Institut für Informatik
der Technischen Universität
München

Two-Handed User Interfaces for Mobile Devices

Tayfur Coskun

Institut für Informatik
der Technischen Universität
München

Fachgebiet Augmented Reality

Two-Handed User Interfaces for Mobile Devices

Tayfur Coskun

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.- Prof. Dr. R. Westermann

Prüfer der Dissertation:

1. Univ.- Prof. G.J. Klinker, Ph.D.
2. Univ.- Prof. Dr. A. Schmidt, Universität Stuttgart

Die Dissertation wurde am 31.03.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 01.09.2014 angenommen.

Danksagung

Ich möchte an dieser Stelle als erstes meiner Familie und meinen Eltern danken, ohne deren finanzielle sowie persönliche und moralische Unterstützung weder mein Informatik Studium noch meine Promotion so möglich gewesen wäre. Völlig gleich in welcher Lebenslage ich mich befand, mit dem Verständniss und der Rückendeckung meiner Eltern konnte ich immer zu 100% rechnen, was keinesfalls selbstverständlich ist. Ebenso möchte ich meinem besten Freund Yavuz Aytekin und meiner Freundin Claudia Grill danken, deren moralische Unterstützung mir ebenfalls stets halfen, die richtigen Entscheidungen zu treffen.

Mein weiterer Dank gilt Prof. Gudrun Klinker. Sie hatte immer ein Ohr für jeden Ihrer Mitarbeiter und unterstützte jeden Einzelnen in seinen/ihren Ideen. Dabei nahm sie sich stets Zeit für mich und meine Kollegen und hatte einen außergewöhnlich positiven und vertrauensvollen Umgang mit ihrem Team. Es war nicht zuletzt ihr zu Verdanken, dass meine Promotion nicht nur sehr erfolgreich verlief, sondern mich zusätzlich mit Freude und Stolz erfüllt hat. Sie war für mich stets ein Vorbild und wird für mich immer ein Vorbild bleiben, und das nicht nur in fachlicher Hinsicht. Nicht minder möchte ich mich bei Prof. Albrecht Schmidt, meinem Zweitgutachter, bedanken, der mir half noch ein mal aus einer anderen Perspektive auf meine Arbeiten zu blicken und mich hinsichtlich der Dissertation hervorragend beriet und betreute.

Des weiteren möchte ich meinen Kollegen im Fachgebiet für Augmented Reality (FAR) danken. In diesem Team fühlte ich mich immer stets sehr wohl. Mit der Zeit wurden meine Kollegen gute und enge Freunde. Ganz besonders möchte ich mich bei Amal Benzina bedanken, mit der ich mich jeder Zeit fachlich austauschen konnte und deren fachliche Meinung für mich immer eine entscheidende Grundlage in meinen wissenschaftlichen Arbeiten bildete. So konnten wir auch zusammen zahlreiche Publikationen gemeinsam veröffentlichen. Darüberhinaus wusste ich auch immer ihren persönlichen und freundschaftlichen Rat zu schätzen. Des weiteren möchte ich mich bei Eva Artinger bedanken, die zusammen mit mir mit dem Projekt SpeedUp an der Gruppe FAR startete. Wir unterstützten und stets gegenseitig und arbeiteten uns gemeinsam in die Materie des Rettungsdienstes und den Ablauf eines MANV ein. Außerdem möchte ich auch noch meinem Kollegen und Freund Manuel Huber besonders danken, da ich mit seiner kompetenten Meinung rechnen durfte und konnte. Seine hilfsbereite und freundliche Art half mir, meinen persönlichen Platz in der Gruppe und der wissenschaftlichen Community zu finden. Des weiteren möchte ich mich noch besonders bei Patrick Maier bedanken, mit dem ich mich fachlich austauschen konnte und der jederzeit bereit war, seine konstruktive Meinung zu den entwickelten Kozepten und den ausgearbeiteten Studien zu geben und auch immer bereit war, sich an verschiedensten Arbeiten und Studien tatkräftig und uneigennützig zu beteiligen.

Nicht zuletzt möchte ich mich bei meinen zahlreichen Studenten bedanken, die mit ihren Bachelor-, Master- und Diplomarbeiten entscheidend zum Erfolg der einzelnen Publikationen und damit auch für den Erfolg dieser Dissertation beitrugen. Dabei hat mich besonders der persönliche Einsatz dieser Studenten beeindruckt. So war nahezu jeder meiner Studenten bereit, über die Benotungsinfrastruktur hinaus, an einer Publikation mit zu wirken. Einige übernahmen hierbei sogar mit Freuden die Hauptautoren Position und besuchten dann auch die entsprechenden Konferenzen/Workshops ihrer Publikationen, was ich mit besonderem Stolz und Freude begleitete und weiterhin betreute. Ohne ihren Einsatz hätten viele der Studien wie sie in dieser Dissertation beschrieben sind, nicht in der Qualität und schon gar nicht in der Quantität realisierbar sein können. Deshalb gilt mein besonderer Dank auch an die folgenden Studenten: Emal Sadran, Phillip Commans, Clemens Binder, Uwe Trottmann, Mathias Gorf, Clara Lange, Thomas Behrens, Md. Raihanul Islam, Christian Wiesner, Carmen Rudolph, Natalia Zarawska, Gel Han, Johannes Roith, Thomas Faltermaier, Lorenzo Piritano, Daniela Korhammer, Kang-Hunn Lee, Theodora Velikova, Sebastian Klingenbeck, Christoph Bruns, Phillip Schmitt, Andreas Schmidt und Andreas Balzer.

Vielen Dank!

Zusammenfassung

Die schnelle und professionelle Reaktion von Rettungskräften in Großschadensereignissen ist essentiel für das Überleben der Beteiligten. Übersteigt die Anzahl Verletzter die dem Rettungsdienst zur Verfügung stehenden Ressourcen, so spricht man von einem Massenanfall von Verletzten (MAnV). In einer solchen Situation ist jede Sekunde entscheidend. Die Rettungskräfte suchen in der ersten Phase verletzte Personen im betroffenen Gebiet und bestimmen die Priorität der Patienten mit Hilfe eines Algorithmus, der Triage genannt wird. Dieser Prozess wird bis heute im Wesentlichen ohne die Unterstützung von IT durchgeführt. Durch die im Projekt SpeedUp vorgeschlagene Integration von IT-Lösungen in diesen Prozess können wichtige Informationen schneller erfasst und Kollegen sowie Vorgesetzten umgehend zur Verfügung gestellt werden. Die so gesammelten Informationen bieten dem Organisatorischen Leiter im Rettungsdienst (OrgL) einen besseren und vor allem schnelleren Überblick über die aktuelle Lage in Echtzeit.

Um auf diese Informationen auch mobil einfach und sicher zugreifen zu können, bietet sich die Nutzung von Tablets an. Aufgrund der unter Umständen unvorhersagbaren, unsicheren und gefährlichen Umgebung eignen sich herkömmliche Tablets aber nur bedingt. Robuste und ausfallsichere Tablets sind in der Regel jedoch deutlich schwerer.

Aus der Erfahrung realer Anwendern (der Feuerwehr TUM) entstand unter Anderem die Anforderung, die Benutzerschnittstelle der SpeedUp Anwendung für robuste mobile Geräte so zu entwickeln, dass das Tablet zu jeder Zeit in beiden Händen gehalten werden und mit den Daumen bedient werden kann. Auf diese Weise wird die Last des schweren Tablets auf beide Arme verteilt und eine komfortable und ergonomischere Benutzung ermöglicht.

Unter Berücksichtigung dieser Anforderungen wurden verschiedene Benutzerschnittstellen in den Kategorien der Interaktion mit digitalen Karten und der mobilen sowie stationären Texteingabe untersucht. Dabei ist die enge Einbindung der Zielgruppe in die Entwicklung von neuen Benutzerschnittstellen essentiell, um zum Einen zu gewährleisten, dass die Endnutzer das neue Werkzeug akzeptieren und nutzen und zum Anderen, um sicherzustellen, dass es den Rettungsdienstablauf tatsächlich unterstützt und nicht wie im schlimmsten Fall möglich sogar behindert. Aus diesem Grund wurden alle dargestellten Benutzerschnittstellen in einem nutzer-zentrierten iterativen Prozess entwickelt. Am Ende jeder Iteration werden die entwickelten Lösungen zusammen mit Anwendern der Feuerwehr TUM, dem Arbeiter-Samariter-Bund München und dem Rettungsdienst Stralsund auf verschiedene Nutzerparameter evaluiert und deren weitere Entwicklung diskutiert. Diese Lösungen und die Ergebnisse der Benutzerevaluationen jeder Iteration werden in der vorliegenden Arbeit dargestellt und kritisch diskutiert. Die aktuelle Version dieser Interaktionslemente wird jeweils vorgestellt und ein Ausblick über mögliche Weiterentwicklungen gegeben.

Abstract

In a disaster, the rapid response of the rescue service is essential for saving lives. If there are more injured people than rescue sources are available, this is called a mass casualty incident (MCI). Each second counts in such a situation. During the first phase the rescue service performs an algorithm (the triage) to determine the severity of the injuries. This procedure is usually done without the help of IT solutions. The project SpeedUp suggests enhancing the current rescue procedure with IT solutions to collect and distribute important MCI related information faster and more accurately. A better and faster overview of the current situation can then be presented to the rescue service organizational commander in real time.

To enable the commander to access this data even when being mobile, it makes sense to use a tablet device. But because of the unpredictable, uncertain and dangerous situation, standard tablets may not be the best choice. Ruggedized tablets fit the situation better, but they are also much heavier.

Based on the experience of the real target group (Fire department TUM), a requirement was formulated that the user interface (UI) of the SpeedUp application must be developed in a way that it can be used all the times with the thumbs only while holding the tablet in both hands. This way, the load of the tablet's weight is shared among both arms, thereby enabling a more comfortable and more ergonomic way to interact with the device.

Based on that requirement different user interfaces have been developed and studied in the context of digital map interaction and both mobile and stationary text input. It is essential to develop the solutions close to the target group to ensure the acceptance of such new tools by the target group and to find out whether the specific solution really improves the current rescue procedure or not. For this reason, all presented solutions have been developed in an iterative, user-centric procedure. At the end of each iteration cycle a user evaluation has been conducted with the Firedepartment TUM, the Arbeiter-Samariter-Bund München or the Rescue Service Stralsund and the results have been discussed with the target group. The developed solutions, as well as the results for each evaluation are presented and critically discussed in this thesis. The most current version of each UI element is also presented and an overview of possible future iterations is given.

