters such as $p \to q$, $b \to d$, $i \to l$, $i \to j$. We also found a significant difference between the *cognitive substitution* errors of the *QWERTY* keyboard and *Color* variant $(Z = -1.845,\ p = 0.065)$. Although we cannot tell for sure, we hypothesize that the *Color* variant lead participants to perform more *cognitive substitution* errors.

6.2.7 Omission Errors

As stated previously in subsection 6.2.4, we considered three types of *omission* errors: *cognitive*, *slide* and *failed omission* errors.

QWERTY keyboard. In Figure 6.9 it is depicted the contribution of each type of *omission* error to the total amount of *omission* errors. *Cognitive omissions* account for 52.72% of the total *omission* errors. *Cognitive* errors are due to the user having an improper model of an intended word/sentence [14]. Sometimes users may forget to insert a specific character or may not be able to spell an intended word correctly.

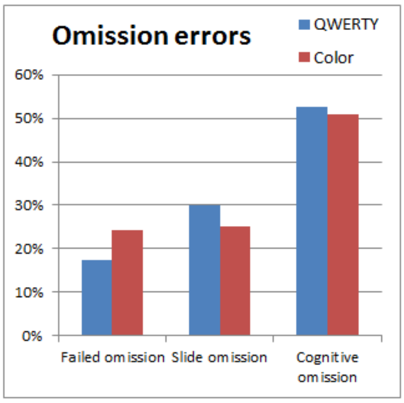


Figure 6.9: The contribution of each type of omission error to the total amount in each condition.

Just like Nicolau [14], we also found that forgetting to enter a blank space between words was a common issue among older adults. During the training phase we clarified the concept of blank space, which all participants understood. Therefore, we hypothesize that this kind of omission is due to lack of practice. This issue was so common that, 44.83% of the total *cognitive omissions* were space omissions. Some participants also forgot complete words: participant #6 forgot to type ”aos”, participant #13 forgot to type ”pertencia” and participant #16 forgot to type ”por”. Other participants forgot to type random letters on several words.

Slide omission errors were the second most common type of *omission* errors, accounting for 30% of the total *omission* errors. This kind of error differs from the previous one, because on this one the user presents the intention to type a character, but fails in the execution. It means that the user presses and lifts his finger on different keys; therefore no output is generated, resulting in the omission of a character. When analyzing in detail this kind of error, we discovered that some users pressed the space bar, and before releasing it, they slid towards the bottom edge of the tablet device, lifting their finger at pixel 340 (y axis) exactly. Due to a problem in our implementation this pixel was not associated with the space bar; therefore, when lifting the finger at pixel 340 (y axis), the interaction was assigned to an empty space, wrongly. This error accounted for 27.27% of *slide omission* errors. Furthermore, we classified *slide omissions* in three subclasses: *correct land-on*, characterized by the finger landing on the intended key, and then sliding to another key; *incorrect land-on*, characterized by the finger landing on a neighbor key, and then sliding to the intended key; and *accidental slide*, on which the user has no intention to tap either of the keys. The first type accounted for 36.36% of the *slide omission* errors, the second 57.58% and the third 6.06%. We found that **all the errors** classified as *correct land-on*, ended always in a key below the intended one; that is, the slide was always performed from the top to the bottom. Contrary to this, 89.47% of the errors classified as *incorrect land-on*, ended in a key above the pressed one. On the remaining cases the slide was performed from the right to the left. This means that, ***when taking into account only the vertical slides, if a user performs a slide starting at a key in a given row, and lifts his finger on a key in the row above, we are 100% sure that the user intended to tap the key in the row above.*** When the opposite happens, ***in 85.71% of the times, the user also wants the key in the row above (the key were he would have landed his finger)***. In the remaining 14.29% times, we do not know exactly what were the intentions of the user, since the slide was completely accidental.

Another kind of *omission* error is the *failed omission* error that accounts for 17.27% of the total *omission* errors. This kind of error exists because we deleted all keys that would not be needed during the test, thus creating empty spaces at the borders of the keyboard (Figure 6.10). This kind of error differs from the *empty* error, because an interaction is only considered a *failed omission* if the touch point is on an empty space, near the intended/expected key; when this happens we can be sure that the intention of the user was to tap that key. This kind of error was common on the left side of the keyboard, near the "A" key, accounting for 84.21% of the *failed omissions*. It also happened twice on the right side of the keyboard, near the "L" key, and once on the left side of the space bar. This kind of error is important to report, because in the final version of the keyboard, these empty spaces will be filled with buttons.

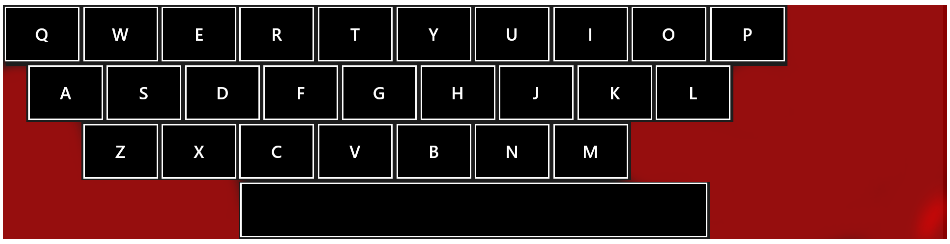


Figure 6.10: The red area illustrates the empty space, on which "empty" errors can occur.

Color variant. In Figure 6.9 it is depicted the contribution of each type of *omission* error to the total amount of *omission* errors. *Cognitive omissions* account for 50.86% of the total *omission* errors. Since the *Color* variant highlights the next most probable keys, it would be expected that, when correct, the suggestion could minimize omissions. Still, *cognitive omissions* were as frequent as on the *QWERTY* keyboard. When further analyzing this type of error, we found that during 52.54% of the *cognitive omissions*, the expected key was highlighted (Table A.5 (a)). Which means that in more than half of the cases, the *Color* variant was helping the participant, and still they forgot to type the intended character. 16.95% (Table A.5 (b)) of the remaining cases the user forgot to type complete words (probably misread the sentence), 20.34% the *prediction system* did not highlight the expected word because the user had already made an error and the remaining 10.17% the key was not highlighted because the word was not in the dictionary.

Slide omission errors were the second most common type of *omission* errors, accounting for 25% of the total *omission* errors. *Correct land-on* accounted for 37.93% of the *slide omission* errors, while *incorrect land-on* and *accidental slides* accounted for 58.62% and 3.45%, respectively. 81.82% of the *correct land-on* errors ended in a key below the intended one; that is, the slide was performed from the top to the bottom. On the remaining cases, the slide was performed from the left to the right.

Regarding *incorrect land-on* errors, 94.12% ended in a key above the pressed one. On the remaining cases, the slide was performed from the left to the right. This means that, ***when taking into account only the vertical slides, if a user performs a slide starting at a key in a given row, and lifts his finger on a key in the row above, we are 100% sure that the user intended to tap the key in the row above***. When the opposite happens, ***we are also 100% sure the user intended to tap the key in the row above (the key were he lifted his finger)***.

Failed omission errors accounted for 24.14% of the total omission errors. This kind of error was common on the left side of the keyboard, near the "A" key, accounting for 89.29% of the *failed omission* errors. It also happened twice on the right side of the keyboard, near the "L" key (7.14%), and once on the right side of the "P" key (3.57%).

Overall Results. *Cognitive omission* errors had approximately the same share across variants. Still, we found that in 52.54% of the *cognitive omission* errors in the *Color* variant, the expected key was highlighted. This means that in more than half of the cases, the *Color* variant was helping users, and still they forgot to type the intended character. Regarding *slide omission* errors, we found a pattern across variants that, to our knowledge, has not been reported by any other author. We found that, ***when taking into account only the vertical slides, if a user performs a slide starting at a key in a given row, and lifts his finger on a key in the row above, his intention was to tap the key in the row above***. When the opposite happens, ***his intention was to also tap the key in the row above (the key were he would have landed his finger)***. We hypothesize that ***when the user slides down, he is moving his hand towards the rest position, below the tablet. When the movement is in the opposite direction, it is a corrective movement, because the user adjusted the touch position in a contrary motion to the resting position.*** To our knowledge, this pattern has not been reported by any other study, presenting an opportunity for improvement of virtual keyboards. When the user performs a completely accidental slide, there is no intention, therefore the user actually did not want to tap any of the keys. *Failed omission* errors occurred mainly near the "A" key in both variants. It also occurred near the "L" key (both variants), "P" key (*Color* variant) and space bar (*QWERTY* keyboard).

6.2.8 Participants' Preferences, Comments and Observations

At the end of the user study participants were debriefed and asked about their preferred keyboard. We also collected comments during and after the test about their opinion regarding all the keyboards.

When asked about their satisfaction (5-point Likert scale) using each of the keyboard variants, participants gave a higher rate to the *QWERTY* keyboard ($Mean = 3.80;\ Median = 4.00$). The *QWERTY* keyboard was followed closely by the *Color* variant ($Mean = 3.75;\ Median = 4.00$) and finally by the *Predict Words* variant ($Mean = 3.10;\ Median = 3.00$). Still, 6 users rated the *Color* variant with the highest score (5), while only 1 user rated each of the remaining keyboards with the highest score.

We had more "complex" questions regarding easiness of use, utility, cognitive effort and easiness to locate the intended key. But, during the study we found that participants' did not understand the difference between them; they reported the questions were all the same. Therefore, we decided to keep things simple and ask only about their satisfaction towards the keyboard.

A participant reported that it was faster to type with the *Color* variant, referring to a specific case when the system was able to suggest always the right letter. Some users told us that the *Color* variant was really helpful, but in order to take advantage of it, it was necessary to pay attention. Some users also reported that the tablet was too sensitive; referring to the fact that it is easy to make typing mistakes.

6.3 Simulation Study

In this simulation study we evaluate the performance of the *Shifted*, *Size Invisible*, *Single Touch* and *Intra-key*, *Inter-key* and *Combined Timed* variants (described in section 4.3). In order to do this we feed the simulation with the log data obtained from the *QWERTY* keyboard condition. The log data contains information about the sentence to transcribe, the pressed and released point of each interaction and the time elapsed between each interaction.

As stated in Chapter 4, we decided to use these variants as simulations, because it would be unrealistic to ask each older participant to perform tests with ten different variants. That would require at least three different sessions with each participant. Although a simulation is not as good as running real user tests, it can give us some insights on the impact these variants would have if users were typing on them. It is important to note that simulations were only performed with variants that were visually similar to the QWERTY keyboard; the differences were only regarding the processing of the touch inputs. Therefore, we believe these results are as valid as the results from the user study.

6.3.1 Research Questions

1. Will the *Shifted* variant be able to reduce significantly neighbor substitution, failed omission and slide omission errors? Will a generic approach be enough?
2. Will the *Size Invisible* variant be able to reduce significantly neighbor substitution, failed omission and slide omission errors? Will a generic approach be enough? How does it compare with the *Shifted* variant?
3. Will the *Single Touch* variant be able to reduce significantly *extra-finger insertion* errors? Will this work for users who interact with both hands?
4. Will the *Intra-key Timed* variant be able to reduce significantly *accidental insertion* errors? Will a generic approach be enough?
5. Will the *Inter-key Timed* variant be able to reduce significantly *double insertion* errors? Will a generic approach be enough?
6. Will the combination of the *Intra-key* and *Inter-key Timed* variants (*Combined Timed* variant) be able to reduce significantly accidental and double insertion errors? Will a generic approach be enough?

6.4 Simulation Results

In this section we analyze accuracy for the six simulated conditions (*Shifted*, *Size Invisible*, *Single Touch* and *Intra-key*, *Inter-key* and *Combined Timed* variants), focusing on the corrected errors.

6.4.1 Quality of Transcribed Sentences

The quality of the transcribed sentences was measured using the *Minimum String Distance* (*MSD*) error rate as calculated in section 6.2.3:
$MSD(required\,text, transcribed\,text) \div Max(\mid required\,text \mid, \mid transcribed\,text \mid) \times 100$

Shifted Simulation. The *Shifted* simulation was able to reduce the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 8.71% $(SD = 10.25\%)$. Participants #2, #5, #7, #8, #11-#15 and #20 benefited with this simulation. Participant #17 had the quality of her sentences slightly reduced with this simulation.

Size Invisible Simulation. The *Size Invisible* simulation was able to reduce slightly the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 9.52% $(SD = 11.19\%)$. We hypothesize that the poor results are related with the fact that older adults committed more errors than younger users. Since the *Size Invisible* variant depends highly on the prediction algorithm, a slight error (*insertion*, *substitution* or *omission*) is enough to make the *prediction system* unable to operate, and therefore behaving as a normal *QWERTY* keyboard. Participants #2, #7, #8, #11, #12, #15 and #20 benefited with this simulation. Participants #5 and #17 had the quality of their sentences slightly reduced with this simulation.

Single Touch Simulation. The *Single Touch* simulation was able to reduce the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 7.68% $(SD = 6.42\%)$. Even though this simulation had better results when compared with the *Shifted* and *Size Invisible* simulations, the results are spread across participants; there is the same number of bad cases (#8-#10, #12, #14 and #18) and good cases (#2, #3, #5, #11, #17 and #20). What happened is that participants #2 and #17 benefited highly with this simulation. The simulation was able to increase the quality of their transcribed sentences in 43.51% and 62.25%, respectively.

Intra-key Timed Simulation. The *Intra-key* simulation was able to reduce the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 7.44% $(SD = 5.85\%)$. Since this variant is built upon *Single Touch*, it is also important to compare it with *Single Touch*. When compared with it, the *Intra-key* simulation was able to reduce the average *MSD* error rate from 7.68% $(SD = 6.42\%)$ to 7.44% $(SD = 5.85\%)$. Participants #2, #3, #5, #6, #11, #12, #16, #17 and #20 benefited with this simulation (when compared with the *QWERTY* keyboard). Participants #1, #7-#10, #13-#15 and #18 had the quality of their sentences slightly reduced with this simulation.

Inter-key Timed Simulation. The *Inter-key* simulation was able to reduce the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 6.94% $(SD = 6.28\%)$. Since this variant is built upon *Single Touch*, it is also important to compare it with *Single Touch*. When compared with it, the *Inter-key* simulation was able to reduce the average *MSD* error rate from 7.68% $(SD = 6.42\%)$ to 6.94% $(SD = 6.28\%)$. Participants #2-#6, #11, #13-#17 and #20 benefited with this simulation (when compared with the *QWERTY* keyboard). Participants #1, #8-#10, #12 and #18 had the quality of their sentences slightly reduced with this simulation.

Combined Timed Simulation. The *Combined* simulation was able to reduce the average *MSD* error rate of the *QWERTY* keyboard from 9.76% $(SD = 11.05\%)$ to 7.28% $(SD = 6.19\%)$. Since this variant is built upon *Single Touch*, it is also important to compare it with *Single Touch*. When compared with the it, the *Combined* simulation was able to reduce the average *MSD* error rate from 7.68% $(SD = 6.42\%)$ to 7.28% $(SD = 6.19\%)$. Participants #2-#6, #11, #12, #15-#17 and #20 benefited with this simulation (when compared with the *QWERTY* keyboard). Participants #1, #7-#10, #13, #14 and #18 had the quality of their sentences slightly reduced with this simulation.

Overall Results. A repeated measures ANOVA did not reveal significant differences between keyboard variants on *MSD* error rate $(F(1.370,\ 20.557) = 1.481,\ p = 0.246)$. Still, it is important to report that in order to fulfill ANOVA's assumptions, we had to remove participants #2, #5 and #17, because they were *outliers*. We used the *labeling rule* [63] $(upper\ limit = Q3 + (1.5 \times (Q3 - Q1)))$ to calculate the upper limit (21.72). Since the error rate of those participants was higher than the upper limit, they were considered *outliers*. These participants are the ones who committed more errors on the *QWERTY* keyboard, and therefore are the ones who can benefit more with the simulations. This is possibly the reason why the repeated measures ANOVA did not reveal significant differences. However, a Wilcoxon test between all the variants revealed significant differences between *QWERTY* and *Shifted* $(Z = -2.746,\ p = 0.006)$, *QWERTY* and *Size Invisible* $(Z = -2.073,\ p = 0.038)$, *QWERTY* and *Inter-key* $(Z = -2.373,\ p = 0.018)$, and *Single Touch* and *Inter-key* $(Z = -3.006,\ p = 0.003)$ variants. We also found a difference between *QWERTY* and *Single Touch* $(Z = -1.223, p = 0.221)$, although at a lower significance level (77.9%). We hypothesize that this is because some participants benefited highly, while others did not benefit at all. These results were expected, except the fact that *Intra-key* and *Combined* variants did not have significant differences, regarding the quality of the transcribed sentences when compared with the *QWERTY* keyboard. In the next subsection (6.4.2) we will get into the detail of the errors each variant corrected.

6.4.2 Typing Errors

In this section we present the errors each simulated variant can correct and its new percentages after the simulation.