Contents

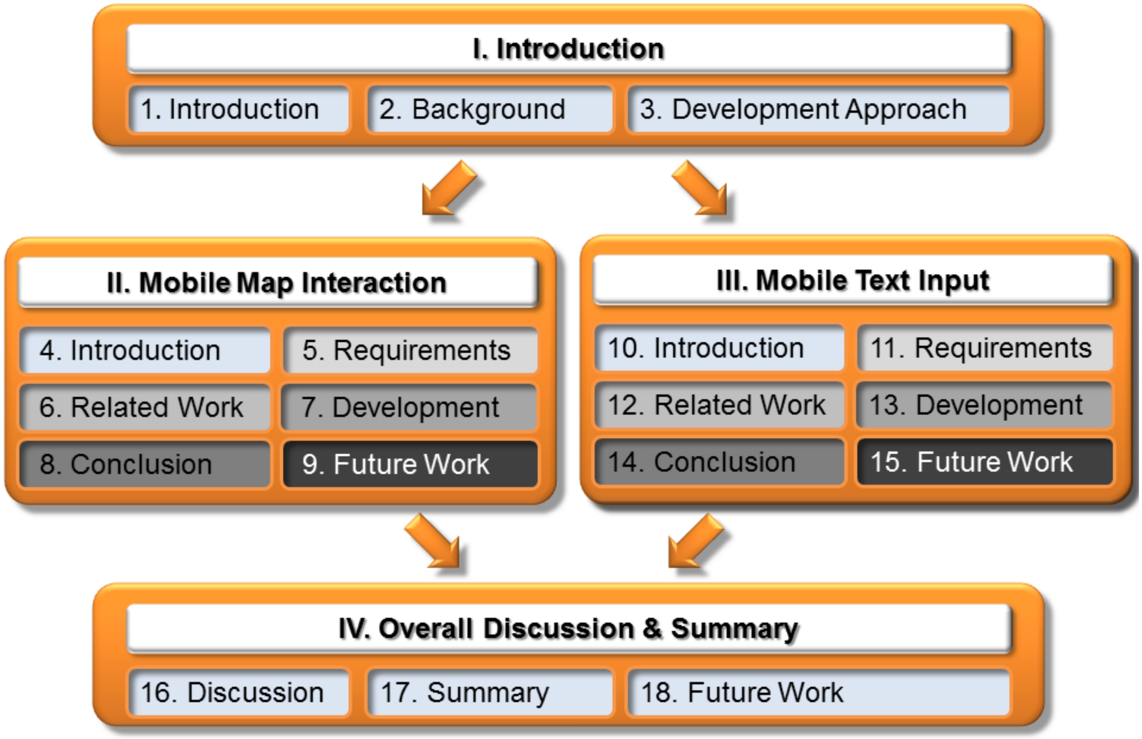
List of Figures

List of Tables

Outline of the Thesis

Thesis Overview

This thesis is organized in four parts: I) Introduction, II) Mobile Map-Interaction, III) Mobile Text Input, and IV) Overall Conclusion and Discussion. The following picture gives on overview of the single parts and chapters. A brief description of the content of each chapter is then given in the following sections.



Part I: Introduction

CHAPTER 1: INTRODUCTION

Chapter 1 introduces the definition of a Mass Casualty Incident (MCI) and describes the relevant parts of its procedure. It further describes the Project SpeedUp and clarifies the role of the Technische Universität München in that project. Last but not least, the role as well as the goal of this dissertation is described.

CHAPTER 2: BACKGROUND Chapter 2 describes the context as well as the fundamentals of this dissertation. The project SpeedUp and the role of TUM are explained in more detail. Additionally, important basics of usability and human-computer-interaction in general are given. Afterwards, some important and relevant aspects of touchscreens are presented.

CHAPTER 3: DEVELOPMENT APPROACH Chapter 3 introduces the reader into the theory of the iterative and user-centered development approach which has been used during the development procedure for all interface metaphors presented in this dissertation.

Part II: Mobile Map Interaction on a Ruggedized Tablets Bezel

CHAPTER 4: INTRODUCTION Chapter 4 introduces the first topic of this dissertation which is to develop mobile map interaction metaphors and concepts for heavy ruggedized tablet devices designed to be used while held in both hands and in the context of an MCI.

CHAPTER 5: REQUIREMENTS Chapter 5 describes the special requirements derived from the unstable MCI context and the heavy ruggedized hardware. Those requirements are collected from both sources: a) The iterative interviews with people from the target group and b) the literature.

CHAPTER 6: RELATED WORK Chapter 6 gives an overview of the related work in the context of digital map interaction and two-handed tablet device interaction.

CHAPTER 7: DEVELOPMENT Chapter 7 describes the concrete development procedure of the map interaction metaphors. This includes all three development iterations. Each iteration consists of the description of each introduced UI concept, the evaluation, as well was its results and the discussion and conclusion of the results.

CHAPTER 8: CONCLUSION Chapter 8 summarizes the conclusions of the three iterations and the final results of the conducted studies.

CHAPTER 9: FUTURE WORK Chapter 9 suggests development approaches for future work based on the results and the collected user feedback.

Part III: Text Input on Stationary and Mobile Devices

CHAPTER 10: INTRODUCTION Chapter 10 introduces the second topic of this dissertation which is to develop a fast and efficient stationary and mobile text input concept for multi-touchscreens which can be used while holding the tablet in both hands.

CHAPTER 11: REQUIREMENTS Chapter 11 explains the requirements on which the text input concept is built on.

CHAPTER 12: RELATED WORK Chapter 12 gives on overview of the related work in the context of stationary and mobile text input concepts. It further describes the theories and concepts of testing, and evaluating text input concepts.

CHAPTER 13: DEVELOPMENT Chapter 13 describes the text input concepts developed and evaluated in the context of this dissertation. It first explains our initial study of developing a text input concept designed to be used only with the thumbs while holding the tablet device in both hands. Next, it explains all three iterations of the stationary version of the Gestyboard concept, which is a text input concept based on the 10-Finger-System. This is followed by the description of the first iteration of the mobile version of the Gestyboard text input concept. Each iteration consists of the description of its concept, the evaluation, as well as its results and the discussion and conclusion of the results.

CHAPTER 14: CONCLUSION Chapter 14 summarizes the conclusion of each iteration and the final results of the text input part of this dissertation.

CHAPTER 15: FUTURE WORK Chapter 15 gives an overview of the most important ideas to improve the Gestyboard concept in a future iteration.

Part IV: Overall Conclusion and Discussion

CHAPTER 16: CONCLUSION Chapter 16 summarizes all conlclusions for map interaction and mobile text input.

CHAPTER 17: DISCUSSION Chapter 17 discusses the results and the conclusions of this dissertation.

CHAPTER 18: FUTURE WORK Chapter 18 gives an overview of ideas for future work in the fied of mobile user interfaces designed to be used in the context of MCIs.

Part I.

Introduction

1. Introduction

When an accident, such as a car crash, happens the rescue service personnel will usually take care of all patients at the same time. Whatever is needed to stabilize an injured patient will be done immediately to be able to transport the patient to the next available hospital as fast as possible. But if more casualties are involved in the accident than resources (all kinds of resources, including human resources) are available to the rescue service to treat each patient simultaneously, the treatment of the patients has to be prioritized. This kind of incident, in which more casualties exist than local rescue resources are available is called a **M**ass **C**asualty **I**ncident (MCI). This is a dynamic definition because it depends on the local rescue resources. Consequently, while an incident with five casualties is not an MCI in a metropolis like Munich it might be an MCI in a smaller village with less rescue resources. The prioritization of the casualties is a critical process because the order of treatment is essential to help and save as many casualties of the incident as possible. Thus, people with more critical injuries have to be treated and transported first. The medical rescue teams do not only have to face an immense organizational challenge, they also have to face a vehement psychological challenge. In order to have a deterministic, fair, and efficient way to determine those priorities, the rescue service personnel execute a so called triage algorithm. One example for a triage algorithm is the **S**imple **T**riage and **R**apid **T**reatment (START) algorithm [13]. A modified version of START is called mStart (modified START) [47] and is used in Munich. Both categorize the patients into different priorities regarding the need of each patient for medical care. The outcome of the triage algorithm is a color code indicating the severity of the specific patient's injuries: green - not seriously injured , yellow - more critically injured (but there is time to treat that patient), red - critically injured (the patient has to get immediately transported to a hospital to save the patient's life), and black - the patient is already dead. For some triage algorithms, there is also a blue color meaning that the patient is critically injured but special resources are not available (yet) to treat the patient. But this color is rarely used. The relief units of the rescue service perform the triage algorithm in field. They write down the number of injured people and their triage-priority on paper. The **A**mbulant **I**ncident **O**fficer[1] then collects the information taken from the relief units and tries to get an as accurate overview of the incident as fast as possible in order to save as many people as possible. Because of the fact that the current procedure is based on pen, paper, and person to person communication the information can be redundant, incomplete, and/or wrong.

This is exactly the point at which the SpeedUp project, described in more detail in Section 2.1, tries to improve the handling of MCIs. As underlying philosophy, modern technology is meant to enhance existing procedures rather completely replace them. One part of the SpeedUp project focused on digitizing the triage process in order to be able to collect

[1]In Germany this role is called **Org**anisatorischer **L**eiter Rettungsdienst (OrgL).

all MCI related information in a database. During the SpeedUp project, PDAs have been used to evaluate that approach [74]. Once this information is stored in a database, it could be used to visualize the MCI area with all collected MCI related information on a map application. The approach of SpeedUp was to provide this map on three different classes of devices: 1) **Mobile Devices** such as PDAs or smartphones 2) **Semi-Mobile Devices** such as ruggedized tablet devices and 3) **Stationary Devices** such as tabletop computers. The role of TUM, explained in more detail in Section 2.2, was to ensure that the target group accept the map application by developing special **u**ser **i**nterface (UI) elements and features based on the strong, special requirements (discussed in Section 5) of MCIs.

This dissertation focuses on the iterative and user-centered UI development for the semi-mobile devices (ruggedized tablets). The reason for using ruggedized tablets rather than standard tablets is given in Section 5.2. Those ruggedized tablets are usually heavy compared to the standard tablets. To enable the AIOs to still use these tablets in an ergonomic and efficient way, even when being mobile, each UI element and interaction feature developed in the context of this dissertation can be used while holding the tablet device in both hands. This requirement is described in Section 5.4. By considering this requirement, different UI elements for interacting with a digital map have been developed from scratch (in deep discussions with people from the target group) and optimized in three iterations. A detailed description of each iteration of this development is given in Part II of this dissertation. However, during our first requirement analysis with people from the target group we learned that general text input is a very important requirement during an MCI. But when holding a tablet device in both hands, standard touchscreen keyboards cannot be used for text input because the thumbs simply are not able to reach the whole keyboard. Because of that fact, a second development track was started in the context of this dissertation: Mobile Text Input. This requirement lead to the innovative text input concept called Gestyboard. Details about the Gestyboard concept and its development are given in Part III of this dissertation. Although a major amount of Part III deals with the development of the stationary part of the Gestyboard the goal of the whole development process is to enable the user to use the eight fingers on the back of the device by using the touch-typing system (also called 10-Finger-System) while holding the tablet in both hands. Because this is a completely new technique for text input, we decided to use modern tablet devices with much better touchscreens than the ruggedized tablets used otherwise in this dissertation for the development. The reason for this decision is given in Part III in more detail. This part first describes our initial text input related study and then describes the three iterations of the development of the stationary version of the Gestyboard. Last but not least, the first iteration of the Gestyboard Backtouch version (the initial intention to develop the Gestyboard) is described.

2. Background

In this chapter the background of this thesis is introduced and explained. First, the SpeedUp project is introduced in Section 2.1. The studies of this dissertation all have been carried out in the context of this project. Then, the role of **T**echnische **U**niversität **M**ünchen (TUM) in this project is described in Section 2.2. This is followed by the most important fundamentals of Usability and Human-Computer-Interaction in general in Section 2.3. Next, some touchscreen related basics are introduced in Section 2.4 because all UIs of this dissertation are based on touchscreens.

2.1. Project SpeedUp

All studies presented in this dissertation have been carried out in the scope of the SpeedUp[1] Project. As a consortium of seven partners, SpeedUp investigated self-organizing and mobile communication and data platforms as well as strategies for the organization and action in complex, large scale situations. Those large scale situations are called **M**ass **C**asualty **I**ncident (MCI). By definition, an MCI is an incident which involves more patients than the locally available rescue workers can manage simultaneously. As a consequence, the leading personnel of the rescue service has to determine the severity of the injunries of each patient first. To be able to do so in a subjective and fast way, the relief units perform a triage algorithm such as the **S**imple **T**riage **a**nd **R**apid **T**reatment (STaRT) [12] which has been developed in 1996 or the **m**odified **S**imple **T**riage **a**nd **R**apid **T**reatment (mSTART) [47] which is an updated version developed in 2006. Both provide a deterministic way to determine the priority of a patient in a short time period. To remember and share this priority information, the patients receive paper triage tags. Each paper triage tag has one of the following five colors indicating the priority of the patients' treatment order: Black, Blue, Red, Yellow, and Green. Black means that the patient is dead, thus the patient does not need medical care. If a patient gets a blue tag, currently available resources are not sufficient to treat that patient. Although alive, people with blue tags are almost treated like people with black tags to increase the chance to be rescued for those who have a real chance to survive. Patients with red tags have the highest priority in an MCI. It is the goal of the **A**mbulant **I**ncidient **O**fficer (AIO) (leader of the rescue service) to get those patients transported to nearby hospitals as fast as possible, since every second counts for those patients. Yellow color means that the injuries of the patient are not that critical, i.e. the patient's live is not in acute danger. This may change over time but there is more time left for the treatment and transportation of yellow patients than for the critically injured red patients. Green patients are the least injured and can usually even walk by themselves.

[1] The project SpeedUp is funded by the German Federal Ministry of Education and Research (BMBF) within the program "Research for Civil Security" (May 1st, 2009 - April 30th, 2012, FKZ: 13N10175). Website: http://www.speedup-projekt.de

The triage procedure is the first step of the rescue service during an MCI, which is the reason why the triage algorithm is called STaRT. However, there are also challenges stemming from the current approach. Important information like the number of patients, their priorities, positions, and injuries are all collected by paper and are shared by phone or face to face. The rescue teams are well trained to manage those situations, but the phase of collecting and sharing important information can be chaotic, slow and error-prone. Besides that, one ethical issue is that patients with yellow tags usually do not feel like being less critically injured than the red tagged patients. Because there is nothing else than the paper triage tag indicating the priority of the patients, it is possible that patients exchange their yellow tags with patients (e.g. unconscious) who were tagged as red nearby to them.

The overall goal of the SpeedUp project was to extend the current approach with an IT infrastructure rather than replacing it. This way, the rescue teams can still rely on the traditional system as a safe fallback while they also gain from the support of an IT infrastructure. By equipping each triage team with a smartphone, not only the teams themselves can be tracked, also the patients' sighting position can be collected during the triage process and distributed through a network [74]. This can then be visualized within a map application on a tablet device for mobile use and a multitouch table for stationary use [4]. This provides a real-time overview for the AIO or other leading personnel of the rescue service. Although this can heavily support the AIO to manage the MCI, it requires the user to deal with new devices and spend time and attention to the device while interacting with it. Within the SpeedUp project, the **T**echnische **U**niversität **M**ünchen (TUM) developed several alternative UI concepts iteratively for those devices and evaluated them with people from the fire department (Fire department TUM) and the rescue service (Arbeiter-Samariter-Bund München).

2.2. The role of TUM in SpeedUp

Our role (TUM) in the SpeedUp project was to study the **H**uman-**C**omputer **I**nteraction (HCI) part of the project and conduct Usability Engineering close to the target group in an iterative development design process. This thesis focuses on HCI aspects of ruggedized mobile tablet devices in MCI situations. Because of the critical, unpredictable and dangerous nature of an MCI there are high demands on the hardware. As a consequence, the commercial standard tablet devices were considered not to be sufficient to be used in that context. Hardware fulfilling those high demands is not only much more expensive, it is also heavier and less comfortable to use. This directly leads to special requirements regarding the UI to be able to support an ergonomic, fast, and comfortable way to interact with the mobile device. The mentioned hardware requirements originated from the environment. They are presented and discussed in further detail in Section 5.2. The raised ergonomic issues and the resulting software requirements are given in Section 5.4. Additionally, there are touchscreen requirements in general which have to be taken into account when developing, designing, and evaluating the UI alternatives.

This dissertation deals with the Human-Computer Interaction procedure for the following two main topics: Mobile map interaction on ruggedized tablet devices (see Chapter

4 and an innovative new gesture based concept for stationary and mobile text input (see Chapter 10. The design process is based on the standard Human-Computer Interaction procedures suggested by ACM SIGCHI and Nielsen's Discount Usability Enginnering. The underlying theories are summarized in Section 2.3.2 and 2.3.2 respectively. The resulting adapted development approach is described in Section 3.

2.3. Usability and Human-Computer Interaction

In this section some important aspects of the theoretical background of Usability and HCI are presented. In Section 2.3.1 the definitions of the term Usability and HCI are given. Then, Jakob Nielsen's discount usability engineering procedure is described in Section 2.3.2. Afterwards, a brief introduction of the terms between-subject design and within-subject design is given in Section 2.3.3. This is then followed by the description of the used questionnaires in Section 2.3.4.

2.3.1. Definition of Usability and Human-Computer Interaction

The *Usability* of a **u**ser **i**nterface (UI) consists of the following measurable components: *Learnability*, *Efficiency*, *Memorability*, *Errors*, and *Satisfaction* [75]. Thus, Usability is a multi-dimensional property of a UI and not just a single one. The respective components may have different priorities in different usability projects. The complexity is additionally increased by the complexity of human nature and psychology [81]. Thus, there are much more parts which have to be considered in order to develop a high quality UI. In 1992, ACM SIGCHI published a curriculum for Human-Computer Interaction in which the term "Human-Computer Interaction" is defined as follows: *"Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them"* [16]. The "surrounding major phenomena" and the interrelationships between them are illustrated in a graphic which is also provided in the curriculum and can be seen in Figure 2.1.

Besides the human and the computer, the use and the context of the software and an iterative development process is part of the field Human-Computer Interaction. Studying the future use and context of a software is essential in order to be able to adapt the software in a way that it suites the requirements of the user's workflow. The iterative development process is also essential because the results of each iteration is the input of the next iteration. This way, the developer learns which approach worked well in the current iteration and which features should be improved for which reason. This enables the UI developer to improve the interface step by step according to the users' subjective rating but also according to objective measurements. The use and context and the iterative development process is especially important in the case of the SpeedUp project which is the scope of this dissertation (see Section 2.1). The real environment in an MCI is much more important than using the software. Thus, the software should support the Ambulant Incident Commander and the Relief Units instead of distracting the attention from the physical reality to virtual information. Additionally, the UI has to be perfectly adapted to the current procedure of the rescue service during a rescue operation such that it can really speed up

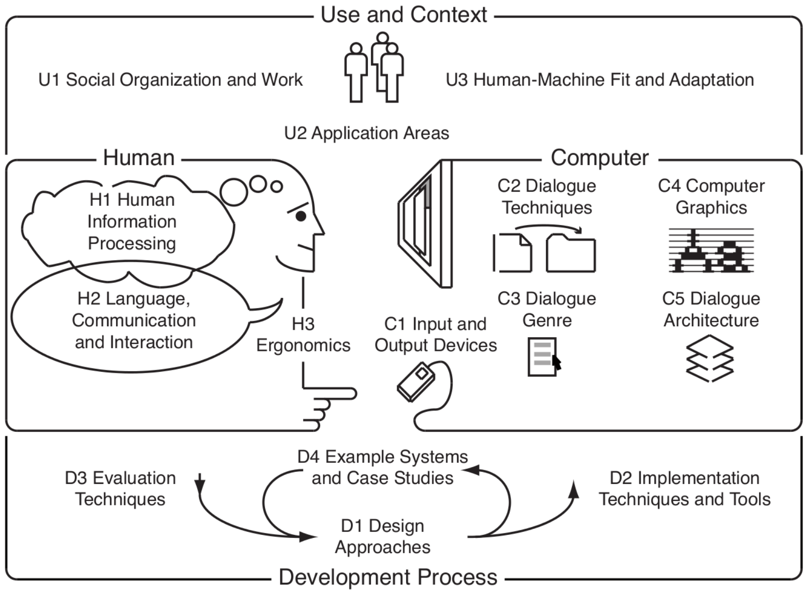


Figure 2.1.: Human-Computer Interaction defined by ACM SIGCHI in 1992 (p.13-17)

the operation instead of slowing it down and putting additional complexity into it. As described in Section 2.1, especially the first minutes are the most chaotic phase in an MCI. Thus, aside from the unique requirements presented in Section 5 and 11 some of the well-known standard requirements like efficiency, effectivity, reliability, intuitivity, simplicity, and interruptibility are strongly emphasized in the context of an MCI. Each second lost because of bad usability or frustrated personnel can cost live in that critical environment.

2.3.2. Discount Usability Engineering (J. Nielsen)

In 1993, Jakob Nielsen proposed using a “discount usability engineering” approach to achieve the right trade-off between a good User Interface and the costs to build it. This “discount usability engineering” consists of four basic techniques [75]:

1. User and task observation
2. Scenarios
3. Simplified thinking aloud
4. Heuristic evaluation

First of all, some kind of **user and task observation** have to be done to gather the needs and the expectations of the users. This first step is very important in order to develop a UI which is adapted to the special needs of the target group. The user should always be in the center of the development process. The second technique Nielsen suggests is the use of simplified **scenarios** in prototypes. This means, that only the relevant part of the new software will be developed for the scenario-based evaluation in order to get tested with people from the target group of the software. This reduces development costs and also the complexity of the evaluation design. The third technique is the **"thinking aloud"** technique. But there are also disadvantages coming with this technique. For example, if the evaluation goal is to find out how efficient and fast the user can work with its UI it does not make sense to use this technique because the user's action will slow down a lot when being asked to think aloud while interacting with the interface. However, if an interview is conducted afterwards, some important feedback might be lost because the user usually cannot remember everything he thought about during the test. It depends on the evaluation goals whether to use this technique or whether to prefer conducting an interview afterwards. The last technique which is part of Nielsen's "Discount Usability Engineering" is called **Heuristic Evaluation** [76]. This is a usability guideline focusing on the most important usability issues. This guideline consists of the following ten items: (1) Simple and natural dialogue, (2) Speak the users' language, (3) Minimize the users' memory load, (4) Consistency, (5) Feedback, (6) Clearly marked exits, (7) Shortcuts, (8) Good error messages, (9) Prevent errors, (10) Help and documentation. A detailed description of each of these ten items can be found in [76].

In 2000, an international standard was published (EN ISO 13407) for user-centered design processes, suggesting similar steps as the four main steps by Nielsen: (1) Understand the users context, (2) Specify Requirements, (3) Develop Solutions, (4) Evaluate solutions. This standard was updated in 2010 (ISO 9241-210). However, the main steps are still the same that Nielsen proposed in 1993. The differences can be found for example in wordings, more depth and more specific descriptions.