Shifted Simulation. The *Shifted* variant is mainly concerned in correcting *neighbor substitution* errors. It can also correct *failed* and *slide omissions*. Therefore, when compared with the *traditional QWERTY* keyboard, no *insertion* or *empty* errors were corrected. The *Shifted* simulation was able to correct 44.74% of the *substitution* errors and 29.09% of the *omission* errors. When focusing solely in the subclasses of errors that can be corrected by this variant, it was able to correct 58.62% of the *neighbor substitutions*, 100% of the *failed omissions* and 39.39% of the *slide omissions*.

Size Invisible Simulation. Just like the *Shifted* variant, the *Size Invisible* variant can only correct *neighbor substitution*, *failed omission* and *slide omission* errors. The *Size Invisible* simulation was able to correct 13.16% of the *substitutions* and 7.27% of the *omissions*. When focusing solely in the errors that can be corrected by this variant, it was able to correct 17.24% of the *neighbor substitutions*, 21.05% of the *failed omissions* and 12.12% of the *slide omissions*.

Single Touch Simulation. The *Single Touch* simulation was able to correct 63.84% of the *insertion* errors, 38.46% of the *empty* errors and introduced 26 new errors. The *Single Touch* variant main goal is to deal with *extra-finger insertions*. When taking into account only this type of error, this variant was able to correct 78.77% of them (counting already with the 26 new errors that emerged).

Intra-key Timed Simulation. The *Intra-key* simulation was able to correct 45.68% of *insertions* when compared with the *Single Touch* simulation, and 80.36% when compared with the *QWERTY* keyboard. It also corrected 20.83% and 51.28% of the *empty* errors when compared with the *Single Touch* simulation and *QWERTY* keyboard, respectively. Still, *Intra-key's* main goal is to avoid *accidental insertions*. When taking into account only this type of error, the variant was able to correct 75% of them. However, this variant also created 32 new errors; that is, it omitted correct characters, because the elapsed time between the press and release of the same key was lower than the defined threshold.

Inter-key Timed Simulation. The *Inter-key* simulation was able to correct 38.27% of *insertions* when compared with the *Single Touch* simulation, and 77.68% when compared with the *QWERTY* keyboard. It also corrected 12.5% and 46.15% of the *empty* errors when compared with the *Single Touch* simulation and *QWERTY* keyboard, respectively. Still, *Inter-key's* main goal is to avoid *double insertions*. When taking into account only this type of error, the variant was able to correct 85.19% of them. However, this variant also created 8 new errors; that is, it omitted correct characters, because the elapsed time between the release of a key and the press of a second key was lower than the defined threshold.

Combined Timed Simulation. The *Combined* simulation was able to correct 55.56% of *insertions* when compared with the *Single Touch* simulation, and 83.93% when compared with the *QWERTY* keyboard. It also corrected 29.17% and 56.41% of the *empty* errors when compared with the *Single Touch* simulation and *QWERTY* keyboard, respectively. Still, *Combined's* main goal is the same as Intra and Inter-key's; that is, to avoid *accidental* and *double insertions*. This variant was able to correct 75% of the *accidental insertions* and 85.19% of the *double insertions*. However, this variant also created 35 new errors; that is, it omitted correct characters, because the elapsed time between the press and release of the same key, and the elapsed time between the release of a key and the press of a second key were lower than the defined thresholds.

Overall Results. A Wilcoxon test between *QWERTY's* and *Shifted's neighbor substitution* errors, revealed significant differences $(Z = -2.456;\ p = 0.014)$, as well as between *QWERTY's* and *Size Invisible's* $(Z = -2.236;\ p = 0.025)$; which means that both *Shifted* and *Size Invisible* variants were able to significantly reduce *neighbor substitutions*. The same occurred between these same variants regarding *failed omissions* (*QWERTY* and *Shifted*: $Z = -2.207;\ p = 0.027$. *QWERTY* and *Size Invisible*: $Z = -2.000; p = 0.046$) and *slide omissions* (*QWERTY* and *Shifted*: $Z = -2.232; p = 0.026$. *QWERTY* and *Size Invisible*: $Z = -1.633;\ p = 0.102$). Although, in this last case with a confidence level of 89.8%. A Wilcoxon test between *QWERTY's* and *Single Touch's extra-finger insertion* errors, revealed significant differences $(Z = -2.670;\ p = 0.008)$ when not taking into account the new errors created by this variant. When taking the new errors into account, we found significant differences $(Z = -1.584;\ p = 0.113)$ at a lower confidence level (88.7%). Therefore, *Single Touch* variant was able to significantly reduce *extra-finger insertions*. A Wilcoxon test between *QWERTY's* and *Intra-key's accidental insertion* errors, revealed significant differences $(Z = -1.633;\ p = 0.102)$ at a lower confidence level (89.8%), when not taking into account the new errors created by this variant. When taking the new errors into account, we also found significant differences $(Z = -3.478;\ p = 0.001)$. On the first case it means that *Intra-key* variant was able to significantly reduce *accidental insertion* errors, while on the second one, it means that *Intra-key* actually created more new errors than corrected *accidental insertions*. A Wilcoxon test between *QWERTY's* and *Inter-key's double insertion* errors, revealed significant differences $(Z = -3.372;\ p = 0.001)$, when not taking into account the new errors created by this variant. When taking the new errors into account, we did not find significant differences. On the first case it means that *Inter-key* variant was able to significantly reduce *double insertions*. A Wilcoxon test between *QWERTY's* and *Combined's accidental* and double insertion errors (the two types of error that could be solved by *Combined* variant), revealed significant differences $(Z = -3.342;\ p = 0.001)$, when not taking into account the new errors created by this variant. When taking the new errors into account we also found significant differences $(Z = -1.947;\ p = 0.052)$ at a lower confidence level (94.8%). On the first case it means that *Combined* variant was able to significantly reduce *accidental* and *double insertions*, while on the second one, it means that *Combined* variant actually created more new errors than corrected *accidental* and *double insertions*.

6.4.3 Corrected Errors

In this section we thoroughly analyze the corrected errors of each simulation.

Shifted Simulation. Since participants mainly used their right hand to interact, and we were aiming at a solution that would fit all users, we decided to shift the touch points towards the top and the left, in order to compensate the global bottom-right skewness verified on the QWERTY data. In order to find the optimal shift that fitted all users, we performed several simulations. These simulations combined shifting the touch points to the top between 0% and 15% of the height of the key (increments of 1%), and to the left between 0% and 15% of the width of the key (increments of 1%). The best result was obtained with a horizontal shift of 2% (or 4-6%) and a vertical shift of 10%. Figure 6.11(a) shows a *Heat map* with the results for all the tested shifts.

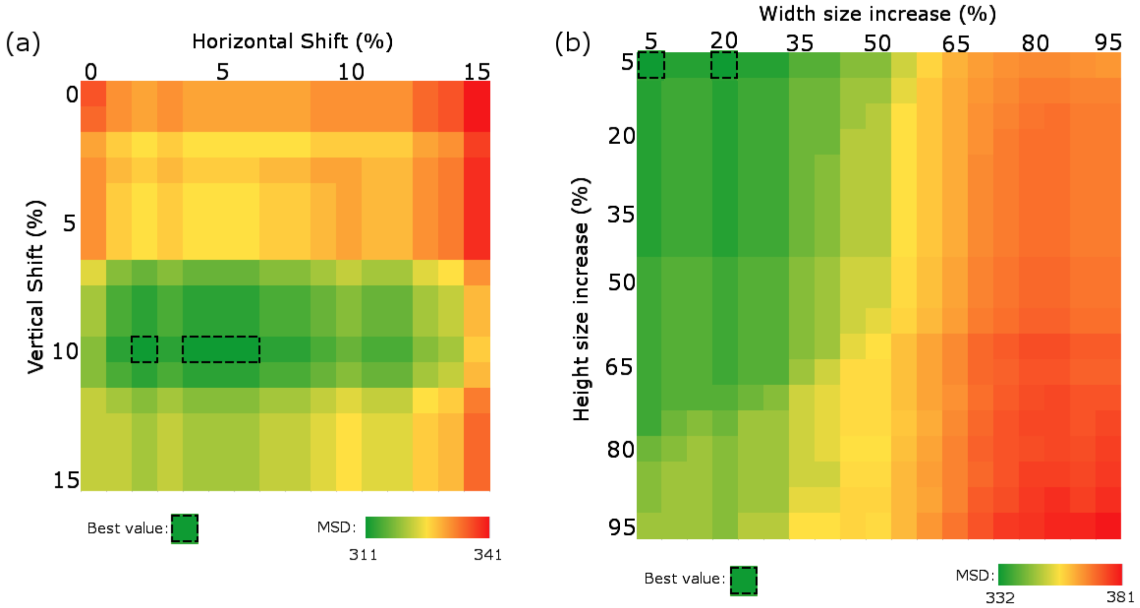


Figure 6.11: Heatmap for the (a) Shifted and (b) Size Invisible variants.

This result proves that shifts are indeed more intense on the vertical direction (y axis) than on the horizontal direction (x axis), because the vertical compensation is higher than the horizontal one. Just because this is the generic shift that suits all the participants, it does not mean that it is the best shift for each participant. For instance, the best shift for participant #2 is between 12% and 15% horizontally and between 8% and 10% vertically. Participant #17 had better results with no shift, and participants #1, #3, #4, #6, #9, #10, #14, #16, #18 and #19 did not benefit with the *Shifted variant*. We also have to take into account that this variant will only benefit users that have made *neighbor substitutions*, *failed omissions* and/or *slide omissions*, which is why users #1, #4, #9, #14, #16, #18 and #19 (from the list above) could have never benefited from this variant, since they did not commit any of those errors.

Participants #3, #6 and #10 (also from the list above) and #11 and #20 did commit *slide omissions* that were not solved. Participant #3 committed a *slide* from ”S” to ”Z” (”S” being the intended key) that is impossible to solve without creating new errors; in order to solve this error, the touch points would need to be shifted 23% instead of 10%, vertically. The remaining *slide omission* errors from the other participants are related with *slides* that start at a certain position of the space bar and end at the bottom of it (pixel 340 y axis). As we already reported in subsection 6.2.7, this was due to a problem in our implementation.

All the *neighbor substitution* errors committed by participants #8, #11, #13 and #15 were corrected by the *Shifted* variant. One of the *neighbor substitutions* committed by participant #7 was not corrected. When analyzing this error we verified that it could only be corrected if the shift was performed in the opposite direction. This happened because this participant used both hands to interact with the keyboard. Therefore, the touch points of her left hand were skewed towards the left-bottom of the keyboard, which means that the compensation should be performed to the right. Only a double shifted keyboard would be able to correct this problem without creating new ones; that is, a keyboard that shifts the touch points of the left side of the keyboard to the right-top, and the touch points of the right side of the keyboard to the left-top.

Participants #2 and #5 did not have all their *neighbor substitutions* errors corrected, because some touch points were too far away from the intended key. This means that the vertical shift would have to be higher than 10% of the height of the key. Participant #2 had a touch point (*neighbor substitution* error) that would need to be shifted 29% of the height of the key to be corrected. These errors are not possible to correct without creating new ones.

All the committed *failed omissions* were corrected by the *Shifted* variant. Most of the cases were near the ”A” key, at the bottom. Other cases were near the ”L” key and near the space bar.

Size Invisible Simulation. The *Size Invisible* variant uses the *letter prediction* algorithm described in Section 4.2 to increase the underlying area of the most likely keys. In order to find the optimal size increase that fits all users, we performed several simulations. These simulations combined increasing the height of the key between 5% and 95% of its height (increments of 5%), and the width of the key between 5% and 95% of its width. The best result was obtained with an increased size of 5% for the width and height of the key. Figure 6.11(b) shows a *Heat map* with the results for all the tested sizes.

Just because this is the generic size increase that suits all the participants, it does not mean that it is the best size increase for each participant. For instance, participants #5, #7, #8 and #17 benefit more with a different horizontal and vertical increase size. Participant #2 and #17 had better results without increasing the size of the most probable keys, and participants #1, #3-#6, #9, #10, #13, #14, #16, #18 and #19 did not benefit with this variant. Similar to the *Shifted* variant, we also have to take into account that this variant will only benefit users that have made *neighbor substitution*, *failed omission* and/or *slide omission* errors. Therefore, users #1, #4, #9, #16, #18 and #19 (from the list above) could have never benefited from this variant, since they did not commit any of those errors.

Participants #3, #6 and #10 (also from the list above) and #2, #5, #11, #12, #14, #17 and #20 did commit *slide omissions* that were not solved. Participant #3 committed a slide from "S" to "Z" ("S" being the intended key) that is impossible to solve without creating new errors; in order to solve this error, the size of the key would need to be increased 46% (23% up and 23% down) instead of 5%, vertically. Participant #11 committed a slide that could have been corrected if the increase size was 6% instead of 5%, horizontally. But since our simulation only tested values from 5% to 95% with increments of 5%, this value was never tested. The remaining errors from the other participants were not corrected by the *Size Invisible* variant because: (1) it was at the beginning of a word. In such case the *Size Invisible* variant has no key highlighted; (2) the user had already committed a mistake, therefore the *prediction system* was not able to highlight any key (or at least, not the right key); (3) participants performed a slide starting at the space bar and ending at the bottom of it, at pixel 340 (y axis) (as we already reported in subsection 6.2.7, this was due to a problem in our implementation).

Most of the *neighbor substitutions* were not corrected by some of the same reasons stated above. 31% of the *neighbor substitutions* were not corrected because the user was typing the first letter of the word; 34% because the participants had already made a mistake; 7% because the intended word was not in the dictionary, which means that, even though there were no errors, the intended key was not highlighted; and finally, 10% because either the height or width increased was not enough.

Regarding *failed omission* errors, 42% were not corrected because the user was typing the first letter of the word; 37% because the participants had already made a mistake, and only 21% were corrected.

Single Touch Simulation. The *Single Touch* variant emerged mainly from two facts: (1) some older users interacted with more than one finger unintentionally, which lead to many errors classified as *extra-finger insertions*; (2) other users interacted with two fingers intentionally, but did not realize that it meant that both touches were recognized, leading to more *extra-finger insertion* errors. We were willing to try this simulation, since the keyboard Nicolau [14] used in his study was single touch (although in his case it was a hardware limitation). Still, we were reluctant, since five of our participants interacted with the keyboard with *both hands*; we could, potentially, omit touches that were not supposed to be omitted. Results show that participants #8-#10 were slightly hindered and participants #2, #3, #5, #11, #12, #14, #17, #18 and #20 were benefited. Remaining participants were not positively or negatively affected by this variant.

When downgrading the virtual keyboard from multi-touch to single touch, there were two possible outcomes regarding the *extra-finger insertion* errors. Let's assume the user wants to tap the "E" key, but accidentally also touches the space bar. Depending on the order of the presses, a *good* or a *bad* outcome can occur. If the user presses the "E" key first, and before releasing it, presses the space bar, the space bar is omitted (*good* outcome). Contrarily, if the user presses the space bar first, and before releasing it, presses the "E" key, the "E" character is omitted (*bad* outcome).

Results show that *Single Touch* was able to provide a *good* outcome 83.60% of the times and a *bad* outcome on the remaining times. Participants #2 and #17 benefited highly with this variant. Participant #2 made a lot of unintentional touches with her thumb when using the *QWERTY* keyboard, that were later corrected by the *Single Touch* variant. Participant #17 also made several *extra-finger insertion* errors that were not fully understood, as we reported in subsection 6.2.5. Although we were not able to fully understand what happened, *Single Touch* was able to correct most of those errors.

Intra-key Timed Simulation. This variant emerged mainly to correct *accidental insertions*. This kind of error is characterized by a reduced time interval between the press and release of a key. The problem is that, this time interval is highly user-dependent, and even within a specific user, it can vary a lot. Therefore, we performed several simulations in order to find the right threshold for each user, which would allow maximizing the rejection of *accidental insertions*, and minimizing the rejection of false positives; i.e., correct characters that happen to have a time interval below the threshold. To do this, we varied the threshold between 25ms and 750ms with increments of 25ms. Table A.8 shows the optimum threshold(s) for each participant. We can verify that, the threshold can be as low as 25ms (participants #1, #4, #7-#10, #14, #15, #18 and #19) and as high as 250ms (participant #17). Just to show how disastrous it is to attribute the optimum threshold of one user to another, let's take a look at the following example: if we fix the threshold value at 250ms (optimum threshold for user #17) for participant #1, the result would be an increase of 141 errors when compared with the optimum threshold of participant #1.

This is why a generic solution will perform worse than an adaptive one (Table A.9). Figure 6.12 (a) shows that the threshold that yields better results $(MSD = 240)$ for the generic *Intra-key* solution is 25ms.

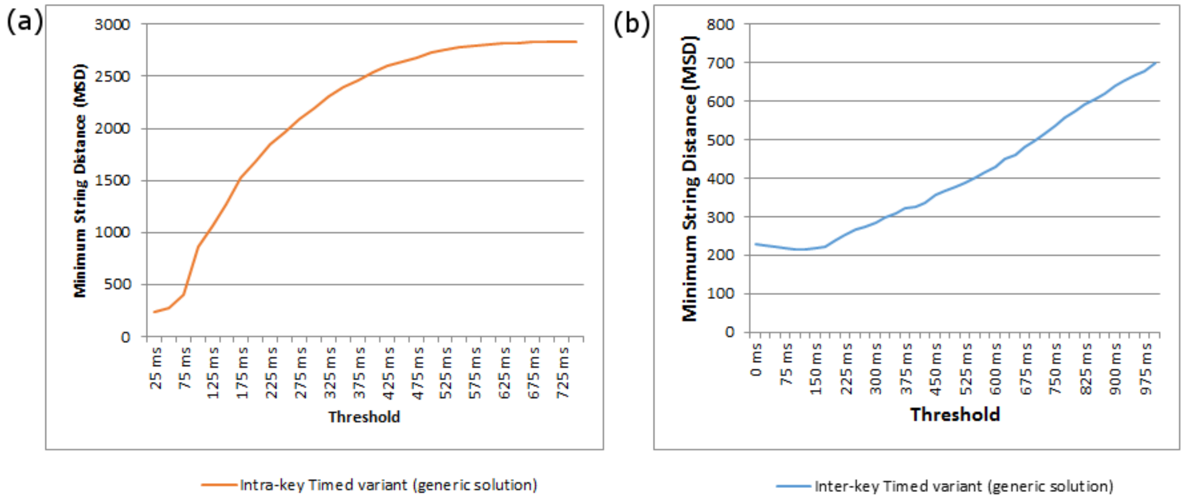


Figure 6.12: Minimum String Distance (MSD) for each of the simulated thresholds for the (a) Intra-key Timed variant and (b) Inter-key Timed variant.

When compared with *QWERTY's* results, participants #2, #3, #5, #6, #11, #12, #16, #17 and #20 benefited with the adaptive *Intra-key* variant. Participants #1, #7-#10, #13, #14 and #18 were hindered while participants #4, #15 and #19 were not positively nor negatively affected. When compared with *Single Touch's* results, participants #2, #3, #5, #6, #11, #12, #16, #17 and #20 benefited with the adaptive *Intra-key* variant. Participants #1, #7, #8, #10, #13 and #14 were hindered while participants #4, #9, #15, #18 and #19 were not positively nor negatively affected.

This variant was able to correct 75% of the *accidental insertion* errors and 55.56% of the *double insertion* errors. Before looking at the results, we were not expecting *double insertions* to be corrected by the *Intra-key* variant. In order to better understand why *double insertions* were corrected by the *Intra-key* variant, we analyzed in detail this kind of error. In terms of output, a word that contains a *double insertion* error will have an extra character repeated (e.g.: ”woord”). As previously stated in subsection 6.2.5, when we were classifying the errors, the repeated character was always the one considered an error. For instance, in one of the sentences, participant #16 typed ”aaos” instead of ”aos”. The second ”a” was considered a double insertion error because it was a repeated character and because the time elapsed between the release of the first ”a” and the press of the second ”a” was 64ms. The results of the *Intra-key* show that the first ”a” of ”aaos” was omitted. This made us realize that there are different patterns when committing *double insertions*: (1) the user types the first ”a” intentionally, and then accidentally types a second ”a”; (2) the user intends to tap the ”a” key, but first taps it accidentally, and then intentionally; (3) finally, a mix of both, were, although there is the intention to tap the ”a” key, both touches are fast, therefore could be considered *accidental*. Results show that 55.56% of the times, *double insertions* were caused by an accidental first touch.

This variant was also responsible for the emergence of 32 new errors. This is precisely because finding the optimum threshold is not an easy task; most of the times it is impossible to set a threshold that rejects all the *accidental insertion* errors, without rejecting correct characters. Because of this, the *Intra-key* variant does not perform as well as we thought it would.

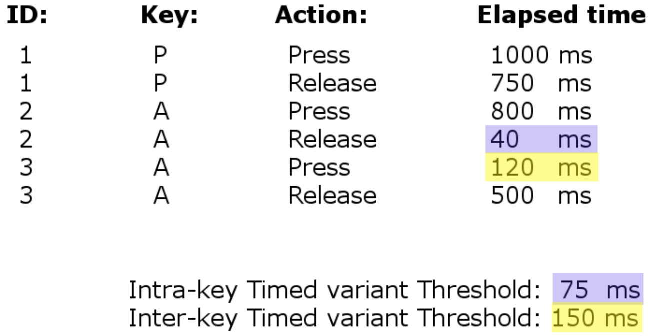
Inter-key Timed Simulation. The *Inter-key Timed* variant emerged mainly to correct *double insertions*. This kind of error is characterized by the insertion of a second character with a reduced time interval between the release of the first key, and the press of the second key. The problem is that, this time interval is highly user-dependent, and even within a specific user, it can vary a lot. Therefore, we performed several simulations for each user in order to find the right threshold for each user, which would allow maximizing the rejection of *double insertions*, and minimizing the rejection of false positives; i.e., correct characters that happen to have a time interval below the threshold. To do this, we varied the threshold between 25ms and 1000ms (1 second) with increments of 25ms. Table A.8 shows the optimum threshold(s) for each participant. We can verify that, the threshold can be as low as 25ms (participants #1, #4, #7, #9 and #19) and as high as 875ms (participant #17). Just to show how disastrous it is to attribute the optimum threshold of one user to another, let’s take a look at the following example: if we fix the threshold value at 875ms (optimum threshold for user #17) for participant #1, the result would be an increase of 51 errors when compared with the optimum threshold of participant #1.

This is why a generic solution will perform worse than an adaptive one (Table A.9). Figure 6.12 (b) shows that the threshold that yields better results $(MSD = 215)$ for the generic *Inter-key* solution is 125ms.

When compared with $QWERTY's$ results, participants #2-#4, #6, #8, #11-#13, #15-#17 and #20 benefited with the *Inter-key* variant. Participants #1, #9 and #18 were hindered and participants #5, #7, #10, #14 and #19 were not positively nor negatively affected. When compared with *Single Touch’s* results, participants #2-#6, #10, #11-#17 and #20 benefited with the *Inter-key* variant. Participants #1 and #18 were hindered and participants #7-#9 and #19 were not positively nor negatively affected.

This variant was able to correct 85.19% of the *double insertions* and it also created 8 new errors. This happened because finding the optimum threshold is not an easy task; most of the times it is impossible to set a threshold that rejects all the double insertions, without rejecting correct characters.

Combined Timed Simulation. The *Combined Timed* variant is the combination of *Intra-key* and *Inter-key Timed* variants. Therefore, it will be able to correct *accidental* and *double insertion* errors, mainly. Both parts keep track of time in order to operate. However, the two parts are independent of each other, which mean that each one will have its threshold and operate independently. Still, it is important to acknowledge that, since the *Intra-key* part operates on the same interaction (i.e., compares the time elapsed between the press and the release of the same interaction), it can prevent *Inter-key* part from operating on certain cases. This means that, if we have a case like in Figure 6.13, were both *Intra-key* and *Inter-key* would operate on their own (*Intra-key* would discard the interaction with id 2 while *Inter-key* would discard the interaction with id 3), when combined, only *Intra-key* will operate. This happens because, *Intra-key* will first check if the press and release time of the touch interaction with id 2, is less than the threshold (let's assume *Intra-key* threshold is 75ms and *Inter-key's* is 150ms); if true, that touch interaction will be omitted. After the press interaction with id 3, *Inter-key* will verify if the elapsed time between that press and the previous release is less than the threshold. But, since touch interaction with id 2 was omitted, the press of id 3 will be compared with the release of id 1; in this case, the interaction with id 3 will not be omitted, because the elapsed time between the release of the key with id 1 and the press of the key with id 3 is higher than 150ms.



ID:	Key:	Action:	Elapsed time
1	P	Press	1000 ms
1	P	Release	750 ms
2	A	Press	800 ms
2	A	Release	40 ms
3	A	Press	120 ms
3	A	Release	500 ms

Intra-key Timed variant Threshold: 75 ms
Inter-key Timed variant Threshold: 150 ms

Figure 6.13: Example that illustrates a case on which Intra-key and Inter-key would operate on their own, but when combined (Combined Timed variant), only Intra-key operates.

This is the reason why the *Combined Timed* variant was not able to perform better than *Inter-key* in general, and regarding *double insertions* in particular. Because the *double insertions* corrected by the *Intra-key* variant independently, were also corrected by the *Inter-key* variant independently. Therefore, when combined, only one of the parts corrected those errors. Therefore the *Combined* variant corrected exactly 85.19% of the *double insertions*, just as *Inter-key*. Regarding *accidental insertions*, it corrected the same as *Intra-key's* (75%). In the end, the Combined variant performed worse than Inter-key because it created more new errors (35).

Similar to the *Intra-key* and *Inter-key* variants, the adaptive solution is better than the generic solution (Table A.9). The best generic thresholds for the *Combined* variant are the same found for each of the independent variants; that is, the thresholds that yield better results $(MSD = 226)$ for the *Combined* variant are 125ms and 25ms, for the *Intra-key* and *Inter-key* threshold, respectively (Figure 6.14).

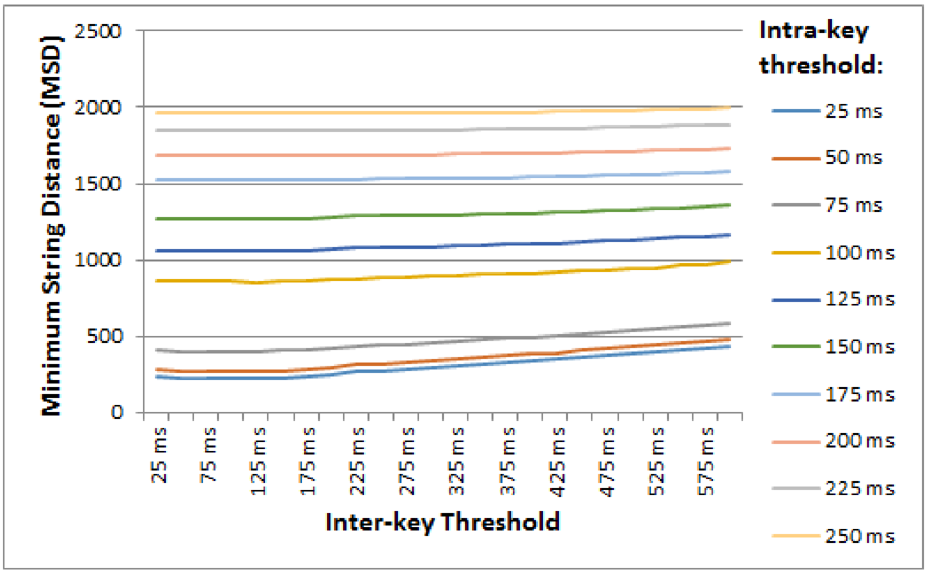


Figure 6.14: Minimum String Distance (MSD) for the combined thresholds for the Combined Timed variant.

Overall Results. The *Shifted* variant performed really well if we take into account the type of errors it can correct. We only analyzed thoroughly the (best) generic solution (H=2%; V=10%), which corrected 58.62% of the *neighbor substitution*, 100% of the *failed omission* and 39.39% of the *slide omission* errors. This solution is more adequate for users who use only their right hand to type. But, even when looking at the deviation of the key center of each user that used the same interaction method, we found that it still is user-dependent. Therefore, an adaptive model that constantly updates the center of each individual key seems to be the best solution. Also, if we take into account that the same user can present different touch typing patterns, depending on the hand posture used for typing [48], an adaptive model that recognizes various hand postures will even perform better.

The *Size Invisible* variant was not able to perform as well as the *Shifted* variant. Most of the times the *prediction system* was not able to operate because the error was at the beginning of a word, the user had already made a mistake, the word was not in the dictionary, or because the key was not big enough. We only analyzed thoroughly the (best) generic solution (H=5%; V=5%), which corrected 17.24% of the *neighbor substitutions*, 21.05% of the *failed omissions* and 12.12% of the *slide omissions*.

The *Single Touch* variant performed well for some participants, but also hindered others slightly. This is because omitting characters can, sometimes, cause the omission of correct characters. Therefore, there is some risk associated with this variant, as opposed to the previous two. Still, this variant was able to correct 78.77% of *extra-finger insertions*, counting already with the 26 new errors that emerged.

Since the *Intra-key* variant aimed to correct *accidental insertion* errors, which are characterized by a reduced time interval between the press and release of a key, we had to define a threshold that would tell the keyboard when to omit a character. Since this time interval is highly user-dependent (and even within a specific user, it can vary a great deal) this variant was not able to correct errors without creating new ones. Still, this variant was able to correct 75% of *accidental insertion* errors.

Inter-key variant was similar to *Intra-key* variant, in the sense that it also needed a threshold that was highly user-dependent. Still, this variant did not create as much errors as *Intra-key* variant. *Inter-key* was able to correct 85.19% of *double insertion* errors.

Since the *Combined* variant is the combination of *Intra-key* and *Inter-key* variants, it aimed to correct *accidental* and *double insertion* errors. Although this means that it has Intra-key's and Inter-key's advantages, it also means it has their disadvantages; that is, creating new errors. Therefore, this variant was able to correct 75% and 85.19% of *accidental* and *double insertion* errors, respectively. But, it even created more errors than the *Intra-key* variant.

6.5 Discussion

Our goal was to investigate how older users inputted text in tablet devices in order to improve their performance. In this section we discuss the obtained results by: 1) answering the previously proposed research questions and 2) identifying implications for design.

6.5.1 Answering Research Questions

After analyzing all data, we are able to answer the research questions proposed at the beginning of the user and simulation studies. User study research questions:

1. *Will inexperienced older users who are not acquainted with the QWERTY keyboard layout perform better with the Color variant?*
   Older users in general, as well as inexperienced older users in specific, performed worse with the *Color* variant regarding input speed. Regarding accuracy, both keyboards performed roughly the same.

2. *Will older users perform better with the Predict Words variant?*
   We are not able to completely answer this question, since most of the users (13) used the *Predict Words* variant as a normal *QWERTY* keyboard. We hypothesize this is the reason why *Predict Words* performance was similar to *QWERTY's*. Still, when taking into account only users who accepted at least one word from the suggested list, *Predict Words* performed worse regarding input speed. Regarding accuracy, no significant differences were found.

3. *How do older adults type on touchscreens regarding accuracy, speed and typing strategies?*
   Participants achieved a maximum of 17.25 wpm using the *Color* variant (mean of 6.19, 5.42 and 5.51 wpm for *QWERTY*, *Color* and *Predict Words* conditions, respectively). Also, input speed was strongly correlated with previous QWERTY experience. Users obtained a minimum *MSD* error rate of 0.61% (mean of 9.76%, 10.30% and 9.61% for *QWERTY*, *Color* and *Predict Words* conditions, respectively). Error rate was not correlated with previous QWERTY experience, suggesting that having some practice with keyboards is not sufficient to compensate the challenges that are imposed by touch devices.

4. *What are the most common types of errors committed by older adults?*
   Insertion errors, specifically *extra-finger insertion* errors are the most common type of errors committed by older adults. Still, this type of error was mainly committed by two participants (#2 and #17) and is due to our implementation being multi-touch. The most common error

committed by almost all participants is *omission* errors. This kind of error occurred across keyboard conditions, suggesting that most *omission* errors are keyboard-independent. This is odd since one of the strengths of the Color variant is to draw the user's attention to the next four most probable keys, which should minimize *omission* errors (specifically *cognitive omission* errors). Following *omissions* were *substitutions* and *empty* errors. *Substitutions* were mainly due to poor aiming (i.e. users landed their finger on the wrong key), while *empty* errors were due to accidental touches around the area of the keyboard.

5. *Does tremor affect text-entry performance? If yes, how is users' performance correlated with hand tremor?*
   The tremor data we collected did not allow us to draw important conclusions. We only found a moderate-to-weak negative correlation between input rate and task-specific tremor for the right hand, which suggests that users with more tremor type text at a lower input rate.

Simulation study research questions:

1. *Will the Shifted variant be able to reduce significantly neighbor substitution, failed omission and slide omission errors? Will a generic approach be enough?*
   The *Shifted* variant was able to correct 58.62% of the *neighbor substitution*, 100% of the *failed omission* and 39.39% of the *slide omission* errors; thus reducing significantly each type of error. The generic approach that shifts all touch interactions as if they were performed only by the right hand, performed well. Still, better results will be obtained if we consider at least three different groups of users (users who use only their *right hand*, users who use only their *left hand* and users who use *both*), and perform the shifts accordingly.

2. *Will the Size Invisible variant be able to reduce significantly neighbor substitution, failed omission and slide omission errors? Will a generic approach be enough? How does it compare with the Shifted variant?*
   The *Size Invisible* variant was able to correct 17.24% of the *neighbor substitution*, 21.05% of the *failed omission* and 12.12% of the *slide omission* errors; thus reducing significantly each type of error. The generic approach that increases the size of the most probable keys 5% of the width of the key horizontally and 5% of the height of the key vertically performed roughly the same when compared with the adaptive solution. Therefore, a generic approach is enough (although, an adaptive solution always yields better results). When compared with the *Shifted* variant, it did not perform as well, mainly because this variant is not able to operate when the user already made a mistake. We could argue that, in order for this variant to perform better, a *prediction algorithm* that operates at the beginning of words would be required; that is, an algorithm that takes into account the probability of appearance of each word and the relative probability of appearance of every syntactic category after each syntactic category [58]. Still, if users commit an error at the beginning of a word, the *prediction system* will be unable to operate.