2.3.3. Within-Subject vs. Between-Subject Design

When comparing different UIs in an evaluation one has to choose between two opposing evaluation methods: **Between-Subject** and **Within-Subject** design. In an Between-Subject design there is a group of test users for each UI alternative. Thus, each test user only sees and rates one interface variant assigned to that group. Opposite to that, there is only one group of users for Within-Subject designs. Here, each test user sees and rates all alternatives. There are well known advantages and disadvantages for both techniques. In an Between-Subject Design, the test users are not able to compare each alternative, since only one is presented to them. On the one hand, this is an obvious disadvantage. On the other hand, this is also an advantage, because test users cannot be biased from the previously solved task with the other UI alternative. In addition to that, the users do not get as fatigue as when evaluating all developed alternatives. Those properties are exactly inverted in the case of a Within-Subject design: The test users are able to compare the alternatives, since all alternatives were presented to them. To compensate the previously mentioned bias effects through learning, the sequence of the alternatives has to be permuted through the test users. The number of test users have to be chosen in a way that each sequence appears ($number\_of\_test\_users = n * x!$ where n is a non-negative integer and x the number of alternatives). Another disadvantage of the Between-Subject design can be found in the differences of the users in the different groups. Because of the wide variety of user skills and knowledge there is a risk that the group will be biased. This can be accommodated by assigning the alternatives in a permuted order to the different users. However, of course more test users are needed [75]. Consequently, if the users from the real target group are rare for any reason, the Within-Subject design is often chosen because of its efficient use of test users. However, one important parameter when designing the evaluation is the time which each test user has to spend with the evaluation tasks. If test users have to spend too much time with the evaluation they might get frustrated and just want to finish the task as fast as possible. As a consequence, the results of the evaluation can be influenced.

2.3.4. Used Questionnaired: SUS, AttrakDiff, and NASA-TLX

In order to be able to compare the quality of the developed interface with other interfaces of other applications it makes sense to use well established questionnaires. Although a specific interface for a specific application will never be completely comparable to any other interface of any other application using the same questionnaires leads to a common understanding of the results. This increases the comparability. Another reason for using already existing questionnaires is the fact, that many researchers spend a lot of time, effort, and knowledge into the development of those questionnaires. Consequently, the questionnaires are optimized according to psychological factors as well as to technical factors. In the studies described in this dissertation, the main questionnaires used were the **S**ystem **U**sability **S**cale (SUS), AttrakDiff, and NASA-TLX. Those questionnaires are described in the following paragraphs.

System Usability Scale (SUS) Having a fast and easy way to learn about the subjective opinion of the users is very important because the users have to already spent time with trying out the new system. If they have to fill out too many questions after the test is finished, the probability that they might get frustrated and annoyed increases. This can have a direct influence on the subjective opinion of the users. The purpose of the SUS questionnaire is to provide such a fast and easy way to seek the subjective opinion from the users of a system on a particular device. The SUS questionnaire was developed by John Brooke and published in 1996 [14]. It consists of 10 items which have to be answered by the users through a 5 (or 7) point Likert scale. The 10 items of SUS are shown in Table 2.1.

The questionnaire results in a SUS score ranging from 0 to 100. The odd numbered items' contribution is the position of the Likert scale minus 1 because they are positive items. To compute the contribution of the even numbered items (negative items) is the maximum value of the Likert scale (5 or 7) minus the scale position. The sum of those scores is then multiplied by 2.5 which results in the final SUS score.

AttrakDiff - Measuring the Pragmatic and Hedonic Quality of a System AttrakDiff is another well-established questionnaire which measures the pragmatic and hedonic quality of a system. The pragmatic quality measures the quality of the system in terms of the usage of the system. The hedonic quality measures the quality of the system in terms of human needs of stimulation and identification [35], [6]. Stimulation targets the human wish to develop oneself in terms of knowledge and capabilities. The interactive system can influence this development positively by providing innovative, interesting, and inspiring

Item	Stimme überhaupt nicht zu 1	2	3	4	5	6	Stimme voll zu 7
I think that I would like to use this system frequently	❑	❑	❑	❑	❑	❑	❑
I found the system unnecessarily complex	❑	❑	❑	❑	❑	❑	❑
I thought the system was easy to use	❑	❑	❑	❑	❑	❑	❑
I think that I would need the support of a technical person to be able to use this system	❑	❑	❑	❑	❑	❑	❑
I found the various functions in this system were well integrated	❑	❑	❑	❑	❑	❑	❑
I thought there was too much inconsistency in this system	❑	❑	❑	❑	❑	❑	❑
I would imagine that most people would learn to use this system very quickly	❑	❑	❑	❑	❑	❑	❑
I found the system very cumbersome to use	❑	❑	❑	❑	❑	❑	❑
I felt very confident using the system	❑	❑	❑	❑	❑	❑	❑
I needed to learn a lot of things before I could get going with this system	❑	❑	❑	❑	❑	❑	❑

Table 2.1.: The 10 items of the System Usability Scale

content, designs, and functions. Identification targets the human wish to personalize their system and therefore identify themselves with the system. The 28 items of the AttrakDiff questionnaire are given in Table 2.2.

To compute the score of the AttrakDiff questionnaire, the developers offer a website in which the subjective user ratings can be entered [6]. The evaluator gets a full report of the result analysis separated into hedonic and pragmatic quality.

NASA-TLX NASA-TLX is a questionnaire which measures the subjective workload assessment of the users [71]. It computes an overall workload score based on a weighted average of ratings on six subscales. Figure 2.2 shows those six subscales.

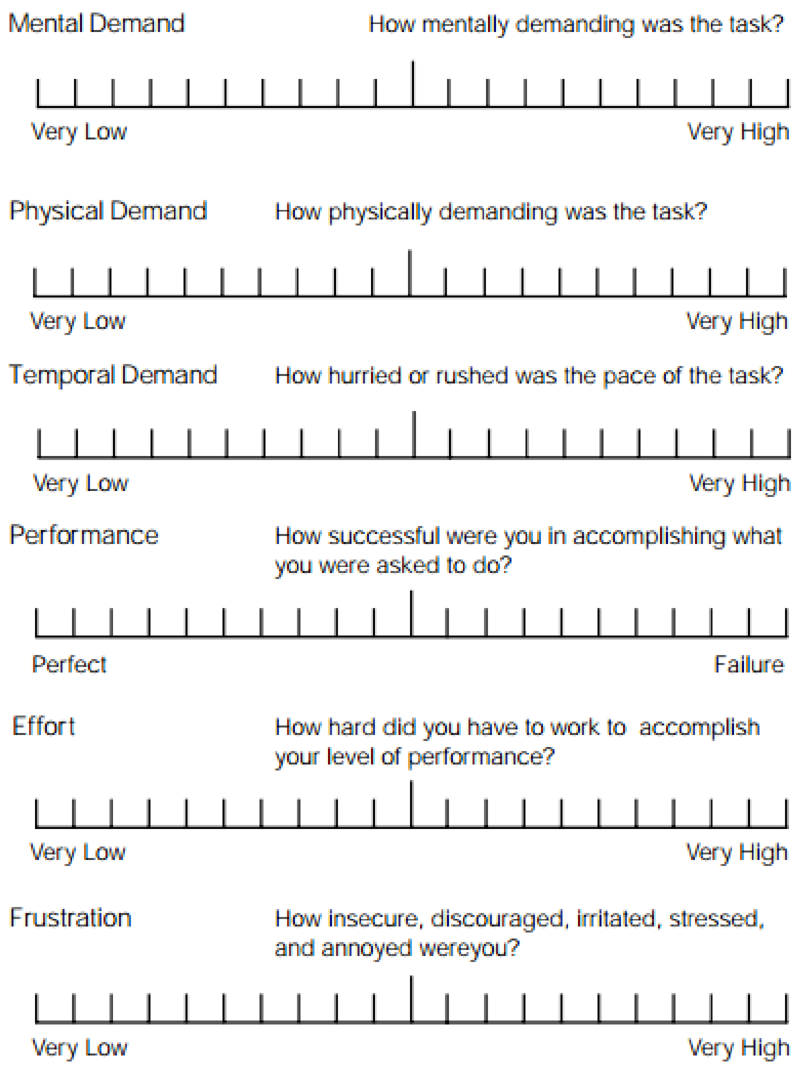


Figure 2.2.: The NASA-TLX questionnaire [72]

human	❑	❑	❑	❑	❑	❑	❑	technical
isolating	❑	❑	❑	❑	❑	❑	❑	connective
pleasant	❑	❑	❑	❑	❑	❑	❑	unpleasant
inventive	❑	❑	❑	❑	❑	❑	❑	conventional
simple	❑	❑	❑	❑	❑	❑	❑	complicated
professional	❑	❑	❑	❑	❑	❑	❑	unprofessional
ugly	❑	❑	❑	❑	❑	❑	❑	attractive
practical	❑	❑	❑	❑	❑	❑	❑	impractical
likeable	❑	❑	❑	❑	❑	❑	❑	disagreeable
cumbersome	❑	❑	❑	❑	❑	❑	❑	straightforward
stylish	❑	❑	❑	❑	❑	❑	❑	stacky
predictable	❑	❑	❑	❑	❑	❑	❑	unpredictable
cheap	❑	❑	❑	❑	❑	❑	❑	premium
alienating	❑	❑	❑	❑	❑	❑	❑	integrating
brings me closer to people	❑	❑	❑	❑	❑	❑	❑	separates me from people
unpresentable	❑	❑	❑	❑	❑	❑	❑	presentable
rejecting	❑	❑	❑	❑	❑	❑	❑	inviting
unimaginative	❑	❑	❑	❑	❑	❑	❑	creative
good	❑	❑	❑	❑	❑	❑	❑	bad
confusing	❑	❑	❑	❑	❑	❑	❑	clearly structured
repelling	❑	❑	❑	❑	❑	❑	❑	appealing
bold	❑	❑	❑	❑	❑	❑	❑	cautious
innovative	❑	❑	❑	❑	❑	❑	❑	conservative
dull	❑	❑	❑	❑	❑	❑	❑	captivating
undemanding	❑	❑	❑	❑	❑	❑	❑	challenging
motivating	❑	❑	❑	❑	❑	❑	❑	discouraging
novel	❑	❑	❑	❑	❑	❑	❑	ordinary
unruly	❑	❑	❑	❑	❑	❑	❑	manageable

Table 2.2.: The 28 Items of the AttrakDiff2 questionnaire

To compute the NASA-TLX score, the authors offer an online version of the questionnaire in which the results can be entered and automatically computed.

2.4. Touchscreens

Even though touchscreen technology only became widely distributed and ubiquitous during the last five years through the wide spread of mobile smartphones and tablets, the first touchscreen technology has already been developed five decades ago. Eric Arthur Johnson is considered to be the first who invented a touch-sensitive screen (capacitive) at the Royal Radar Establishment in Malvern, UK around 1966. In 1969, a US patent about Johnson's touch displays (US Patent 3,482,241) has been published [44]. In this section, the different technologies which have been developed since the first introduction of touch sensors are described in Section 2.4.1. Afterwards, the advantages as well as the challenges of touchscreens, considering HCI aspects, are presented in Section 2.4.2.