3. *Will the Single Touch variant be able to reduce significantly extra-finger insertion errors? Will this work for users who interact with both hands?*
   The *Single Touch* variant was able to correct 78.77% of the *extra-finger insertion* errors (counting already with the new 26 errors that emerged). This variant benefited mostly users who made *accidental insertions* by interacting intentionally with one finger and unintentionally with other (simultaneously). Since this variant disables multi-touch, we were afraid that users who used

both hands to type would be greatly hindered. Of the five participants who used *both hands* to interact, only one benefited with this variant, but none was hindered.

4. *Will the Intra-key Timed variant be able to reduce significantly accidental insertion errors? Will a generic approach be enough?*
   The *Intra-key Timed* variant was able to correct 75% of the *accidental insertion* errors, but it also originated 32 new errors. This variant was also able to correct 55.56% of the *double insertion* errors which indicates that *double insertion* errors can be characterized in different ways: (1) an initial accidental interaction, followed by an intentional one; (2) a first intentional interaction, followed by an unintentional one; (3) or, two unintentional interactions. We only analyzed thoroughly the adaptive approach; that is, the approach that calculates the best threshold for each individual user. Even with this approach it is hard to find a threshold value that maximizes the omission of accidental insertion errors and minimizes the rejection of false positives; i.e., correct characters that happen to have a time interval below the threshold. The generic approach had catastrophic results, because users have different interaction patterns regarding the time that elapses between the press and release of a key.

5. *Will the Inter-key Timed variant be able to reduce significantly double insertion errors? Will a generic approach be enough?*
   The *Inter-key Timed* variant was able to correct 85.19% of the *double insertion* errors, but it also originated 8 new errors. We only analyzed thoroughly the adaptive approach; that is, the approach that calculates the best threshold for each individual user. Even with this approach it is hard to find a threshold value that maximizes the omission of *double insertion* errors and minimizes the rejection of false positives; i.e., correct characters that happen to have a time interval below the threshold. The generic approach was not as good as the adaptive one, because users have different interaction patterns regarding the time that elapses between the release of a key and the press of a second key. Still, it was not as bad as *Intra-key's* generic approach.

6. *Will the combination of the Intra-key and Inter-key Timed variants (Combined Timed variant) be able to reduce significantly accidental and double insertion errors? Will a generic approach be enough?*
   The *Combined Timed* variant was able to correct 75% and 85.19% of *accidental* and *double insertion* errors, respectively. This variant also created more new errors (35) than the Intra-key and *Inter-key* variants. The comments regarding the thresholds of both *Intra-key* and *Inter-key* variants apply here, since this variant is the combination of both. Similarly to *Intra-key* and *Inter-key*, the generic approach is not enough.

6.5.2 Design Implications

From the results, we derive the following design implications for text-entry solutions in tablet devices for older adults.

Keep visual changes to a minimum. As verified in the user study, visual changes that aim to focus the user attention on the most probable keys, have a negative impact in text-input speed. Also, the *Color* variant had twice the *cognitive substitution* errors, when compared with the *QWERTY* condition. When further analyzing this kind of error, we found out that 44.83% of the times, the substitution character inserted by the user was highlighted, while the expected one was not. Although we cannot tell for sure, we think users were influenced by the highlighting of the keys. Therefore, visual changes should only occur to give feedback about the pressed and released key.

Shift the touch points to the top and to the opposite side of the hand the user is using to type. As verified in the user study, different touch patterns emerge from different interaction techniques. For instance, users who used only their *right hand* to interact with the virtual keyboard had a tendency to touch on the bottom-right of targets. This means that this kind of users benefit with a top-left shift of their touch points to compensate the tendency. Conversely, users who only used their *left hand* benefit with a top-right shift of their touch points. Users who interacted with *both hands* will, hypothetically, benefit with a top-left shift of the touch points performed on the right side of the keyboard (keys: y, u, i, o, p, h, j, k, l, b, n and m), and a top-right shift of the touch points performed on the left side of the keyboard (keys: q, w, e, r, t, a, s, d, f, z, x, c and v).

When a vertical slide occurs between two keys of subsequent rows, assign the press and release to the key in the row above. We verified in our user study that, when users perform a vertical slide from one row to a subsequent row (up or down), most of the times the user intends to select the key from the row above. This was verified across variants. When the slide was performed from the bottom to the top, the user intended to tap the key in the row above 100% of the times, for both the *QWERTY* and *Color* conditions. When the slide was performed from the top to the bottom, the user also intended to tap the key in the row above 85.71% and 100% of the times, for the *QWERTY* and *Color* conditions, respectively. In the remaining 14.29% times (*QWERTY* keyboard), we do not know exactly what were the intentions of the user, since the slide was completely accidental.

Choose single touch over multi-touch. Older users are all different with different necessities and capabilities. If there is a need of including all types of older users, *Single Touch* is the right choice. The quality of the sentences of two of the most problematic participants (#2 and #17) increased drastically, with few new errors for other participants.

Omit touch interactions that are below a certain threshold (between press and releases of different interactions). During the simulation study, we verified that the *Inter-key* variant was able to increase the quality of the transcribed sentences, when compared with the *Single Touch* variant. This means that, having a keyboard that omits interactions based on the time elapsed between the release of a first key and the press of a second key can enhance older adults' accuracy. The threshold value must be defined carefully since this value is highly user-dependent.

Allow personalization. We found patterns that emerge from the different interaction methods (only *right hand*, only *left hand* or *both*) adopted by the participants. Still, when looking at each participant of the same group, we can verify that some of them have different typing behaviors, particularly hit point locations and inter-key time interval (*Inter-key* variant). Therefore an adaptive model that constantly updates the center of each individual key seems the best solution.

6.5.3 Limitations

The user study reported in this chapter does not contemplate error correction. While this was necessary to assess natural typing patterns, understanding users' correcting strategies is also needed.

Further research should focus on reporting error correction effect on touch-based devices.

Participants interacted mainly with their *right hand* (13). Following studies should also focus in users that only use their *left hand*, and users that use *both hands*, in order to verify the new hypotheses that emerged from this study.

Participants did not have much time to experiment with the *Color* and *Predict Words* variants, before beginning the evaluation. We believe that this might have hindered the performance of the users regarding these variants. It may also be the reason why the *Predict Words* variant was mainly used as a normal *QWERTY* keyboard. Another study should be performed on which more time should be provided for users to get acquainted with the new variants.

Regarding the simulation study, *Intra-key*, *Inter-key* and *Combined Timed* variants were somehow limited since we only tried to find the optimum threshold through the use of increments of 25ms. Also, performing a simulation study is not the same as performing a user study. Future studies should allow users to experiment these new variants.

6.6 Conclusion

We investigated text-entry performance of 20 older adults on a touch-based device (tablet). Our user study featured three virtual keyboard conditions (*traditional QWERTY*, *Color* and *Predict Words*) and assessed each user tremor profile. Users typed faster with the traditional *QWERTY* keyboard condition. Regarding the quality of transcribed sentences, no significant differences were found across conditions. Errors committed were mostly *extra-finger insertions*, followed by *cognitive omissions*, *cognitive insertions*, *empty*, *neighbor substitutions*, *double insertions*, *slide omissions*, *failed omissions*, *cognitive substitutions* and finally, *accidental insertions*.

We also performed a simulation study that takes into account the typing behavior learned from the user study. We performed several simulations using six new variants that take advantage of the users' typing pattern (*Shifted* variant), uses a language model (*Size Invisible* variant), downgrades the keyboard from multi-touch to single touch (*Single Touch* variant), and uses time-based features to avoid errors (*Intra-key*, *Inter-key* and *Combined Timed* variants).

All these variants, except *Intra-key* and *Combined* were able to successfully reduce the error rate.

Lastly, we identify some design implications that should improve typing accuracy and encourage researchers to create more effective solutions for older adults. Future research should apply the design implications described here and investigate their effect on text-entry performance.

7
Conclusions

The goal of this work was to better understand how older users interact with touch devices, in order to obtain the needed knowledge to design text entry solutions more adequate for older adults.

To fulfill this thesis goal, we conducted two different studies. On the first one, the baseline study, we tested a traditional QWERTY (virtual) keyboard and five different variants (*Color*, *Width*, *Predict Words*, *Shifted* and *Size Invisible*) that were based on state-of-the-art solutions. The goal of this study was to assess which were the most promising variants and also assess if our application was robust enough to be used in the older adults user test. On the subsequent study, the study with older adults, we maintained the most promising variants (*Color*, *Predict Words*, *Shifted* and *Size Invisible*) and removed the least promising one (*Width* variant). In this study we opted to test only the variants that had visual changes (*Color* and *Predict Words*) and perform a simulation of the remaining variants by using the input data of the QWERTY condition. This decision was taken because we did not want to overload the older users with too many tasks. We also created four new variants (*Single Touch* and *Intra-key*, *Inter-key* and *Combined Timed* variants) based on Nicolau's [14] studies and our own observations during the tests. Since these variants differed from the others only regarding the processing of touch inputs we were also able to used them as simulations. Our goal in this study was to better understand older users typing patterns and find the most fitting variant(s).
In the first study (**Baseline Study**) our goal was to assess which of the variants were most promising. The ***Color* variant** was able to reduce the error rate significantly, specially *cognitive omissions*. This was possible because on the *QWERTY* condition, most of the cognitive omissions were on the space bar. Users sometimes completely missed the touch area captured by the tablet, hitting its bevel instead. On the *Color* condition they were able to acknowledge it because the space bar remained highlighted. Still, this result was obtained at the cost of reducing input rate. Even though the ***Width* variant** is very similar to the *Color* variant (in concept), the *Width* variant was not able to reduce error rate significantly, and it even performed worse than the *Color* variant regarding input rate. The ***Predict Words* variant** performed even worse than the *Width* variant, regarding error rate and input rate. Still, users were satisfied and found it easy to use. The ***Shifted*** and ***Size Invisible* variants** were able to reduce neighbor substitution errors. The *Size Invisible* variant outperformed the *Shifted* variant regarding the correction of *neighbor substitutions*, because in general we did not find the bottom-right pattern reported by other authors [14, 37, 59], on which we had based our *Shifted* variant.

In the second study (**Study With Older Adults**) our goal was to better understand older users typing patterns and find the most fitting variant(s). Older users in general, as well as inexperienced older users in specific, performed worse with the ***Color* variant**, regarding input speed. We hypothesize that the highlighting of the keys distracted the users and also lead to more cognitive substitutions. Still, users were satisfied with the *Color* variant, and emphasized it was a big help, since it helped them to locate the desired key. The ***Predict Words*** **variant** was as good as the traditional QWERTY keyboard in terms of input speed and error rate. Still, thirteen of the participants used the *Predict Words* variant as the *traditional QWERTY* keyboard; that is, ignoring the suggested words. The ***Shifted*** **simulation** was able to reduce significantly the error rate, specifically *neighbor substitution*, *failed omission* and *slide omission* errors. This was possible because the touch typing pattern of our users was similar to the one described by several authors [14, 37, 59] and the compensating shift we applied was based on the results of those authors. Furthermore, we found that **vertical shifts increase gradually from row to row** until we reach the space bar. We also found that the **horizontal shift pattern is closely related with the hand the user is using to type** as hypothesized by Nicolau [14]; that is, the **shift is more intense towards the side of the keyboard of the hand the user is using to type**. For instance, when users interact only with their right hand, the horizontal shift is more intense to the right on the right side of the keyboard. When users interact with both hands, the horizontal shift is more intense to the right on the right side of the keyboard, when compared with the center of the keyboard. Conversely, the horizontal shift is more intense to the left on the left side of the keyboard. Even though the ***Size Invisible*** **simulation** did not perform as well as the *Shifted* variant, it was able to reduce significantly the error rate, specifically *neighbor substitution*, *failed omission* and *slide omission* errors.

We also found that, in general, a single touch keyboard is more advantageous for older users than a multi-touch one. The ***Single Touch*** simulation was able to reduce significantly *extra-finger insertions*. Regarding the Timed simulations, the ***Intra-key*** did not perform as well as we expected, because it is hard to define a threshold that allows maximizing the rejection of *accidental insertion* errors, and minimizing the rejection of false positives. Although it is also hard to define a threshold for the ***Inter-key Timed*** **variant**, it was able to reduce significantly ***double insertion*** **errors**. The ***Combined Timed*** **variant** did not perform as well as we expected, mainly because of the Intra-key threshold. Finally, a result that, to our knowledge, has **never** been reported by other authors, is that **when users perform a vertical slide from one row to a subsequent row (up or down), 96.4% of the times the user intends to select the key from the row above.**

From our results we conclude that it is indeed possible to develop text-entry solutions more adequate for older adults. Still, our approach has several limitations, which should be addressed in future works.

7.1 Future Work

Re-Evaluate the *Color* and *Predict Words* variants. In our user study, we only allowed each older user to type two sentences with each variant before beginning the evaluation. A new study should be performed that allowed users to experiment each of these variants for a longer period of time (e.g.: one week). After this week, a study should be performed, to assess if the users were able to improve their performance on these two variants.

Evaluate the *Shifted* variant with more participants that only use their left hand and/or both hands to type. The majority of our participants used only their right hand to type. This allowed us to investigate the touch typing behavior of these users. Since less users used their left hand and both hands, our results were not as strong. Therefore, a new study should be performed in order to verify the touch typing patterns reported by us.

A *conditional Single Touch* variant, that rejects interactions based on a prediction algorithm should be developed and evaluated. Our Single Touch approach was limited, because it rejected interactions based only in the order of touch interactions (it rejected a second press, if the first press was not yet released). A new approach should reject touch interactions, based on the probability of each of the pressed keys. The same could also be applied to the *Inter-key* variant.

The *Single Touch* and *Inter-key Timed* variants should be evaluated in a user study (instead of a simulation study). Our simulation indicates that these variants are promising, still it is important to assess how would users react to them. In the case of the Inter-key variant, it would also be important develop a mechanism that adapts the threshold to the user, depending on its past interactions.

Bibliography

[1] WHO, “Global health and aging,” http://www.who.int/ageing/publications/global_health.pdf, December 2012.

[2] R. Pak and A. McLaughlin, *Designing Displays for Older Adults*, ser. Human Factors & Aging Series. CRC PressINC, 2010.

[3] A. D. Fisk, W. A. Rogers, N. Charness, S. J. Czaja, and J. Sharit, *Designing for Older Adults: Principles and Creative Human Factors*, 2nd ed., ser. Human Factors and Aging Series. CRC Press, 2009.

[4] C. J. Ketcham, R. D. Seidler, A. W. Van Gemmert, and G. E. Stelmach, “Age-related kinematic differences as influenced by task difficulty, target size, and movement amplitude,” *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 2002.

[5] Medline Plus, “Aging changes in the bones - muscles - joints,” http://www.nlm.nih.gov/medlineplus/ency/article/004015.htm, December 2012.

[6] L. L. Carstensen and C. R. Hartel, *When I’m 64*, 1st ed. The National Academies Press, 2008.

[7] M. Kobayashi, A. Hiyama, T. Miura, C. Asakawa, M. Hirose, and T. Ifukube, “Elderly user evaluation of mobile touchscreen interactions,” in *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part I*, ser. INTERACT’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 83–99. [Online]. Available: http://dl.acm.org/citation.cfm?id=2042053.2042065

[8] F. Werner, K. Werner, and J. Oberzaucher, “Tablets for seniors - an evaluation of a current model (ipad),” in *Ambient Assisted Living*, ser. Advanced Technologies and Societal Change, R. Wichert and B. Eberhardt, Eds. Springer Berlin Heidelberg, 2012, pp. 177–184.

[9] M. Annett, F. Anderson, D. Goertzen, J. Halton, Q. Ranson, W. F. Bischof, and P. Boulanger, “Using a multi-touch tabletop for upper extremity motor rehabilitation,” in *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group: Design: Open 24/7*, ser. OZCHI ’09. ACM, 2009, pp. 261–264.

[10] T.-H. Tsai, H.-T. Chang, Y.-M. Chang, and G.-S. Huang, “Sharetouch: A system to enrich social network experiences for the elderly,” *J. Syst. Softw.*, vol. 85, no. 6, pp. 1363–1369, Jun. 2012.

[11] T. Apted, J. Kay, and A. Quigley, “Tabletop sharing of digital photographs for the elderly,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’06. ACM, 2006, pp. 781–790.

[12] A. M. Piper, R. Campbell, and J. D. Hollan, “Exploring the accessibility and appeal of surface computing for older adult health care support,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’10. ACM, 2010, pp. 907–916.

[13] M. R. Victoria Meza-Kubo, Alberto L Morán, “Intouchfun, a pervasive collaborative system to cope with elder’s isolation and cognitive decline,” *Proceedings of the UbiComp*, 2010.