2.4.1. Technology

Today, different types of touchscreen technologies exist. Ruggedized tablet devices usually use other technologies than the widely distributed standard smartphones and tablet devices. The choice of the technology has an impact on the HCI experience of the user. For this reason, the most important touchscreen technologies (in the context of HCI) are presented in the following.

One important aspect of touchscreens is how the touch event is detected by the hardware. In general, the *touch activation mechanism* can either be based on the pure contact with the finger or on a specific force which the user has to apply before the touch is detected. In the case of capacitive touchscreen, a bare finger contact is sufficient for the system to detect the touch. This way, the touchscreen feels very sensitive for the user, making such touchscreens very comfortable to use. Among others, this might be the reason why most of current smartphones and tablet devices use capacitive displays. As a negative side effect, it is not possible to touch the display without interacting with it when it is on. But more relevant than that in the context of the rescue service is that it is also not possible to interact with the capacitive display when wearing normal (non-prepared) gloves. Although the target group, the AIO, usually does not wear gloves, it can happen due to the weather or other context related reasons that they wear gloves. The mobile device should not force the user to remove them in that case before interacting with the device, since this would dramatically reduce the value and the acceptance of the device. In contrast, touchscreens with resistive sensors can be used even when wearing gloves. Those sensors can be activated by anything. Just the pressure on the display has to be high enough. Resistive touchscreen technology has also improved a lot in recent years. By now, a slight level of force is sufficient to activate a touch for standard devices. Unfortunately, for ruggedized tablet devices this is not the case. To protect the device for example against rain, the whole device needs to be consolidated. This is also the case for the screen itself. As a consequence, the touchscreen of a ruggedized tablet device is also covered by a transparent protection layer. The user has to apply even more force when touching the screen in order to make the system recognize the touch. This has an major influence on the usability of such a device and has to be considered in the development process.

Another important property of touchscreens is the used *input "device"*. Due to the missing precision of the finger (see Section 2.4.2), the first commercial touchscreen devices used a stylus as the standard input device. Thus, the first approach was to adapt the technology to that what the user were used too. The user was used to very precise interaction with the mouse and its cursor. However, currently available smartphones just offer the finger interaction by default. This is because in recent years the smartphone companies started to adapt the UI of the operational system of the mobile device instead of developing additional input devices. This is exactly, what TUM suggested for the use of ruggedized tablet devices: Adapt the UI instead of (or in addition to) the technology to the users' needs.

Another relevant issue to improve the interaction experience in general with touchscreens is to simulate the missing intrinsic feedback of real hardware buttons. An inexpensive way to get close to this goal is to simulate the sound of a real button when pressed, giving some acoustic feedback. Another way is to also provide vibrato-tactile feedback. This of course has to be supported by the hardware. While almost all new smartphones and tablet devices support this feature, for now this is usually not the case for ruggedized tablet devices.

2.4.2. Advantages and Challenges of Touchscreens

The most obvious advantage of touchscreens is that the user can directly interact with interface elements making it more intuitive and natural. Furthermore, additional input devices such as traditional mice and keyboards are not necessary, resulting in a more compact system which also increases the mobile capability. In addition to that, UI elements on touchscreens are much more flexible than for example hardware buttons. Consequently, the UI can be adapted to the users' current actions and context. For example the shape, size and/or color can be changed indicating the current state of the element or it can be completely hidden, if it is not needed during the current context. This kind of context dependent buttons can also be obtained by dynamic hard buttons but users often feel confused by them, since their performed actions are changing, but there label is not. On touchscreens all buttons can be labeled with visual symbols which makes their actions easier to understand and therefore more intuitive.

Despite the great success and distribution of touchscreens in recent years, there are also challenges coming with them. For example, when the user presses a soft button, its label is occluded by the finger. While this is not a problem in some cases (like the simple activation of a button) it can be annoying and frustrating the user in other cases (e.g. when dragging or adjusting something). Another issue is that users are used to the "mouse-over" effect to display for example additional information about the element below the cursor. This well-known feature is missing in the case of touchscreens. Besides this, people are also not as precise with their fingers as with the traditional mouse cursor. This is because of the fact that the resolution of a fingertip is low [2]. Consequently, the UI elements have to be large enough to avoid missing them when trying to touch them. According to literature, an accuracy of 99% can still be reached if the button width is larger than 22 mm [55], [53]. If a soft button needs to be smaller (for example for small screen devices), techniques to reduce the size without losing accuracy can be used. With the so-called "take-off" strategy for example the size of a soft button can be reduced to 6.8 mm [87]. A cursor is placed above the fingertip when using this technique. But it also reduces the interaction speed of the user. Among others, Lee et al. compared the performance of soft buttons and hard buttons [54]. In this study, it has been shown that the performance of a soft button suffers when the size falls below a fraction of the finger width. Last but not least, the missing tactile feedback leads to one of the most important challenges related to touchscreens. In general, there are two ways to deal with this challenge. One way is to provide some kind of tactile feedback with special hardware (e.g. vibrato-tactile or audio feedback). The quality of this artificial feedback is not as high as it is for hardware buttons. It significantly increases the overall efficiency of the usability though [58], [7]. The second way is to design the UI elements in a way that they do not rely on tactile feedback. An example for this second approach is given in chapter 10.

3. Development Approach

The development procedure for the mobile map application UI as well as for the mobile text input application is based on Nielsen's concept of Discount Usability Engineering, described in Section 2.3.2. The user is the center of the entire development process, which consists of several iterations in which the UI improves from iteration to iteration. Figure 3.1 illustrates our adapted approach.

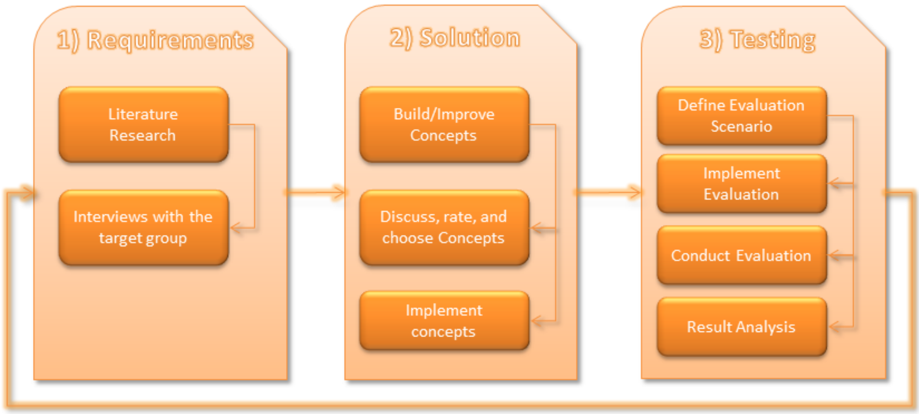


Figure 3.1.: General development approach built upon Nielsens discount usability engineering

The first step of each iteration is to gather field-related requirements. This part is called "Use and Context" in the curricula of Human-Computer Interaction described in Section 2.3.1. In the case of this dissertation the requirements were mainly collected through literature research and iterative interviews with the target group of the system. Of course, the outcome of a previous iteration is also an important source for updating existing requirements to new findings and to define new requirements, if necessary. This step is similar to Nielsen's first step called "User and task observation". Although multiple user and task observations have been performed in our development process too, the outcome of the interviews were much more efficient in our case, since a real MCI is rare and if one occurs, the highest priority will always be to rescue life and not showing scientists their way of handling the incident. To get a feeling of the users tasks in action it was also helpful to attend real exercises for MCIs and to attend the critical discussions which are always performed right after the exercises. However, when attending such an exercise as a "third person" it is extremely difficult to see any system in the process of the exercise, because there are many people involved with different roles and it is impossible to follow everything. The development process of the solutions for the collected requirement is then step two in our approach. This coincides with Nielsen's second step: "Scenarios". When using ”Scenarios“ just that part of an UI which is needed for the evaluation as a prototype will be developed. This also conforms to the ACM SIGCHI’s interpretation of the term Human-Computer Interaction. In our case, in all our iterations we developed/improved multiple alternatives to enable the same interaction feature to be able to compare the results to each other. Before implementing the concepts, it is always important to get other UI experts’ feedback to improve the concepts even before it is introduced to future users from the target group.

The concepts rated best by the experts are then chosen and implemented to be evaluated with the target group. This is step 3 in our approach. First, the scenario in which our concept can be tested is defined. Then everything needed for the evaluation get implemented (logging, timer, User Guidance for the evaluation part, the task itself, ...). Afterwards, an evaluation involving the intended users of the system is conducted and the results are analyzed and interpreted. This differs a little bit from Nielsen’s Usability Engineering. For evaluating, we did not adhere to Nielsen’s Discount Usability Engineering because the thinking aloud technique (Nielsen’s step 3) also slows the user down while interacting. In our case, this was not possible because execution time and efficiency were always important parameters. Instead, interviews were conducted based on [73] and the test users were anonymously recorded for later observations and analysis. Those interviews are also always a great input for the requirement analysis of the next iteration. Step 4 of Nielsen’s Usability Engineering is the heuristic evaluation. The categories listed there are still valid and every UI should follow those rules. The heuristic evaluation was always a part of the discussions in step 2 (discussion, rate, and choose concepts) in our case.

Part II.

Mobile Map Interaction

4. Introduction

Part II of this thesis focuses on the user-centered development process of the UI concepts for mobile map interaction on heavy ruggedized tablet devices. The work described in this part has been performed in the context of this thesis and the SpeedUp project. The purpose of SpeedUp was to investigate the integration of modern technologies with the standard traditional way of handling an MCI. A major part of the project concerned developing a map application in which MCI-related Data can be visualized. The AIOs (Ambulant Incident Officer) are able to gain a much better overview of the MCI than compared to the traditional way of handling the MCI. The role of the TUM (Technische Universität München) within the project was to develop and evaluate different UI solutions which can be comfortably and efficiently used during the critical situation of an MCI. The challenge here is to develop the interaction metaphors closely to the target group to ensure that all requirements of the target group and of the MCI itself are met by the single interaction metaphors. SpeedUp is introduced in Section 2.1 in more detail. This part of the thesis presents the development approach of the map application interaction metaphors. These studies have been conducted from 2009 to 2013 at TUM. The goal of this process is to develop UI solutions to enable the AIOs to perform the main digital map interaction operations while holding the heavy tablet device in both hands: Scrolling the map, zooming the map, and selecting items on the map.

The general development approach is described in Chapter 3. The special requirements related to the hardware itself, the touchscreen, the MCI environment, and also to ergonomic issues are discussed in Chapter 5. Related work is presented in Chapter 6. This is followed by the description of the three iterations which have been performed in the context of this thesis in Chapter 7. Each iteration closes with its own conclusion and discussion. The overall conclusion considering mobile map interaction on ruggedized tablet devices are given in Chapter 8. Last but not least, a future work outlook for that field is given in Chapter 9.

5. Requirements

In this chapter the gathered requirements in the context of the first development step in each iteration (Figure 3.1) are classified and described. First, common requirements are given in Section 5.1, then requirements related to the used hardware are presented in Section 5.2 which is followed by touchscreen related requirements in Section 5.3. Afterwards, special ergonomic requirements on which the UI design is based are addressed in Section 5.4.