[14] H. Nicolau and J. Jorge, “Elderly text-entry performance on touchscreens,” in *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ser. ASSETS ’12. New York, NY, USA: ACM, 2012, pp. 127–134. [Online]. Available: http://doi.acm.org/10.1145/2384916.2384939

[15] WHO, “Definition of an older or elderly person,” http://www.who.int/healthinfo/survey/ageingdefnolder/en/index.html, December 2012.

[16] D. Quillen, “Common causes of vision loss in elderly patients,” http://www.aafp.org/afp/1999/0701/p99.html, December 2012.

[17] M. A. Bullimore, J. M. Wood, and K. Swenson, “Motion perception in glaucoma,” *Investigative Ophthalmology & Visual Science*, 1993.

[18] Glaucoma Research Foundation, “Can glaucoma affect contrast sensitivity?” http://www.glaucoma.org/q-a/can-glaucoma-affect-contrast-sensitivity.php, December 2012.

[19] WHO, “Vision 2020 action plan,” http://www.who.int/ncd/vision2020_actionplan/contents/3.5.1.htm, December 2012.

[20] K. S. Coyne, M. K. Margolis, T. Kennedy-Martin, T. M. Baker, R. Klein, M. D. Paul, and D. A. Revicki, “The impact of diabetic retinopathy: perspectives from patient focus groups,” *Family Practice*, 2004.

[21] M. I. Wallhagen, E. Pettengill, and M. Whiteside, “Sensory impairment in older adults part 1: Hearing loss,” *American Journal of Nursing*, 2006.

[22] S. V. Saxon, M. J. Etten, and E. A. Perkins, *Physical change & aging: a guide for the helping professions*, 5th ed. Springer Publishing Company, 2009.

[23] American Speech-Language-Hearing Association, “Types of hearing loss,” http://www.asha.org/public/hearing/disorders/types.htm, December 2012.

[24] N. J. Nusbaum, “Aging and sensory senescence,” *Southern Medical Journal*, 1999.

[25] M. M. Wickremaratchi and J. G. Llewelyn, “Effects of ageing on touch,” *Postgraduate Medical Journal*, 2006.

[26] E. Wood, T. Willoughby, A. Rushing, L. Bechtel, and J. Gilbert, "Use of computer input devices by older adults," *The Journal of APplied Gerontology*, 2005.

[27] J. D. Kester, A. S. Benjamin, A. D. Castel, and F. I. M. Craik, *The Handbook of Memory Disorders*. John Wiley & Sons, Ltd., 2002.

[28] J. C. Cavanaugh and F. Blanchard-Fields, *Adult Development And Aging*. Cengage Learning, 2006.

[29] D. R. Riddle, "Brain aging: Models, methods, and mechanisms," 2007.

[30] P. Verhaeghen and J. Cerella, "Aging, executive control, and attention: a review of meta-analyses," *Neuroscience & Biobehavioral Reviews*, 2002.

[31] Nielsen Norman Group, "Mental models," http://www.nngroup.com/articles/mental-models/, December 2012.

[32] F. Ruotolo, T. Iachini, G. Ruggiero, and R. Pizza, "Age and gender differences in some components of spatial cognition," 2008.

[33] W. A. Rogers, A. D. Fisk, A. C. McLaughlin, and R. Pak, "Touch a screen or turn a knob: Choosing the best device for the job," *Human Factors*, vol. 47, no. 2, pp. 271–288, 2005.

[34] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '09. New York, NY, USA: ACM, 2009, pp. 1083–1092. [Online]. Available: http://doi.acm.org/10.1145/1518701.1518866

[35] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "Understanding users' preferences for surface gestures," in *Proceedings of Graphics Interface 2010*, ser. GI '10. Toronto, Ont., Canada, Canada: Canadian Information Processing Society, 2010, pp. 261–268. [Online]. Available: http://dl.acm.org/citation.cfm?id=1839214.1839260

[36] C. Stöbel, "Gestural interfaces for elderly users - help or hindrance?" Master's thesis, Faculté des Sciences, de la Technologie et de la Communication, 2012.

[37] N. Henze, E. Rukzio, and S. Boll, "100,000,000 taps: analysis and improvement of touch performance in the large," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, ser. MobileHCI '11. New York, NY, USA: ACM, 2011, pp. 133–142.

[38] C. Wacharamanotham, J. Hurtmanns, A. Mertens, M. Kronenbuerger, C. Schlick, and J. Borchers, "Evaluating swabbing: a touchscreen input method for elderly users with tremor," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 623–626. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979031

[39] C. Wacharamanotham, “Swabbing: touchscreen-based input technique for people with hand tremor,” in *Work: A Journal of Prevention, Assessment and Rehabilitation*, 2012.

[40] C. Stöbel, “Familiarity as a factor in designing finger gestures for elderly users,” in *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*, ser. MobileHCI ’09. New York, NY, USA: ACM, 2009, pp. 78:1–78:2. [Online]. Available: http://doi.acm.org/10.1145/1613858.1613950

[41] Z. X. Jin, T. Plocher, and L. Kiff, “Touch screen user interfaces for older adults: button size and spacing,” in *Proceedings of the 4th international conference on Universal access in human computer interaction: coping with diversity*, ser. UAHCI’07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 933–941. [Online]. Available: http://dl.acm.org/citation.cfm?id=1766311.1766419

[42] T. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves, “Towards accessible touch interfaces,” in *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*, ser. ASSETS ’10. New York, NY, USA: ACM, 2010, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/1878803.1878809

[43] H. Nicolau, T. Guerreiro, J. Jorge, and D. Gonçalves, “Towards mobile touch screen inclusive user interfaces: Differences and resemblances between motor-impaired and able-bodied users,” in *Proceedings of the 13th IFIP TC13 Conference on Human-Computer Interaction (INTERACT’11)*, 2011.

[44] J. Himberg, J. Häkkilä, P. Kangas, and J. Mäntyjärvi, “On-line personalization of a touch screen based keyboard,” in *Proceedings of the 8th international conference on Intelligent user interfaces*, ser. IUI ’03. New York, NY, USA: ACM, 2003, pp. 77–84.

[45] L. Findlater and J. Wobbrock, “Personalized input: improving ten-finger touchscreen typing through automatic adaptation,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 815–824.

[46] S. Azenkot and S. Zhai, “Touch behavior with different postures on soft smartphone keyboards,” in *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, ser. MobileHCI ’12. New York, NY, USA: ACM, 2012, pp. 251–260.

[47] L.-P. Cheng, H.-S. Liang, C.-Y. Wu, and M. Y. Chen, “igrasp: grasp-based adaptive keyboard for mobile devices,” in *CHI ’13 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA ’13. New York, NY, USA: ACM, 2013, pp. 2791–2792.

[48] Y. Yin, T. Y. Ouyang, K. Partridge, and S. Zhai, “Making touchscreen keyboards adaptive to keys, hand postures, and individuals: a hierarchical spatial backoff model approach,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’13. New York, NY, USA: ACM, 2013, pp. 2775–2784.

[49] R. G. Stone, “Mobile touch interfaces for the elderly,” in *Proceedings of ICT, Society and Human Beings*. International Association for Development of the Information Society (IADIS), 2008.

[50] J. Goodman, G. Venolia, K. Steury, and C. Parker, “Language modeling for soft keyboards,” in *Proceedings of the 7th international conference on Intelligent user interfaces*, ser. IUI ’02. New York, NY, USA: ACM, 2002, pp. 194–195.

[51] I. S. MacKenzie and X. Zhang, “Eye typing using word and letter prediction and a fixation algorithm,” in *Proceedings of the 2008 symposium on Eye tracking research & applications*, ser. ETRA ’08. New York, NY, USA: ACM, 2008, pp. 55–58.

[52] K. Al Faraj, M. Mojahid, and N. Vigouroux, “Bigkey: A virtual keyboard for mobile devices,” in *Proceedings of the 13th International Conference on Human-Computer Interaction. Part III: Ubiquitous and Intelligent Interaction*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 3–10.

[53] L. Magnien, J. Bouraoui, and N. Vigouroux, “Mobile text input with soft keyboards: Optimization by means of visual clues,” in *Mobile Human-Computer Interaction - MobileHCI 2004*, ser. Lecture Notes in Computer Science, S. Brewster and M. Dunlop, Eds. Springer Berlin Heidelberg, 2004, vol. 3160, pp. 337–341.

[54] D. Lucas, “Disabled ’r’ all,” Master’s thesis, Instituto Superior Técnico, 2012.

[55] A. Gunawardana, T. Paek, and C. Meek, “Usability guided key-target resizing for soft keyboards,” in *Proceedings of the 15th international conference on Intelligent user interfaces*, ser. IUI ’10. New York, NY, USA: ACM, 2010, pp. 111–118.

[56] G. Lepicard and N. Vigouroux, “Touch screen user interfaces for older subjects: effect of the targets number and the two hands use,” in *Proceedings of the 12th international conference on Computers helping people with special needs*, ser. ICCHP’10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 592–599. [Online]. Available: http://dl.acm.org/citation.cfm?id=1880751.1880851

[57] K. Yatani, K. Partridge, M. Bern, and M. W. Newman, “Escape: a target selection technique using visually-cued gestures,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’08. New York, NY, USA: ACM, 2008, pp. 285–294. [Online]. Available: http://doi.acm.org/10.1145/1357054.1357104

[58] N. Garay-Vitoria and J. Abascal, “Text prediction systems: a survey,” *Univers. Access Inf. Soc.*, vol. 4, no. 3, pp. 188–203, Feb. 2006.

[59] N. Henze, E. Rukzio, and S. Boll, “Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 2659–2668.

[60] I. S. MacKenzie and R. W. Soukoreff, "A character-level error analysis technique for evaluating text entry methods," in *Proceedings of the second Nordic conference on Human-computer interaction*, ser. NordiCHI '02. New York, NY, USA: ACM, 2002, pp. 243–246. [Online]. Available: http://doi.acm.org/10.1145/572020.572056

[61] S. I. Mackenzie and W. R. Soukoreff, "Text entry for mobile computing: Models and methods,theory and practice," *Human-Computer Interaction*, vol. 17, 2002.

[62] P. Bain, J. Mally, M. Gresty, and L. Findley, "Assessing the impact of essential tremor on upper limb function," *Journal of Neurology*, vol. 241, no. 1, 1993. [Online]. Available: http://dx.doi.org/10.1007/BF00870673

[63] D. C. Hoaglin and B. Iglewicz, "Fine-tuning some resistant rules for outlier labeling," *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1147–1149, 1987.

[64] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '03, 2003, pp. 754–755.

A

Additional figures and tables

This appendix contains additional figures and tables of:

- Keyboard's Flowcharts.
- The baseline study.
- The study with the Elderly.

A.1 Keyboards' Flowcharts

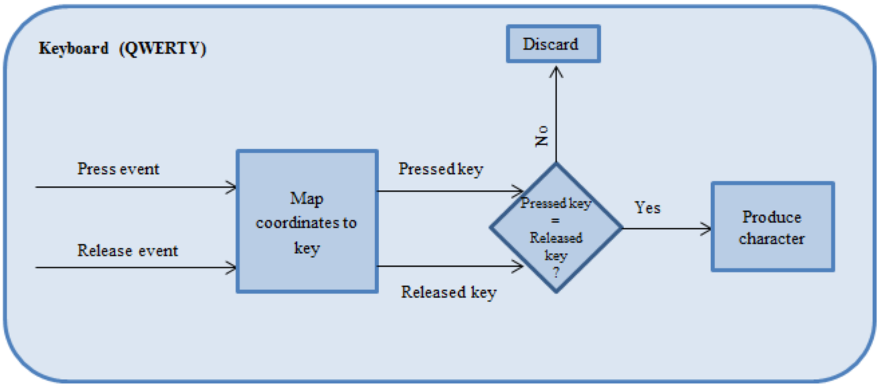


Figure A.1: Flow chart of the traditional QWERTY keyboard.

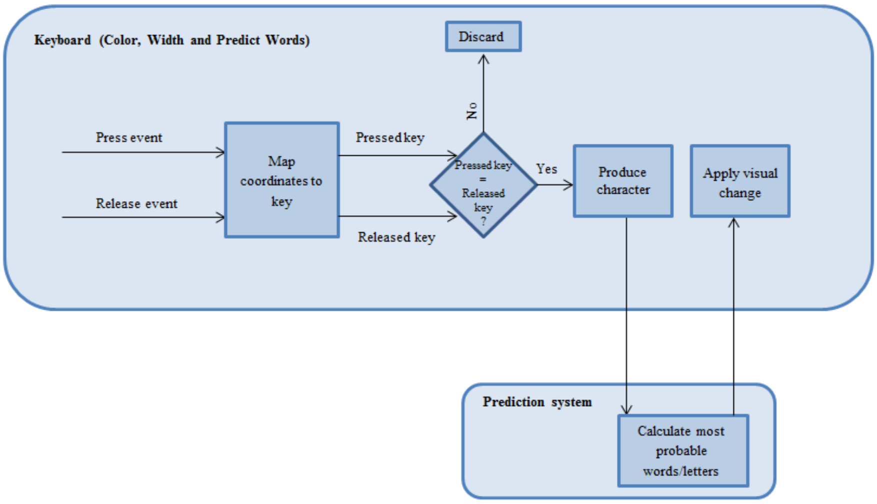


Figure A.2: Flow chart of the Color, Width and Predict Words variants.

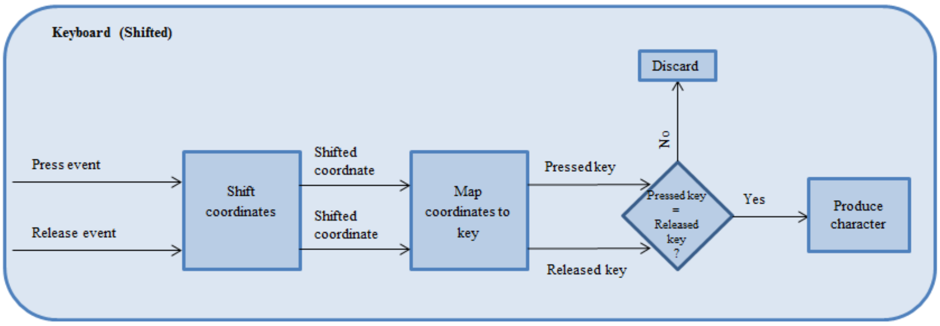


Figure A.3: Flow chart of the Shifted variant.

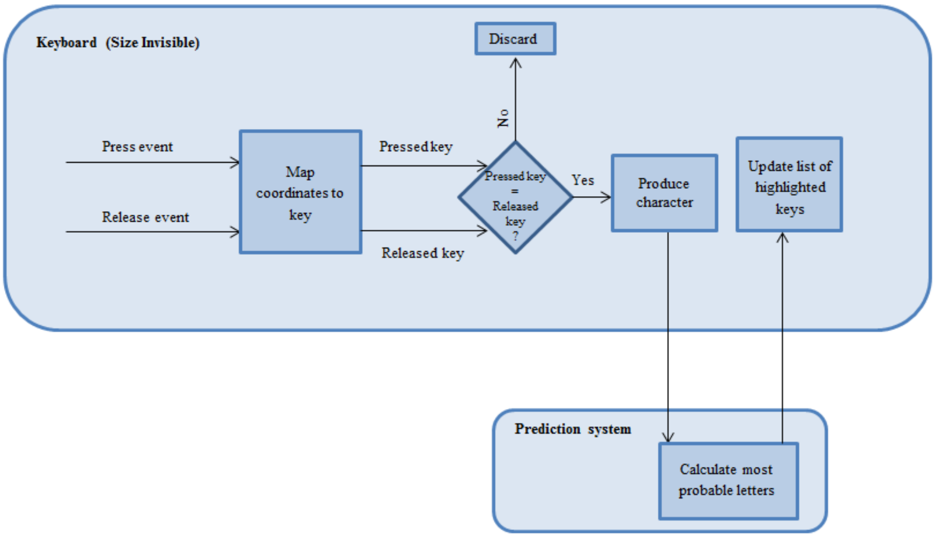


Figure A.4: Flow chart of the Size Invisible variant.

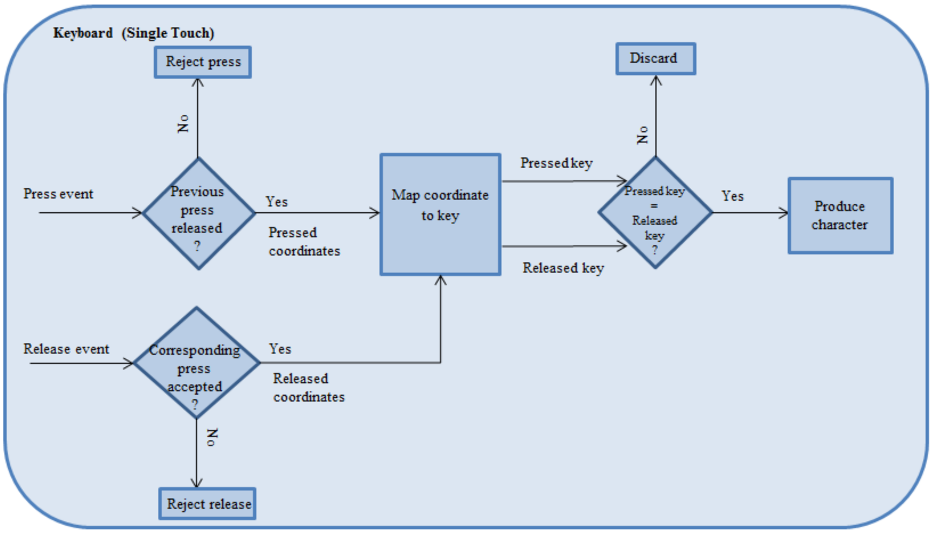


Figure A.5: Flow chart of the Single Touch variant.