5.1. Common Requirements

One common requirement is to provide an intuitive, easy to learn and easy to remember UI to the user. In fact, this requirement is very important for all UIs, and it is especially important in the case of an MCI situation. The reason behind that is the fact that a special training of rescue personnel to use the new system would cost effort and time. Another reason is that the user should be able to immediately understand the information presented by the UI. The user should also be able to perform the basic interaction with the application when using it the first time. Furthermore, the target group (the AIO in our case) will not accept a new application if they have to spend too much time to learn how to use it. Things need to work well in emergencies (not a standard situation).

The second common requirement is the attractiveness of the interface. If the users do not like the appearance of the interface, then there is a higher risk for rejection [1], [34]. Last but not least, the next two requirements are especially important in the case of an MCI: Efficiency and accuracy. The efficiency is particularly important since adding IT in the scope of the rescue service intends to speed up the rescue process. However, developing efficient UI elements should not lead to inaccurate processes.

5.2. Hardware Requirements

Each second counts in an MCI situation. To avoid any delays and frustration because of computation times, the performance of the hardware should be as high as possible. Another basic hardware requirement is the capability of tracking the position of the hardware and with that the position of the AIO as well. All of the rescue units will each carry a smartphone, also equipped with a GPS receiver. This way, the positions of the rescue teams and the leading personnel can be visualized providing an overview of the current situation and resources. The tablet PC has to be equipped with WiFi to be able to connect to a nearby server and get the PDAs GPS positions. As fall back strategy, other possibilities to submit data like GPRS, EDGE and UMTS should also be available. However, the specific tracking technology and the network connection is not the scope of this work. Another relevant hardware requirement is the battery life that should last at least four hours. Additionally, a car docking station for the tablet PC should be available to be able to recharge batteries during the trip. Due to the nature of an MCI, the environment in which the ruggedized mobile device is used can be unstable, unsafe, critical, and unpredictable. The used hardware has to endure environments in which fire, water or dust exists. There are known standards which define the degree of protection for those devices. In an interview with the Fire department TUM we learned that the device should comply at least with IP55 defined in DIN EN 60529. That means that the device offers a full protection against contact and interior injurious dust deposits and it is protected against water (out of a nozzle) from all directions. The SpeedUp project profited here from the experience of the Fire Department TUM because they already used such a ruggedized tablet device, the XPlore 104 in the second version (shown in Figure 5.2 a)) , and shared their experience with us. According to that, the device itself fulfills the hardware requirements but also raises new requirements related to ruggedized touchscreens (see section 5.3) and the heaviness (see section 5.4) of those devices.

5.3. Touchscreen Related Requirements

The high degree of hardware robustness (IP55) leads to the following two facts: The first is that there usually is a fully transparent protection layer above the touchscreen. The second is that most tablet devices, which fulfill such a high protection standard, have resistive rather than capacitive displays. To perform a touch on a resistive display, the user has to exceed a specific amount of pressure with the finger on the touchscreen. Because of the mentioned protection layer, this pressure has to be even much higher than for standard non-ruggedized resistive displays. One advantage is that the user can touch the screen without immediately interacting with it. Another advantage is that the touchscreen can also be used while wearing gloves. We learned from the ASB that the AIO does not always wear gloves, but due to the circumstances and the environment of the MCI it still can happen. The mentioned protection layer additionally decreases the sensitivity of the touchscreen. The negative side-effect of protected resistive touchscreens on ruggedized tablet devices is that sometimes an intended touch from the user is not recognized as a touch event by the display. If the software does not receive any touch, the system is actually not able to do anything against this problem. Additionally, if the user is performing, for example, a sliding gesture on the touchscreen it also can happen that the display loses the touch event while the user is still sliding the finger on the screen. This is because during sliding gestures users tend to release the pressure on the screen a little bit. When the display loses the touch and then recognizes it again, it is like a release and a new touch event, while the user was actually moving the finger around on the screen. Consequently, there is a touch event in between the sliding gesture of the user. This can lead to non-intended activations of features. This again leads to frustration by the user. However, if the time in which the contact to the finger is lost is very short, the interface can be implemented such that it ignores the new touch event and continues tracking the sliding gesture of the user.

It is also worth mentioning that this touchscreen issue has been rated as a minor problem by the Fire department TUM and that should be ignored according to their opinion.

When asked for the reason for this, the personnel from the Fire department all answered the same: "When we intend to perform a touch, we do it right, like tough men do." However, in the second half of the SpeedUp project, the **A**rbeiter-**S**amariter-**B**und (ASB) (a large rescue service organization in Germany) in Munich became part of SpeedUp. The mobile market, touchscreen technology, and also the users' expectations have developed extremely fast during the first year of SpeedUp. While the Fire department TUM did not change the requirements we gathered at the beginning, the ASB which was new in SpeedUp, had different expectations and requirements considering the hardware. Almost the entire staff of the ASB was already using a smartphone with modern capacitive highly responsible displays. Consequently, they were surprised how much more pressure they had to put on the ruggedized tablets screen before the touch was recognized as a touch by the system. They would have preferred a less ruggedized tablet device before accepting a device which feels like being from the past compared to the new commercial smartphones and tablets. However, the ruggedized tablet market is also evolving right now. Especially Google's Android OS leads to a completely new generation of ruggedized tablet devices. Because those new-generation tablet devices just started to appear at the end of the SpeedUp project, this hardware was not available for the SpeedUp project at that time. Thus, it would be interesting to repeat the studies presented in this section with those new hardware in future. In our studies however, we used the Xplore iX104 (Figure 5.2 a)) during the first half and the Durius V200 (Figure 5.2 b)) during the second half of the project. The first one was using a modified version of Windows XP, and the latter used Windows 7 as the OS.

5.4. Special Ergonomic Requirements

The special situation and the dangerous and critical context of an MCI do not only emphasize the existing conventional UI requirements described in the previous section, but it also introduces special ergonomic requirements. As mentioned before, the staff of the Fire Department TUM has been using the ruggedized tablet Xplore iX104 in their daily work and shared their experience with us. With a weight of approximately 2 kg, it is quite heavy and exhausting to use. To interact with the device they hold it in one hand and use the other free hand to interact with it with the stylus of the Xplore ix104. This is illustrated in Figure 5.1 b). The usage of the stylus is imposed by the fact that the existing UIs for the tablet PC were not suitable for the use by the finger, since the UI elements are too small and not adapted to the low resolution of a human´s finger. Although this way of interacting with the tablet is better than with a ruggedized convertible notebook (see Fig. 5.1 (a)), it's still not ergonomically optimal. The fire department staff complained that it is exhausting to hold the tablet PC with one hand. They also tried to find a practical hardware solution, for example to carry the tablet PC with a shoulder carrier bag. However, mounting the heavy device to a person was not accepted because it would hinder the person's body movement and flexibility. Another solution is to allow performing all available actions with the device while holding the tablet in both hands like shown in Figure 5.1 c). In fact, this thesis focuses on the challenge to develop a User Interface for mobile map interaction, which allows holding the device in both hands all the time while interacting with it. This makes the interaction with the device much more ergonomically acceptable and comfortable because the force of the weight is distributed among both hands making it feel less heavy and therefore less exhausting. However, to be able to provide such an ergonomic way of interacting with the device, the UI has to be completely adapted to this requirement. Just the thumbs of the user are able to touch the screen and their reaching distance is limited. As a consequence, each interactive UI element has to be placed in an area on the screen which can be reached with the thumbs in a comfortable way. The ideal case would be, if the hands do not even need to be moved up and down, so that each interactive element can be reached by just moving the thumb.

(a) Convertible Tablet: On one leg with both hands

(b) Standard Tablet: Using the stylus

(c) Standard Tablet: Held in both hands

Figure 5.1.: Three different styles to interact with a ruggedized tablet in a mobile context.

For the first evaluation of this concept we assumed that the user can reach a rectangle area on both sides. The bezels of the hardware are not symmetrical on the Xplore iX104, which makes the interaction areas not symmetrical as shown in Figure 5.2 a). The left side of the screen was reserved for interaction elements with a width of 19mm which is thinner than the recommended 22mm [55], [53]. For that reason, only elements not requiring high precision during interaction were available on the left side. The right bezel frame of the hardware is much thinner; i.e. more screen space is available. Thus, larger UI elements with a width of 44mm are positioned on the right border of the touchscreen. It should also be mentioned that we flipped the tablet by 180° to have the thin bezel side on the right side. This way, a rectangular shape with a width of 44mm was reserved for interaction elements on the right side which suits better for right-handed users. For left handed users the tablet could be flipped back to its original orientation. These values were initial estimations, but they were accepted by the target group. For this reason, we kept those distances with that device. We expected that the optimal position for the user interface elements are at the horizontal line going through the tablet PC center of gravity as shown in Figure 5.2. The optimal shape would be the almost-circular shape which can be reached with the thumb without moving the hand. But because of the extremely limited remaining space in that case, most of our UI elements are placed in the initially used rectangle area. We additionally developed a UI element which is capable of scrolling and zooming the map and also providing the selection of items on the map in a compact multi-feature element (see section 7.3 for details). This element can be placed in the optimal place which can be reached comfortably with the thumbs. Yet, providing a compact multi-feature element also increases the complexity of that element. Most recent studies also start dealing with the optimal position of those elements by studying the anthropometric data of human hands. Some of those studies are described in section 6.2.

(a) A) Ruggedized Tablet: Xplore iX104 (2kg)

(b) B) Ruggedized Convertible: Getac V200 (2.7kg)

Figure 5.2.: The two ruggedized tablets which have been used for developing and testing during SpeedUp

In the second half of the SpeedUp project, the ruggedized convertible tablet Getac V200 has been used for developing and testing instead of the Xplore iX104. Four reasons led to this decision:

1. Even though the touchscreen of the Getac v200 also is not as sensitive as touchscreens of currently available home tablets, it is much more sensitive than the touchscreen of the Xplore iX104.
2. The bezels of the Getac V200 are symmetrical.
3. The performance of the Getac v200 is much better than the older Xplore iX104 (in its first version - there are also new versions available), enabling us to provide better UI animations and designs.
4. A fast mobile text input mechanism has been rated as very important by the target group. Although we developed a promising mobile text input concept (see Section 13.5) it made sense to additionally perform our tests with a convertible notebook. If the additional thickness of the device and also the additional weight would be accepted by the target group, providing a convertible tablet instead of a standard tablet is reasonable and logical.

6. Related Work

In this chapter the related work dealing with digital maps are presented in Section 6.1. This is followed by a description of related work dealing with the two-handed interaction requirement in Section 6.2.

6.1. Digital Maps

Digital maps have a long history of development and research on different devices with different subfields. Some of them are: User guidance, information and map visualization, context-aware aystems, network connectivity, synchronization of data, the dimensions of the map (2D, 2.5D, or 3D), layer based maps, service oriented maps, and the user interface of the system. In 2005, Baus et al. published a survey in which they compared different properties of a representative set of map applications [10]. Till then, the focus of map application development was rarely the UI. According to Baus et al. the reason for that might be that most systems were still facing technical issues. This assumption seems to be true, because with the rapid development of mobile devices (such as smartphones and tablets) more and more attention has been spent to the UI of systems in general for all kinds of applications and operating systems. In fact, the quality of the interface is an important factor which determines the usability and the usefulness of a map application.

6.1.1. History of Maps

Cartography always has been an important issue for mankind. They help people to understand where they are and what is located around a specific location or next to themselves. Mankind started to create maps long before modern technologies such as satellites and electronic maps existed. In 2000, Rappenglück published a work describing the discovery of a rock picture in Spain which may represent the circumpolar constellation of the Northern Crown [83]. This map of the stars has been created at around 12,000 to 11,000 BC. BBC News reported about another even older discovery of a map showing the Summer Triangle constellation [11]. This map is thought to has been created around 14,500 BC. An example of an ancient map which might represent a location on earth has been published in 2006 by Meece et al. [68]. This drawing is considered to show a map of Çatalhöyük, Anatolia (today Turkey) and was created 7,000 BC. However, scientists are discordant about the correct interpretation of that drawing. However, many discoveries show that ancient maps existed indeed. Generally accepted as the earliest known map is surprisingly a mobile one. In 1930, an artifact was found 200 miles north of Babylon (today Iraq) [37],[107]. Figure 6.1 shows an image of that artifact.

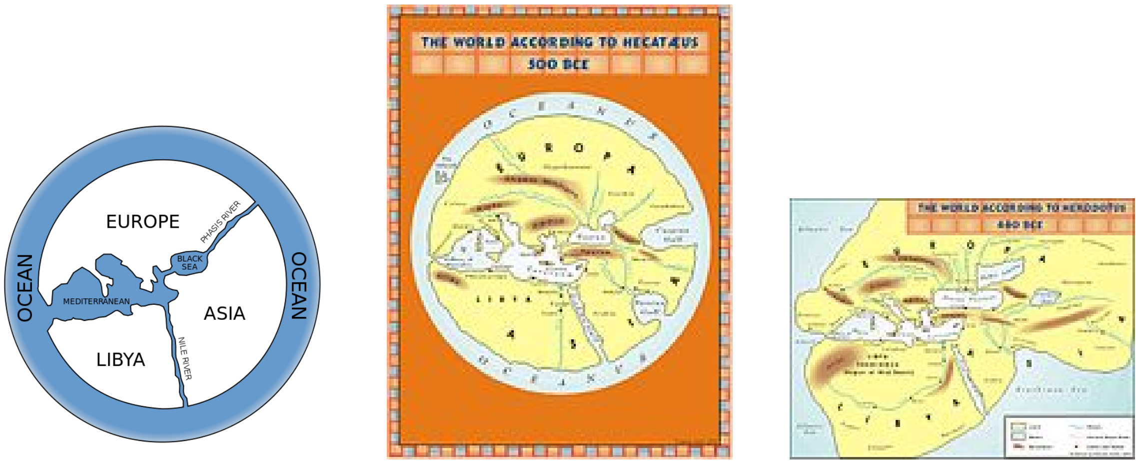
This artifact is in fact a tablet showing the unique map of the Mesopotamian world with Babylon in its center. According to the British Museum Magazine [30] it has a dimension

Figure 6.1.: Ancient Babylonian Tablet [36].

of 12.2 x 8.2 cm. It was created approximately in 2300 BC [107]. This map shows the Babylonian view of the world. Maps based on mathematical analysis and science appeared much later. Anaximander, an ancient Greek born between 611 to 610 BC, is considered as being the first geographer because he was the first who drew a map of the known world [49]. Figure 6.2 a shows an image of how this map could have been looked like [103]. Figure 6.2 b and c shows the world maps of Anaximander's successors Hekateaus (around 500 BC) and Herodotus (around 450 BC).

This shows mankind's efforts and demand for maps. With time, they became more and more accurate all the time improving with the existing knowledge (e.g. mathematics) new discoveries (e.g. territorial) and technologies (e.g. compass) of the respective time.

In modern time, a major change in the way of using maps happened when they became digitized and therefore lead to a new generation of map development in economics as well as in science. Digital Maps offered for the first time a way not only to gather information from a much more accurate and detailed map, but also to interact with it. Presented data can be filtered and manipulated and the map itself can be moved and zoomed. Additionally, it is possible to select different locations or items on the digital map to further interact with them. This new generation of map usage also lead to new efforts in science to improve them. Although decades of science have been spent on improving digital maps, they still get a lot of attention in many fields among researchers, but also among developers and users. The reason for that is that they are still adapting to the improving technology. While the first digital maps were stored on local magnetic storage devices, they are nowadays ubiquitous and get frequently updated. Also the way of interacting with them changed from the early command-style query languages used by experts only (1970s to the early 1980s) to the newer GUI based (Graphical User Interface) maps which can also be used by non-experts. Those GUIs were WIMP (windows, icons, menus, and pointer) based. While WIMP based GUIs are still valid and commonly used the rapid technology development in recent years again changed or at least introduced new user related requirements.



(a) Anaximander (610 BC) (b) Hekateaus (500 BC) (c) Herodotus (440BC)

Figure 6.2.: a) A reconstruction of the first world map of Anaximander [103] b) Hekateaus' world map [104] c) Herodotus' world map [104]

Smartphones and tablet devices offered (and/or required) users a new way to interact with them: the touchscreen. Additionally, they also became mobile. Consequently, while for the first digital maps, users were sitting in front of a computer at a safe place, people nowadays are able to use maps in every situation, even when walking or driving a car. Although the latter is not recommended, it shows an important change of requirements for today's digital maps on today's mobile devices. In fact, this is also relevant in the context of this thesis. While developing the digital map interface for the rescue service it is important to know in which context and under which circumstances the user is going to use the device. Since the users of the MCI map application are going to use the map application in a critical and stressful situation, this has to be considered while developing the user interface.

6.1.2. Characteristics of Human Interaction with Digital Map Applications

In 1999, Egenhoffer et al. [28] published an article which focuses on the analysis of the characteristics of human interaction with Geographic Information Systems (GIS). In this work three groups of interaction were classified: Interaction Paradigms, Interactions Modalities, and Interaction Styles. Table 6.1 shows these three groups and the components of each group, described by Egenhoffer.

The work of this dissertation mainly focuses on the browsing paradigm and uses text and graphics as modalities. Considering the interaction style, the work of this thesis places itself in GUI, WIMP, and direct manipulation. However, the latter is not completely true, because touchscreen based interaction styles are considered to be direct, because the user directly manipulates or interacts with the desired element by touching it. In our case, this is not possible due to the requirement to interact with the device while holding it in both hands. It is consequently a touchscreen style based on indirect manipulation. This is

Table 6.1.: The three interaction categories and the members of each group by Egenhoffer et al. [28]

Interaction Paradigms	**Interaction Modalities**	**Interaction Styles**
Querying	Text	Command-line input
Browsing	Speech and sound	GUI, WIMP and direct manipulation
Interviewing	Graphics	Filtering
Analyzing	Animation and Video	Delegation
Updating	Sketching	
Experiencing		

actually a mixture of WIMP and direct manipulation.

6.1.3. Multi-modal Map Interaction

One goal in the field of map development and research is to make the interaction with them more efficient. To achieve that, multi-modal interaction techniques have been investigated. This means, that instead of using one pointer (controlled by a mouse for example) to interact with the map application, two or more modalities to interact with the device exist. Cohen et al. [18] for example investigated the effect of using a pen and the voice together to control a map application for a military task in 2000. This dual-modality interface is called QuickSet and is compared against a standard GUI interface in an evaluation with 4 test users from the military. According to Cohen et al. the interaction speed increased 3 to 4 times, depending on the specific subtask.

Other interaction modalities can also be added or mixed with one another. One modality for example which does not occur in table 6.1 is the orientation of the device. By tilting a tablet or smartphone, the map can, for example, be scrolled without further interaction via the map's GUI. The idea itself is not new. In 1996 Rekimoto et al. [84] for example investigated this idea already for electronic hand-held devices. In 2004, Oakley et al. investigated the effect of additionally using vibrotactile feedback when scrolling with the tilting interaction modality. In this work, methods are described by which a tilting movement can be used and by which a vibrotactile feedback can be used to present scrolling related information. The authors of this work believe that the dual modality of using vibrotactile feedback together with the tilting of the device can improve the effectiveness of the interface. In 2010, Ramsay et al. [82] published a related study with the focus of the multimodal interaction when using field maps on mobile devices. They developed an application called SHAKE in which a map could be moved by tilting the mobile device. To interact with information, special gestures were defined. Besides others, these features and interaction modalities were compared with the same features on a standard map. This shows that multi-modal interaction is currently a topic of much interest.

However, using the tilting of a device was not an option in our case for the following two reasons. First, the ruggedized tablet devices used in our studies both were not equipped with the needed sensors to enable this feature. Second, using this feature contradicts the requirements of an application which is intended to be used in critical and unsure situations of an MCI. On the one hand, fine tilting movements require soft movements by the user. But soft movements cannot be expected when the user is in a hurry, for example, or needs to spent his/her attention on multiple field related and MCI related important processes. On the other hand, when exceeding a specific tilting degree, the user is not able to comfortably look at the screen anymore. This is the reason, why using tilting as a second interaction modality was not an option in our case. Using speech as an alternative second interaction modality was rejected in a discussion with the Fire Department TUM for similar field related reasons. During an MCI, there is usually a lot of noise in the area. This would hinder the speech recognition system to work reliably. This is, again, an important requirement for all equipment and every tool (thus also for the user interface) intended to be used during an MCI.

6.2. Two-Handed Tablet Device Interaction

In Section 5.4 the ergonomic requirement to interact with the device while holding it in both hands has been introduced. As mentioned in that section, the size and the shape of the interaction areas on the left and right bezel of the tablet were initially estimated and discussed in non-formal studies in UI expert groups. Because the size and shape of the interaction area worked very well in the first evaluation we focused on developing the single UI elements for interacting with the digital map application. In 2012, Microsoft published a study which explicitly focused on the reachability of thumbs [78]. The heat-map, published in the context of that study, visualizes areas which are comfortably reachable by the thumb (green area), not so comfortably reachable (yellow area) and also the maximum reachability (red area). Different groups of test users took part in that study: Small, medium, and large male and female. Thus, 6 groups participated in total. Additionally, two different ways of holding the device were tested. The first one is called "corner grip" and is not relevant in our case, because this way of holding the device does not consider the horizontal line which goes through the center of gravity. Since weight is an important factor in our case the second way of holding the device fits better. This is the so called "side grip"in which the hands are positioned more to the center of the borders. Figure 6.3 shows this heat-map which has the expected shape mentioned in Section 5.4.

The green area in the heat-map indicates not only the comfortably reachable area, but it also means that each participant of the study could reach that area. Because Microsoft's study was not published until 2012, the heat-map was not used in our studies (2010-2012).

Another study which focuses on two-handed interaction metaphors on hand-held tablet computers is BiTouch [100]. As in our setup, there is a difference between the use of the non-dominant hand and the use of the dominant hand. While the latter one is used for any interaction, as in a normal tablet UI, the former one is intended to be used to support the dominant-hand. Different features have been tested: Navigating a document, shifting to uppercase for text input and even zooming a map. A controlled experiment has been conducted to learn about the effect of tablet orientation and hand position. In a comparison, bimanual taps outperformed one-handed control in both orientations in that study.