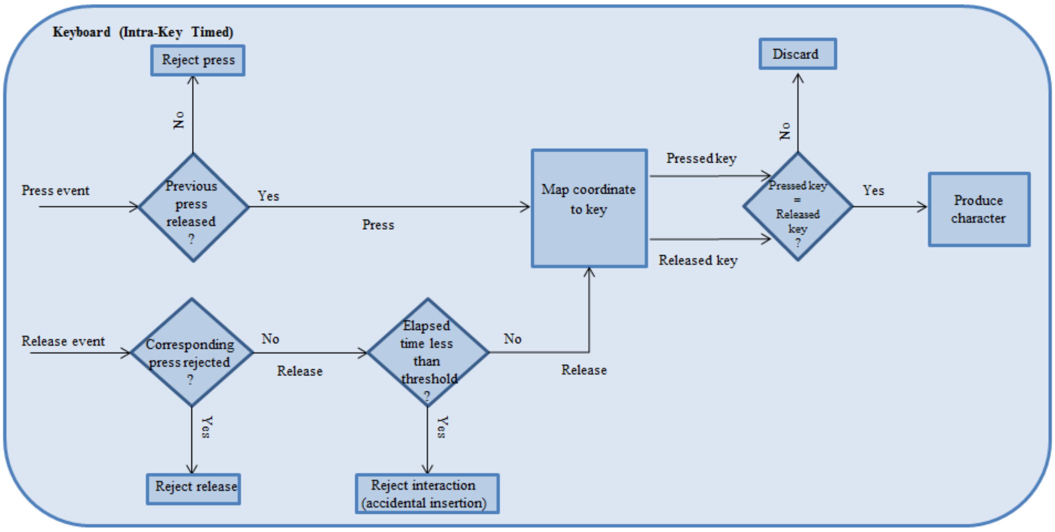


Figure A.6: Flow chart of the Intra-key Timed variant.

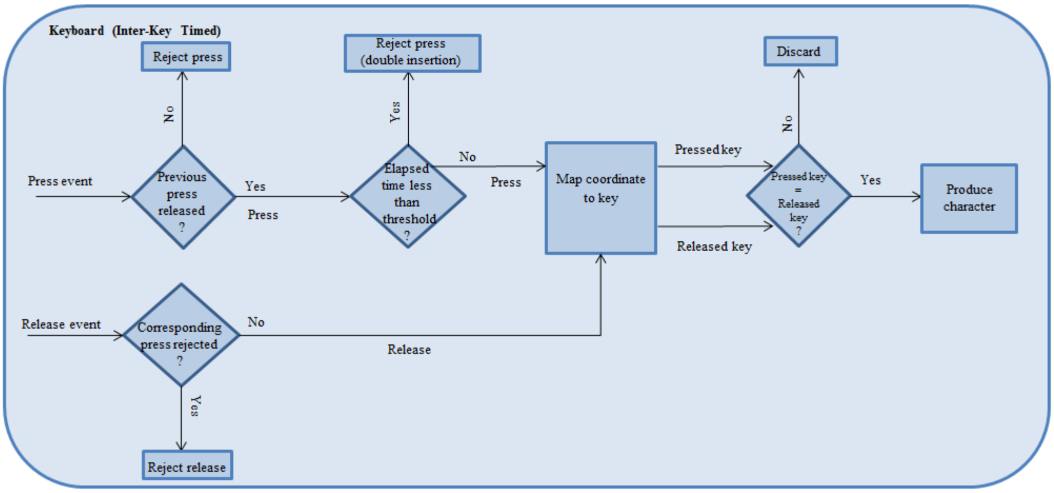


Figure A.7: Flow chart of the Inter-key Timed variant.

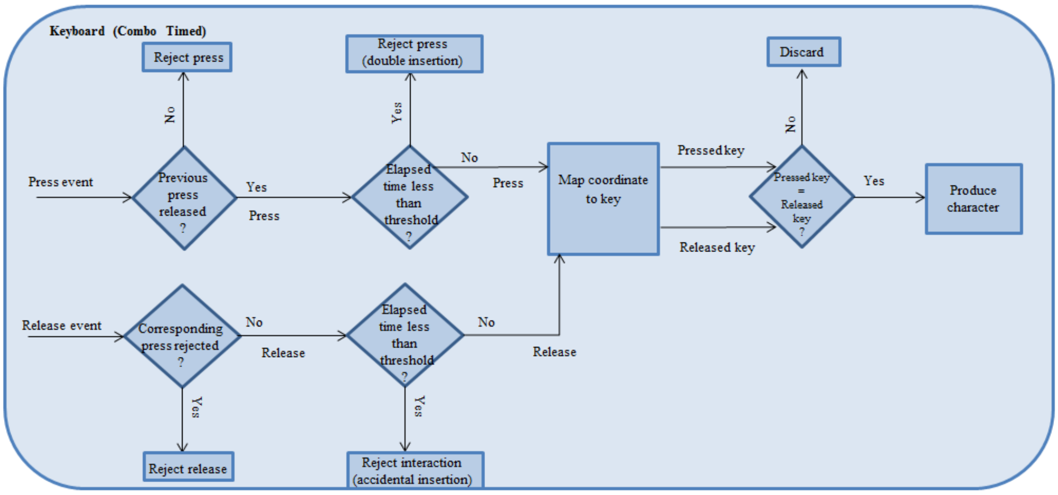
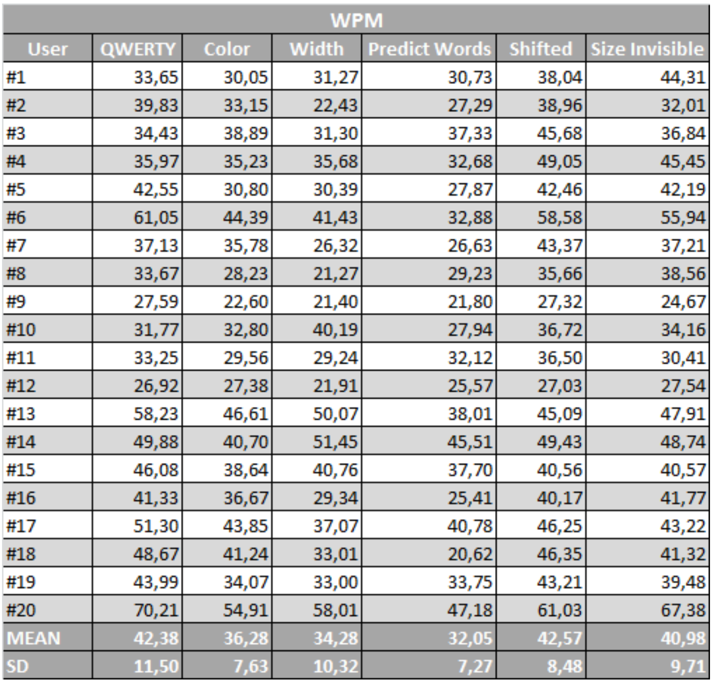


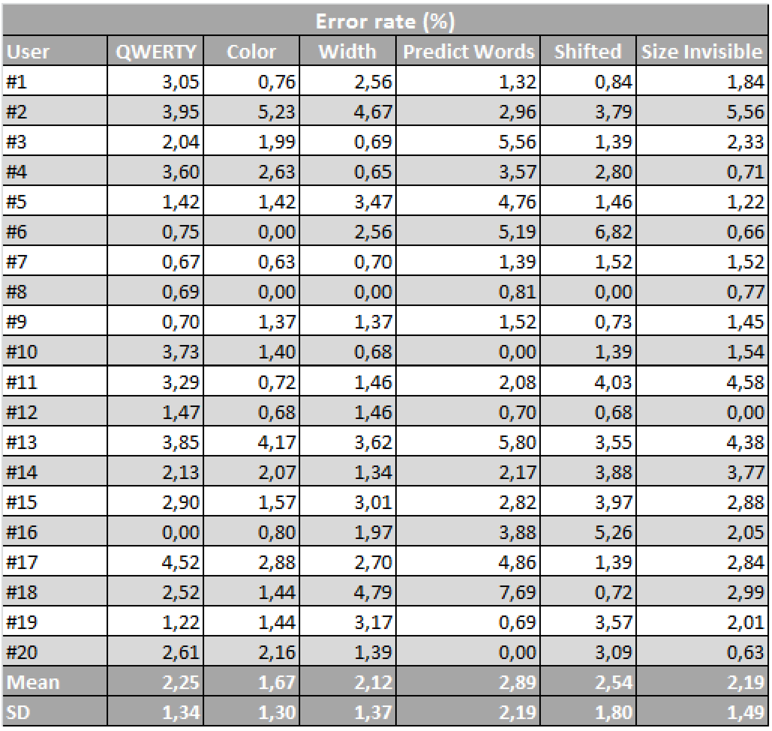
Figure A.8: Flow chart of the Combined Timed variant.

A.2 Baseline Study



WPM						
User	QWERTY	Color	Width	Predict Words	Shifted	Size Invisible
#1	33,65	30,05	31,27	30,73	38,04	44,31
#2	39,83	33,15	22,43	27,29	38,96	32,01
#3	34,43	38,89	31,30	37,33	45,68	36,84
#4	35,97	35,23	35,68	32,68	49,05	45,45
#5	42,55	30,80	30,39	27,87	42,46	42,19
#6	61,05	44,39	41,43	32,88	58,58	55,94
#7	37,13	35,78	26,32	26,63	43,37	37,21
#8	33,67	28,23	21,27	29,23	35,66	38,56
#9	27,59	22,60	21,40	21,80	27,32	24,67
#10	31,77	32,80	40,19	27,94	36,72	34,16
#11	33,25	29,56	29,24	32,12	36,50	30,41
#12	26,92	27,38	21,91	25,57	27,03	27,54
#13	58,23	46,61	50,07	38,01	45,09	47,91
#14	49,88	40,70	51,45	45,51	49,43	48,74
#15	46,08	38,64	40,76	37,70	40,56	40,57
#16	41,33	36,67	29,34	25,41	40,17	41,77
#17	51,30	43,85	37,07	40,78	46,25	43,22
#18	48,67	41,24	33,01	20,62	46,35	41,32
#19	43,99	34,07	33,00	33,75	43,21	39,48
#20	70,21	54,91	58,01	47,18	61,03	67,38
MEAN	42,38	36,28	34,28	32,05	42,57	40,98
SD	11,50	7,63	10,32	7,27	8,48	9,71

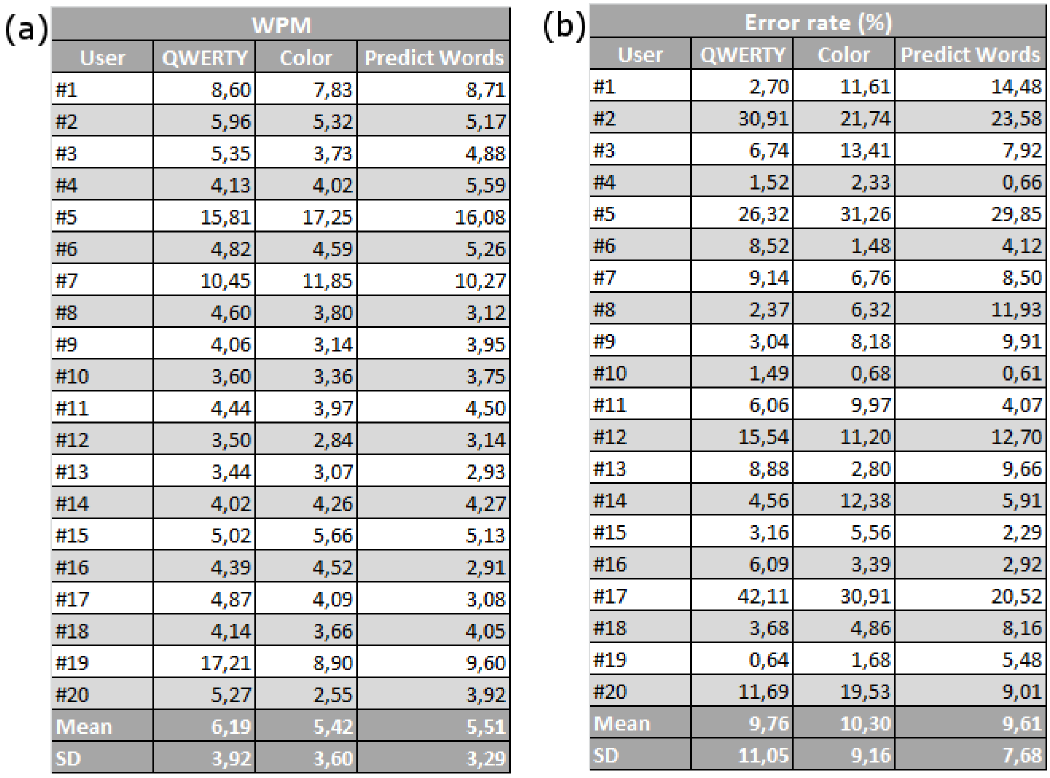
Table A.1: Words per minute for each participant in each keyboard condition.



Error rate (%)						
User	QWERTY	Color	Width	Predict Words	Shifted	Size Invisible
#1	3,05	0,76	2,56	1,32	0,84	1,84
#2	3,95	5,23	4,67	2,96	3,79	5,56
#3	2,04	1,99	0,69	5,56	1,39	2,33
#4	3,60	2,63	0,65	3,57	2,80	0,71
#5	1,42	1,42	3,47	4,76	1,46	1,22
#6	0,75	0,00	2,56	5,19	6,82	0,66
#7	0,67	0,63	0,70	1,39	1,52	1,52
#8	0,69	0,00	0,00	0,81	0,00	0,77
#9	0,70	1,37	1,37	1,52	0,73	1,45
#10	3,73	1,40	0,68	0,00	1,39	1,54
#11	3,29	0,72	1,46	2,08	4,03	4,58
#12	1,47	0,68	1,46	0,70	0,68	0,00
#13	3,85	4,17	3,62	5,80	3,55	4,38
#14	2,13	2,07	1,34	2,17	3,88	3,77
#15	2,90	1,57	3,01	2,82	3,97	2,88
#16	0,00	0,80	1,97	3,88	5,26	2,05
#17	4,52	2,88	2,70	4,86	1,39	2,84
#18	2,52	1,44	4,79	7,69	0,72	2,99
#19	1,22	1,44	3,17	0,69	3,57	2,01
#20	2,61	2,16	1,39	0,00	3,09	0,63
Mean	2,25	1,67	2,12	2,89	2,54	2,19
SD	1,34	1,30	1,37	2,19	1,80	1,49

Table A.2: Error rate for each participant in each keyboard condition.

A.3 Study With The Elderly



(a)

	WPM		
User	QWERTY	Color	Predict Words
#1	8,60	7,83	8,71
#2	5,96	5,32	5,17
#3	5,35	3,73	4,88
#4	4,13	4,02	5,59
#5	15,81	17,25	16,08
#6	4,82	4,59	5,26
#7	10,45	11,85	10,27
#8	4,60	3,80	3,12
#9	4,06	3,14	3,95
#10	3,60	3,36	3,75
#11	4,44	3,97	4,50
#12	3,50	2,84	3,14
#13	3,44	3,07	2,93
#14	4,02	4,26	4,27
#15	5,02	5,66	5,13
#16	4,39	4,52	2,91
#17	4,87	4,09	3,08
#18	4,14	3,66	4,05
#19	17,21	8,90	9,60
#20	5,27	2,55	3,92
Mean	6,19	5,42	5,51
SD	3,92	3,60	3,29

(b)

	Error rate (%)		
User	QWERTY	Color	Predict Words
#1	2,70	11,61	14,48
#2	30,91	21,74	23,58
#3	6,74	13,41	7,92
#4	1,52	2,33	0,66
#5	26,32	31,26	29,85
#6	8,52	1,48	4,12
#7	9,14	6,76	8,50
#8	2,37	6,32	11,93
#9	3,04	8,18	9,91
#10	1,49	0,68	0,61
#11	6,06	9,97	4,07
#12	15,54	11,20	12,70
#13	8,88	2,80	9,66
#14	4,56	12,38	5,91
#15	3,16	5,56	2,29
#16	6,09	3,39	2,92
#17	42,11	30,91	20,52
#18	3,68	4,86	8,16
#19	0,64	1,68	5,48
#20	11,69	19,53	9,01
Mean	9,76	10,30	9,61
SD	11,05	9,16	7,68

Table A.3: (a) Words per minute and (b) error rate for each participant in each keyboard condition.