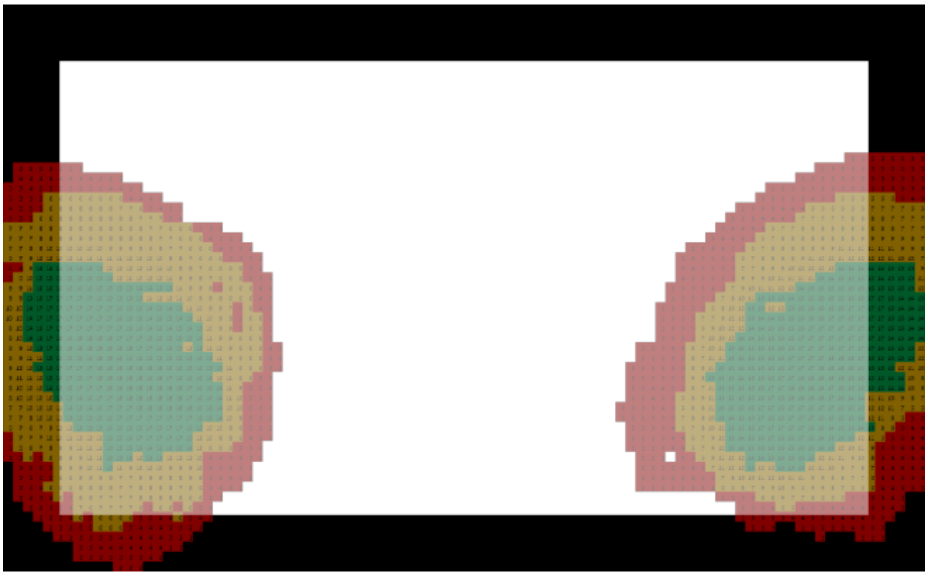


Figure 6.3.: Thumb reachability heat-map published by Microsoft in 2012 as a design guideline.

However, there is an important difference to our setup. Only one hand was intended to hold and carry the device. The other hand was free. In our setup, we want to distribute the heaviness of the device to both hands to make it more comfortable to carry it, while interacting with it at the same time.

7. Development of the User Interface

The three iterations of the user-centered development process which has been conducted in the context of this thesis are given in this section. As described in Section 3, the outcome of each iteration is the input for the following iteration. For this reason, each part of this section concludes with the results of the respective evaluation. The effect of the results on the development process is then given in the following section.

7.1. First Iteration

In the first iteration three different UI concepts have been developed to select items, three to scroll the map, and two for zooming the map in and out. In the first part of this section (Section 7.1.1) the selection techniques are described. The scrolling techniques are described in the following section (Section 7.1.2). The techniques to zoom a digital map in or out are excluded from this thesis because both of our initial suggestions for two-handed zooming were accepted by the users. For this we simply adapted well-known classic UI elements, a slider and a plus and a minus button, to the thumb-interaction by increasing their width. Details about the zooming UI elements can be read in [19].

7.1.1. Selecting Items on a Digital Map 1.0

The details of the first iteration on selecting items have been published at the workshop MoCoMed 2010 [21]. The UI of the map application has been developed gradually to increase the chance of its acceptance by the target group (AIO). The goal of this section is to find an answer to the question: **How can patients be selected?** There will be different items such as vehicles, special places, relief units or patients in the SpeedUp application. Since the UI has to be completely controllable from the bezel of the screen, items cannot be selected by directly touching them. Three different UI concepts which fulfill this main requirement are introduced in this section. All of them employ special UI elements (buttons) at the edge of the tablet PC screen where they can be reached with the thumb while holding the tablet PC in both hands. And since the space on the edge is limited, only a subset of all visible patients can be selectable at once. Therefore, how to determine this subset is the main topic of this section. To be able to compare the three selections methods with each other, all three should use the same buttons to finally select one patient among the determined subset. Those buttons are of type FishEye described at the end of this section.

7.1.1.1. Concepts to Select Items

The three investigated selection methods are: **Horizontal Line (SE-1.1)**, **Selection Quad (SE-2.1)** and **Automatic Mapping (SE-3.3)**.

Horizontal Line (SE-1.1) The first approach to select patients which are not directly reachable with the thumb is to draw a horizontal line from the user's right thumb position to the left hand-side of the screen (see figure 7.1). This line disappears when the display is not touched anymore. If the user slides the right thumb up or down the horizontal line follows it. The speed of movement is mapped one-to-one to the thumb's speed in order to make the movement more intuitive. Patients who are intersected by the line are members of the subset described before: Their ids are shown in a vertical list in the left bezel area, reachable and selectable by the left thumb. This way, each patient who is visible in the current segment of the map can be selected by moving the horizontal line with the right thumb and then pressing the left thumb on the specific id.

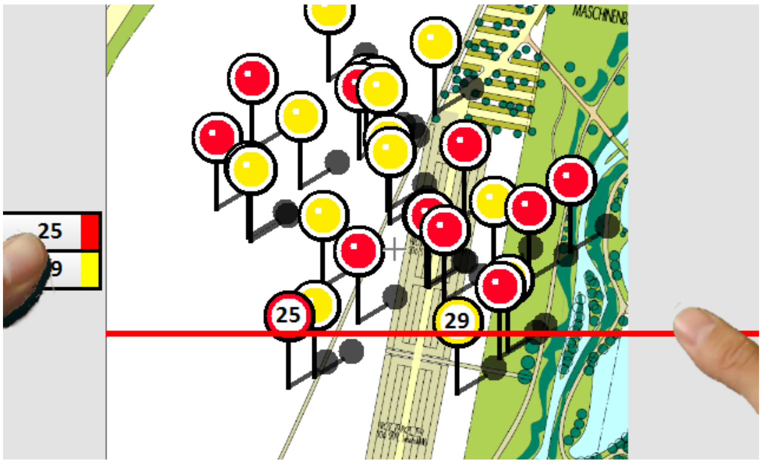


Figure 7.1.: Horizontal Line (SE-1.1)

Selection Quad (SE-2.1) The next approach is to draw a square on the tablet PC screen as shown in figure 7.2. Each patient inside this square is mapped to a member of the sub-list of patients on the left side of the tablet PC and can therefore be selected with the left thumb. The user is able to move this square by a graphical joystick. The graphical joystick has first been introduced in the context of this thesis in year 2009 (published in 2010 [19]). Thus, there were no prior experience with it in the community and this invention was completely new at that time. Nowadays, the virtual joystick is used in many games or applications on modern smartphones and tablets. In the contrast to the selection with the Horizontal Line, there is a risk that there are too many patients inside the square, so there has to be an option to rescale it. For this issue the joystick is extended with a plus and a minus button to control the size of the square. Even though this will increase the complexity of this selection method compared to the horizontal line, it also provides more flexibility to the user. Another advantage of this method is that the square can be used to scroll the map as well, if the screen´s edge is reached, by holding the joystick in the desired direction. With this technique the users can even zoom or center the map to the current square. In this first iteration, only the selection feature of the square has been evaluated to avoid mixing up several correlated factors.

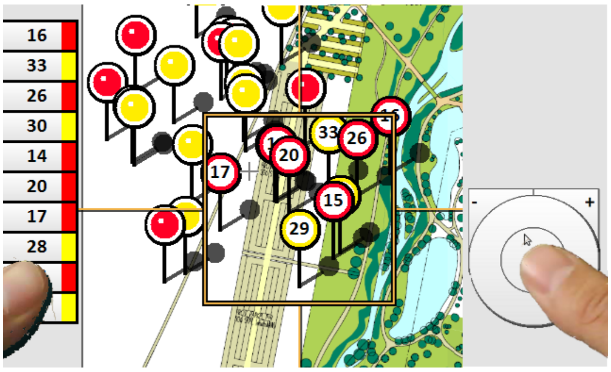


Figure 7.2.: Selection Quad (SE-2.1)

Automatic Mapping (SE-3.1) The last approach to select items on a digital map is to map all of the visible items to both edges of the screen. On the one hand an obvious advantage is that the user can select an item without the step of determining a subset first (see figure 7.3). This alternative provides a button to toggle the selection on and off which makes it easier and faster compared to the horizontal line (SE1) and the Selection Quad (SE2). On the other hand this method introduces some undesirable effects. Since each visible patient can be selected, each patient needs a correspondent button on the bezel which requires more space of the limited interaction area. Additionally, more buttons have to be searched before the desired patient can be selected. Consequently, there is a risk that the cognitive load will be increased using this technique.

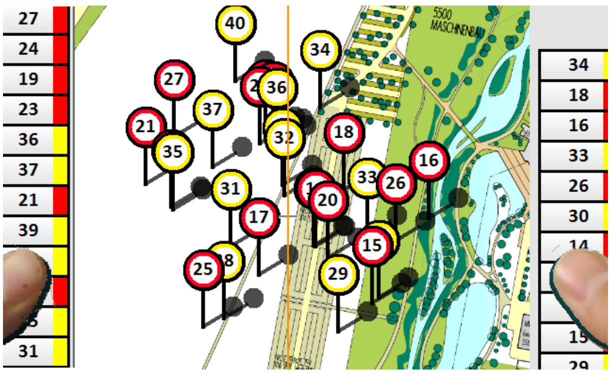


Figure 7.3.: Automatic Mapping (SE-3.1)

FishEye Buttons As previously mentioned, the limited space for the interaction at the bezel of the screen limits the number of buttons to be placed there. Additionally, special attention must be paid to the general requirement that buttons should not be smaller than 22mm [55], [53] in width and should be even larger for the scope of this thesis. We decided to use buttons with a bifocal technique described in [92]. In this technique the focus is set to a special point and a particular size of the space around this point becomes stretched. In our case the position of the user's thumb will be the focus point when it hits the desired button. The other buttons which are at a certain distance from this focus point are rendered a lot smaller. This does not reduce the accuracy because the button at this focus point is immediately larger when touched. So the limited available space is used more efficiently. In the scope of this work we are not evaluating the FishEye buttons, we just want to evaluate different selection alternatives to determine the subset of selectable patients.

7.1.1.2. Evaluation - Formative User Study

To develop intuitive, practical and powerful user interfaces, it is essential to work with future users of the system, the target group. The users themselves need something to obtain a realistic feeling about the application and its features. Therefore, the first step is to gather the users' initial requirements. After this step, we are able to develop different UI alternatives which satisfy those initial requirements. First, the participants of the first iteration are described, followed by a paragraph explaining the procedure of the evaluation. After that the results are presented and discussed.

Participants According to literature the most important problems of a UI will be learned from the first few test users. It is recommended to use between three to five test users, because more users will not necessarily improve the results of the evaluation [77]. Therefore, the system was initially evaluated by five test users. In this first test, those test users were not from the rescue service. This decision was made because the **initial** usability issues are the same for most people during the **first** evaluation of an UI, whether they are from the rescue service or not. Actually, it is more important to choose people with low experience in the field of human computer interaction in this first step. For that reason, people from different fields were randomly chosen: two biologists, a political scientist, a sociologist and a lawyer. The outcome of the first evaluation is then used to filter, improve or combine those elements depending on the user feedback and the quantitative results. The remaining enhanced and optimized UI elements will **then** be evaluated by the rescue service team (AIO) in the second iteration. All of the five test users (two females and three males) had low to average experience with computer interaction in general and with touchscreens in particular according to their own estimation. All of them were between 25 and 28 years old.

To find out usability issues and to be able to rate the different UI alternatives, the test users have to use and experience the UI elements. Therefore, it made sense to define the same task in all three alternatives to be performed by the users. The time duration to finish a task was measured for all users for each alternative. For each of the interface elements, the users fill out two questionnaires: SUS ((**S**ystem **U**sability **S**cale)) [14] and AttrakDiff [