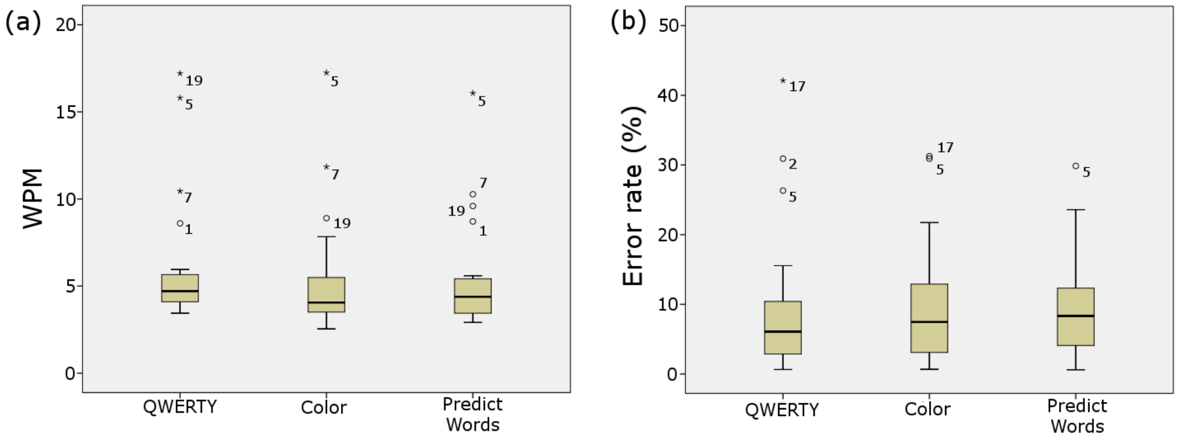
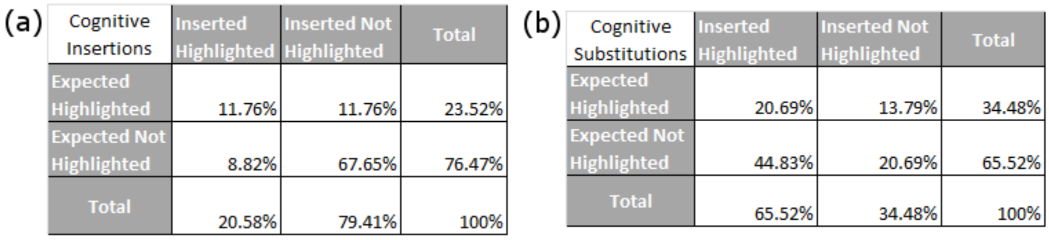


Figure A.9: Participants' (a) WPM and (b) error rate for each variant with outliers.



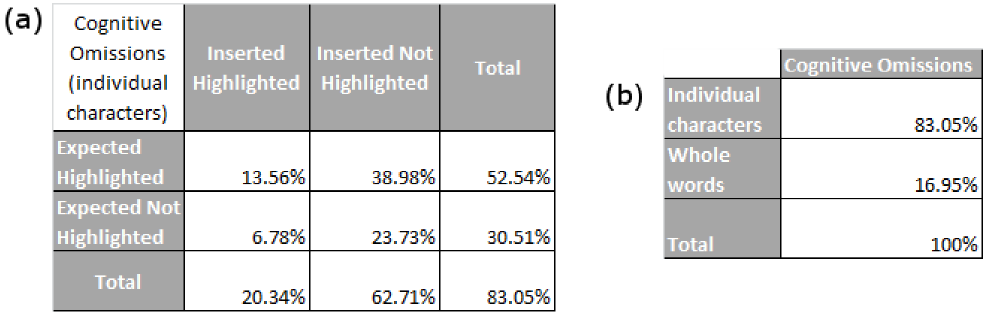
(a)

Cognitive Insertions	Inserted Highlighted	Inserted Not Highlighted	Total
Expected Highlighted	11.76%	11.76%	23.52%
Expected Not Highlighted	8.82%	67.65%	76.47%
Total	20.58%	79.41%	100%

(b)

Cognitive Substitutions	Inserted Highlighted	Inserted Not Highlighted	Total
Expected Highlighted	20.69%	13.79%	34.48%
Expected Not Highlighted	44.83%	20.69%	65.52%
Total	65.52%	34.48%	100%

Table A.4: (a) Cognitive insertions and (b) cognitive substitutions for the Color variant.



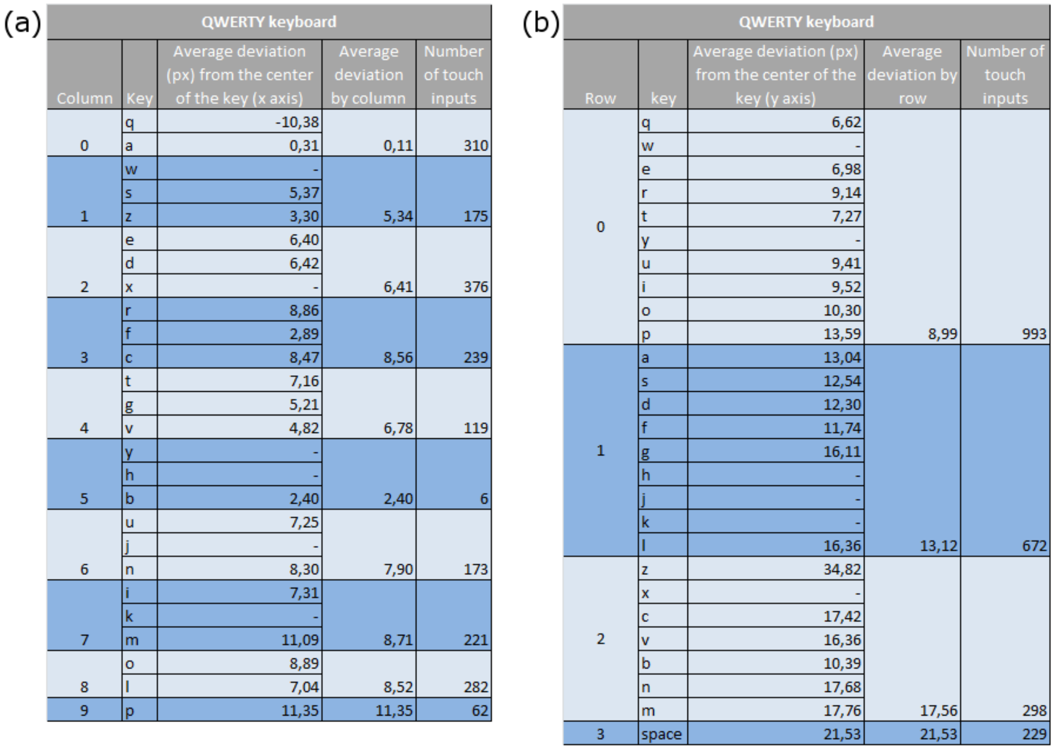
(a)

Cognitive Omissions (individual characters)	Inserted Highlighted	Inserted Not Highlighted	Total
Expected Highlighted	13.56%	38.98%	52.54%
Expected Not Highlighted	6.78%	23.73%	30.51%
Total	20.34%	62.71%	83.05%

(b)

	Cognitive Omissions
Individual characters	83.05%
Whole words	16.95%
Total	100%

Table A.5: Cognitive omissions for the Color variant.



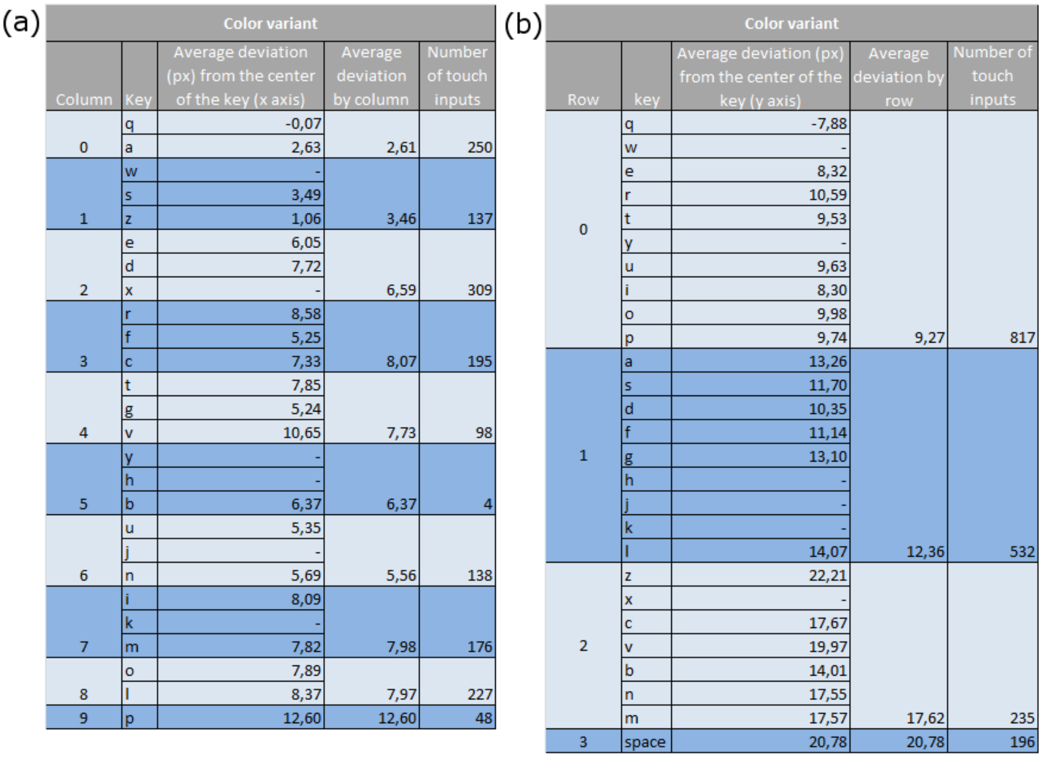
(a)

QWERTY keyboard				
Column	Key	Average deviation (px) from the center of the key (x axis)	Average deviation by column	Number of touch inputs
0	q	-10,38	0,11	310
	a	0,31		
1	w	-	5,34	175
	s	5,37		
	z	3,30		
2	e	6,40	6,41	376
	d	6,42		
	x	-		
3	r	8,86	8,56	239
	f	2,89		
	c	8,47		
4	t	7,16	6,78	119
	g	5,21		
	v	4,82		
5	y	-	2,40	6
	h	-		
	b	2,40		
6	u	7,25	7,90	173
	j	-		
	n	8,30		
7	i	7,31	8,71	221
	k	-		
	m	11,09		
8	o	8,89	8,52	282
	l	7,04		
9	p	11,35	11,35	62

(b)

QWERTY keyboard				
Row	key	Average deviation (px) from the center of the key (y axis)	Average deviation by row	Number of touch inputs
0	q	6,62	8,99	993
	w	-		
	e	6,98		
	r	9,14		
	t	7,27		
	y	-		
	u	9,41		
	i	9,52		
	o	10,30		
	p	13,59		
1	a	13,04	13,12	672
	s	12,54		
	d	12,30		
	f	11,74		
	g	16,11		
	h	-		
	j	-		
	k	-		
	l	16,36		
2	z	34,82	17,56	298
	x	-		
	c	17,42		
	v	16,36		
	b	10,39		
	n	17,68		
	m	17,76		
3	space	21,53	21,53	229

Table A.6: Average deviation of the touch points from the center of each key for the QWERTY condition for participants who only used their right hand. (a) shows the x-axis deviation (Column view), while (b) shows the y-axis deviation (Row view).



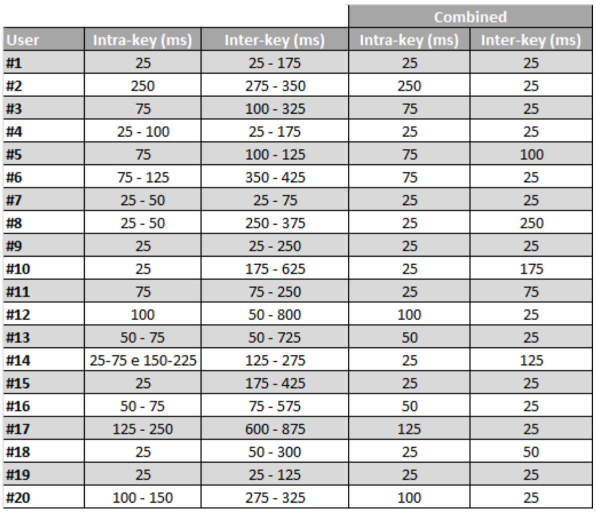
(a)

Color variant				
Column	Key	Average deviation (px) from the center of the key (x axis)	Average deviation by column	Number of touch inputs
0	q	-0,07	2,61	250
	a	2,63		
1	w	-	3,46	137
	s	3,49		
	z	1,06		
2	e	6,05	6,59	309
	d	7,72		
	x	-		
3	r	8,58	8,07	195
	f	5,25		
	c	7,33		
4	t	7,85	7,73	98
	g	5,24		
	v	10,65		
5	y	-	6,37	4
	h	-		
	b	6,37		
6	u	5,35	5,56	138
	j	-		
	n	5,69		
7	i	8,09	7,98	176
	k	-		
	m	7,82		
8	o	7,89	7,97	227
	l	8,37		
9	p	12,60	12,60	48

(b)

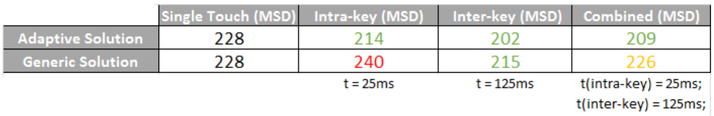
Color variant				
Row	key	Average deviation (px) from the center of the key (y axis)	Average deviation by row	Number of touch inputs
0	q	-7,88	9,27	817
	w	-		
	e	8,32		
	r	10,59		
	t	9,53		
	y	-		
	u	9,63		
	i	8,30		
	o	9,98		
	p	9,74		
1	a	13,26	12,36	532
	s	11,70		
	d	10,35		
	f	11,14		
	g	13,10		
	h	-		
	j	-		
	k	-		
	l	14,07		
2	z	22,21	17,62	235
	x	-		
	c	17,67		
	v	19,97		
	b	14,01		
	n	17,55		
	m	17,57		
3	space	20,78	20,78	196

Table A.7: Average deviation of the touch points from the center of each key for the Color condition for participants who only used their right hand. (a) shows the x-axis deviation (Column view), while (b) shows the y-axis deviation (Row view).



			Combined	
User	Intra-key (ms)	Inter-key (ms)	Intra-key (ms)	Inter-key (ms)
#1	25	25 - 175	25	25
#2	250	275 - 350	250	25
#3	75	100 - 325	75	25
#4	25 - 100	25 - 175	25	25
#5	75	100 - 125	75	100
#6	75 - 125	350 - 425	75	25
#7	25 - 50	25 - 75	25	25
#8	25 - 50	250 - 375	25	250
#9	25	25 - 250	25	25
#10	25	175 - 625	25	175
#11	75	75 - 250	25	75
#12	100	50 - 800	100	25
#13	50 - 75	50 - 725	50	25
#14	25-75 e 150-225	125 - 275	25	125
#15	25	175 - 425	25	25
#16	50 - 75	75 - 575	50	25
#17	125 - 250	600 - 875	125	25
#18	25	50 - 300	25	50
#19	25	25 - 125	25	25
#20	100 - 150	275 - 325	100	25

Table A.8: Optimum thresholds for each participant on each of the Timed variants (Intra-key, Inter-key and Combined).



	Single Touch (MSD)	Intra-key (MSD)	Inter-key (MSD)	Combined (MSD)
Adaptive Solution	228	214	202	209
Generic Solution	228	240	215	226
		t = 25ms	t = 125ms	t(intra-key) = 25ms; t(inter-key) = 125ms;

Table A.9: Minimum String Distance (MSD) for Intra-key, Inter-key and Combined Timed variants generic solution. The thresholds are fixed at the value indicated below the table.

B

Materials Used in the Studies

This appendix contains:

- Set of sentences used in text-entry evaluations.
- Evaluation monitor checklist.
- Evaluation script.
- Satisfaction questionnaire (that also include background questions).

B.1 List of Sentences

These sentences were taken from another study [14] that followed a similar approach to MacKenzie et al. [64] to select the sentences. The extracted sentences had an average of 5 words and 4.48 characters per word and were extracted from a Portuguese written language corpus. Moreover, each sentence had a minimum correlation with the language of 0.97. The following sentences were used in all text-entry evaluations presented in this dissertation:

- uma poderosa corrente de insatisfacao
- lugares de estacionamento mais procurados
- assembleia de cooperantes autorizou ainda
- decisoes a tomar na reuniao
- muito para alem do necessario
- a restauracao do estomago presidencial
- autor em casos de interoperacionalidade
- sera equipada com modernas tecnologias
- algo de muito necessario para
- da posicao anteriormente assumida pelo
- indeterminado para questoes a colocar
- consensual desde a primeira votacao
- de apoio a causa timorense
- uma dimensao de representacao social
- luciano patrao disse nao terem
- de incompreensao toldaria a sua
- um espaco tradicionalmente reservado aos
- reunioes da comissao parlamentar de
- dao entrada nas competicoes europeias
- de desorientacao e desanimo alastrou

- receitado alguns medicamentos para os
- da posicao anteriormente assumida pelo
- alimentares ou a doencas por
- relacionamento com as respectivas autoridades
- de apoio a causa timorense
- reuniao sectorial da comissao permanente
- comunicada as autoridades por elementos
- a impressao de desorientacao numa
- aumentando o risco de represalias
- varios aquartelamentos conforme a especialidade
- ocasionar o desaparecimento de algumas
- e do repoliciamento das ruas
- queda das compartimentacoes regionais pelo
- ou ainda em editoras escolares
- as interpelacoes da vereadora comunista
- castro almeida nao precisou de
- instalado entre as democracias europeias
- estava sobretudo relacionada com espionagem
- de respeitar a decisao autonoma
- duas prestacoes ao alegado intermediario
- da retoma das economias europeia
- seleccionador maturana e disse ao
- instrucao do processo teria alegadamente
- nao tem aparecido as reunioes
- material necessario ao desaparecimento dos
- se pode considerar eutanasia como
- producao de alimentos e materias
- dado ministerio ou apenas alteracoes
- seu lado considera importante a
- o seu metodo ainda precisara
- acudir aos problemas de estacionamento
- responder a uma solicitacao de
- americano aos produtos alimentares de
- o desespero da minoria culta
- mediante a instauracao de processo
- sua posicao de maior entre
- metropolitano das imediacoes serao encerradas
- cosmologia ou a eternidade necessaria
- ainda acentuados pelo desaparecimento misterioso
- decisao relativa ao prosseguimento da
- oposicao da inglaterra durante meses
- sobretudo pela apetencia dos americanos
- uma investigacao ordenada pelo secretario
- presidente da maior associacao levou
- reaccao aos insultos alegadamente proferidos

- de acelerar os pagamentos comunitarios
- espacial ate aos pormenores cuidadosamente
- agradou especialmente ao ministro das
- citados pela reuter nao escondiam
- aos investidores a regulamentacao do
- conversacoes eram a ultima oportunidade
- na comissao europeia terao de
- sera uma troca de opinioes
- decisoes a tomar na reuniao
- comissao instaladora da empresa entregou
- a europa de casais monteiro
- estalou a crise monetaria do
- lado esquerdo pertencia aos americanos
- os indicadores apresentarem uma evolucao
- as operacoes militares da onu
- comensalidade nos temos a prefiguracao
- pode conceder amnistia aos autores
- os centrais paulo madeira e
- acreditamos nele ou nao acreditamos
- denuncia sobre o alegado separatismo
- treinador escoces apostou ainda em
- resultado era de somenos importancia
- reuniao da comissao de arte
- eventual ostracismo na propria sociedade
- com anterioridade as nacoes europeias
- pecas fundamentais do relatorio e

Monitor Checklist

In this section we present the checklists used during the evaluation sessions.

Checklist 0 – The day before the evaluation

- Print the evaluation notes
- Print the task descriptions
- Print the checklists
- Perform one test

Checklist 1 – Day of the evaluation

Introducing the Session

- Greet the participant
- Give an overview of the evaluation session
- Explain the logging

Before the Evaluation

- Ask the user to stay in a comfortable position
- Start the application
- Undock the tablet from the base
- Be sure that there are no distractions
- Explain each of the variants and allow them to type two sentences per variant.

Before each Task

- Ask the user if he/she is comfortable
- Ask the user to adjust the comfort position
- Explain the task and ask the user to perform the required actions

During the Task

- Annotate questions, opinions, interruptions and other notes
- Try to keep the user relaxed

After each Task

- Ask if the user has any comment about the task he just did

After the evaluation

- Debrief the participant
- Thank the participant for his availability, opinions and help
- Close the application
- Add the user number generated from the logs to the questionnaire (e.g.: user11)
- Copy the log file to another folder

Checklist 2 – Day after the Evaluation

- Review the evaluation logs
- Insert the gathered data in raw tables

Checklist 3 – After all evaluation sessions

- Write the report

Evaluation Monitor Script

Antes de mais obrigado por aceitar participar neste trabalho. O meu nome é Élvio e sou aluno no Instituto Superior Técnico. O objectivo do meu trabalho é perceber como podemos melhorar os teclados virtuais nos Tablets, de forma a permitir que os idosos tenham mais facilidade em introduzir texto correctamente de forma mais rápida. Na próxima hora irá escrever algumas frases neste tablet, utilizando teclados virtuais QWERTY [explicar o que é um teclado QWERTY] com características diferentes.
De momento, desenvolvemos 2 teclados diferentes, que são variantes do teclado QWERTY normal. Queremos perceber quais destas variantes são as mais promissoras.

Esta sessão está dividida em três partes:

1. Existe uma primeira fase de familiarização, na qual terão a possibilidade de experimentar 3 teclados virtuais diferentes. Poderão duas frases por cada variante (o que perfaz um total de 6 frases na fase de familiarização).
2. De seguida, passaremos à fase de avaliação, na qual irão introduzir 5 frases por variante (total de 15 frases na fase de teste). No entanto, a primeira frase é sempre de treino. No fim de cada variante iremos perguntar-lhe se tem algum comentário a fazer sobre a variante testada.
3. Existe uma última fase onde irá responder a um pequeno questionário sobre o teste que acabou de efectuar.

Fase de Familiarização

Funcionamento da aplicação: Na parte superior da aplicação é-nos indicado a frase a inserir. O utilizador apenas tem de introduzir a frase utilizando o teclado virtual apresentado. Após finalizar a introdução da frase, tem de pressionar o botão “Próxima frase” para passar para a próxima frase . Após esgotar as 6 frases disponíveis para a fase de familiarização (2 frases por variante), o utilizador poderá pressionar no botão “Iniciar Experiência”, para dar início à experiência. Como está na fase de treino, poderá escolher qual a variante do teclado que quer usar [mostrar como escolher a variante].

Durante a fase de treino, e mesmo durante a fase de avaliação, não poderá corrigir o que escreve, ou seja, caso se engane, deve passar para a letra seguinte sem se preocupar em apagar.

QWERTY: Este teclado é uma representação virtual do conhecido teclado QWERTY, que dispõe as letras da seguinte forma (mostrar teclado QWERTY). [Permitir que o utilizador escreva duas frases com este teclado]

Variação de cor: Este teclado é muito semelhante ao teclado anterior, com a diferença que tenta prever a próxima letra que o utilizador pretende inserir. Desta forma, o teclado irá iluminar uma, duas, três ou no máximo quatro teclas, que são as letras mais prováveis. No entanto, o utilizador continua a ter liberdade para escolher outra letra (até porque a predição pode não estar correcta). [Permitir que o utilizador escreva duas frases com esta variante]

Com sugestão de palavra: Este teclado é em tudo semelhante ao QWERTY, com a diferença que irá sugerir uma, duas, três ou no máximo quatro palavras, que são as palavras mais prováveis. [Permitir que o utilizador escreva duas frases com esta variante]

Fase de Avaliação

Na fase de avaliação, irão fazer exactamente o que fizeram na fase de familiarização (introduzir frases com diferentes teclados). A única diferença é que os teclados serão escolhidos de forma aleatória. O tempo só começa a contar quando o utilizador começar a introduzir o texto (primeira letra), o que significa que o utilizador pode ler a frase calmamente. Após a introdução da frase, pode pressionar no botão “Próxima frase” para passar à próxima frase.

O que estamos a avaliar é o método de introdução de texto e como este se adequa aos utilizadores e nunca a pessoa em questão.

Volto a relembrar que nesta tarefa não poderá corrigir o que escreve, ou seja, caso se engane deve passar para a letra seguinte sem se preocupar em apagar.

Após cada variante, recolher comentários:

USER: ___

QWERTY (1):

Variação de cor (2):

Variação de largura (3):

Predição de palavras (4):

Centro de massa (5):

Variação tamanho invisível (6):

Questionário de Satisfação

Obrigado por ter participado na avaliação do nosso protótipo de teclados QWERTY virtuais.

Pedimos agora que responda ao seguinte questionário, que não demora mais que 10 minutos a completar. As respostas serão guardadas anonimamente e não serão fornecidas a terceiros.

* Required

Perfil do utilizador

---

1. **Sexo?** *

   Mark only one oval.

   ○ Masculino
   ○ Feminino

2. **Qual a sua idade?** *

   Mark only one oval.

   ○ <= 18 anos
   ○ 19 - 30 anos
   ○ 31 - 40 anos
   ○ 41 - 50 anos
   ○ 51 - 60 anos
   ○ 61 - 70 anos
   ○ 71 - 80 anos
   ○ 81 - 90 anos
   ○ >= 91 anos

3. **Habilitações literárias (concluído)?** *

   Mark only one oval.

   ○ Básico
   ○ Secundário
   ○ Licenciatura
   ○ Mestrado
   ○ Doutoramento

4. **Mão dominante?** *

Mark only one oval.

○ Direita
○ Esquerda

5. **Tem algum problema de visão?** *

Mark only one oval.

○ Sim
○ Não

6. **Comentário**

....
....
....
....
....

7. **Sofre de algum tipo de tremor?** *

Mark only one oval.

○ Sim
○ Não

8. **Comentário**

....
....
....
....
....

Experiência com teclados QWERTY

9. **Costuma utilizar máquina de escrever?** *

Mark only one oval.

- ○ Sim, diariamente.
- ○ Sim, semanalmente.
- ○ Sim, mas apenas raramente.
- ○ Usava no passado, diariamente.
- ○ Usava no passado, semanalmente.
- ○ Usava no passado, raramente.
- ○ Não, nunca utilizei.

10. **Há quanto tempo utiliza?**

Mark only one oval.

- ○ Menos de 1 mês
- ○ Entre 1 mês e 6 meses
- ○ Entre 6 meses e 1 ano
- ○ Há mais de 1 ano

11. **Costuma utilizar teclados QWERTY físicos (ex.: computador/portátil)?** *

Mark only one oval.

- ○ Sim, diariamente.
- ○ Sim, semanalmente.
- ○ Sim, mas apenas raramente.
- ○ Não, nunca tinha utilizei.

12. **Há quanto tempo utiliza?**

Mark only one oval.

- ○ Menos de 1 mês
- ○ Entre 1 mês e 6 meses
- ○ Entre 6 meses e 1 ano
- ○ Há mais de 1 ano

13. **Costuma utilizar teclados QWERTY Virtuais (ex.: no telemóvel ou tablet)?** *

Mark only one oval.

- ○ Sim, diariamente.
- ○ Sim, semanalmente.
- ○ Sim, mas apenas raramente.
- ○ Não, nunca tinha utilizado. *After the last question in this section, skip to question 27.*

14. **Há quanto tempo utiliza?**
Mark only one oval.

- ○ Menos de 1 mês
- ○ Entre 1 mês e 6 meses
- ○ Entre 6 meses e 1 ano
- ○ Há mais de 1 ano

Experiência de utilização com teclados virtuais QWERTY em TABLETS

Tenha em conta que, nesta secção, todas as perguntas se referem à utilização de teclados virtuais QWERTY em TABLETS.

Se não utiliza teclados virtuais em tablets, passe para a secção seguinte.

15. **Com que frequência utiliza?**
Mark only one oval.

- ○ Diariamente.
- ○ Semanalmente.
- ○ Raramente.

16. **Há quanto tempo utiliza?**
Mark only one oval.

- ○ Menos de 1 mês
- ○ Entre 1 mês e 6 meses
- ○ Entre 6 meses e 1 ano
- ○ Há mais de 1 ano

17. **Como considera a sua perícia?**
Mark only one oval.

	1	2	3	4	5	
Muito má	○	○	○	○	○	Muito boa

18. **Em que tarefas utiliza?**
Check all that apply.

- ☐ SMS.
- ☐ Emails.
- ☐ Redes sociais.
- ☐ Tirar notas.
- ☐ Other: ....

19. **Costuma utilizar predição de palavras?**
Para auto-completar a palavra com base numa sugestão do sistema.
Mark only one oval.

- ○ Sim.
- ○ Não.
- ○ Não, porque o meu dispositivo não permite.

20. **Costuma utilizar correcção de palavras?**
Correcção de erros automática, depois de escrita a palavra.
Mark only one oval.

- ○ Sim.
- ○ Não.
- ○ Não, porque o meu dispositivo não permite.

Experiência de utilização com teclados virtuais QWERTY em TELEMÓVEIS

Tenha em conta que, nesta secção, todas as perguntas se referem à utilização de teclados virtuais QWERTY em TELEMÓVEIS.

Se não utiliza teclados virtuais em telemóveis, passe para a página seguinte.

21. **Com que frequência utiliza?**
Mark only one oval.

- ○ Diariamente.
- ○ Semanalmente.
- ○ Raramente.

22. **Há quanto tempo utiliza?**
Mark only one oval.

- ○ Menos de 1 mês
- ○ Entre 1 mês e 6 meses
- ○ Entre 6 meses e 1 ano
- ○ Há mais de 1 ano

23. **Como considera a sua perícia?**
Mark only one oval.

	1	2	3	4	5	
Muito má	○	○	○	○	○	Muito boa

24. **Em que tarefas utiliza?**
Check all that apply.

- ☐ SMS.
- ☐ Emails.
- ☐ Redes sociais.
- ☐ Tirar notas.
- ☐ Other: ....

25. **Costuma utilizar predição de palavras?**
Para auto-completar a palavra com base numa sugestão do sistema.
Mark only one oval.

- ○ Sim.
- ○ Não.
- ○ Não, porque o meu dispositivo não permite.

26. **Costuma utilizar correcção de palavras?**
Correcção de erros automática, depois de escrita a palavra.
Mark only one oval.

- ○ Sim.
- ○ Não.
- ○ Não, porque o meu dispositivo não permite.

Satisfação de utilização

27. **Como introduziu o texto? ***
Mark only one oval.

- ○ Com o tablet apoiado na mão, utilizando a mão livre para escrever.
- ○ Com o tablet apoiado na mesa, utilizando uma mão para escrever.
- ○ Com o tablet apoiado na mesa, utilizando ambas as mãos para escrever.
- ○ Other: ....

28. **Classifique cada teclado relativamente à sua satisfação na utilização. ***
Mark only one oval per row.

	1 - Muito insatisfeito	2 - Insatisfeito	3 - Neutro	4 - Satisfeito	5 - Muito satisfeito
QWERTY	○	○	○	○	○
Variação de cor	○	○	○	○	○
Com sugestão de palavra	○	○	○	○	○

29. **Classifique cada teclado consoante a facilidade de utilização.**
Mark only one oval per row.

	1 - Muito difícil	2 - Difícil	3 - Normal	4 - Fácil	5 - Muito fácil
QWERTY	○	○	○	○	○
Variação de cor	○	○	○	○	○
Com sugestão de palavra	○	○	○	○	○

30. **Classifique cada teclado consoante a utilidade de cada variante, em comparação com o QWERTY.**
Mark only one oval per row.

	1 - Inútil	2 - Pouco útil	3 - Semelhante	4 - Útil	5 - Muito útil
Variação de cor	○	○	○	○	○
Com sugestão de palavra	○	○	○	○	○

31. **Classifique cada teclado relativamente ao esforço cognitivo necessário para o utilizar.**
Mark only one oval per row.

	1 - Muito elevado	2 - Elevado	3 - Normal	4 - Baixo	5 - Muito baixo
QWERTY	○	○	○	○	○
Variação de cor	○	○	○	○	○
Com sugestão de palavra	○	○	○	○	○

32. **Classifique cada teclado relativamente à facilidade em localizar a letra pretendida.**
Mark only one oval per row.

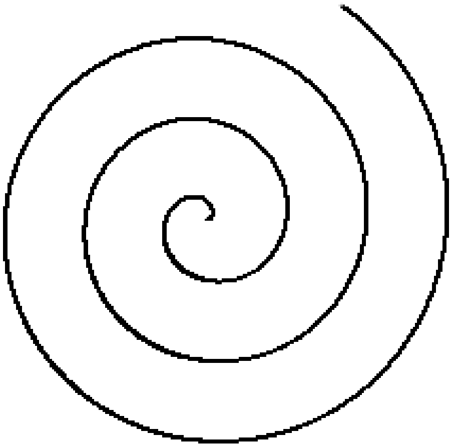
	1 - Muito difícil	2 - Difícil	3 - Normal	4 - Fácil	5 - Muito fácil
QWERTY	○	○	○	○	○
Variação de cor	○	○	○	○	○

33. **Considera que a palavra que quer escrever se encontra, na maioria das vezes, na lista de palavras sugeridas?**
Mark only one oval.

	1	2	3	4	5	
Discordo completamente	○	○	○	○	○	Concordo completamente

Desenhe uma espiral sem apoiar a mão ou o braço na mesa.

Mão direita:



Mão Esquerda: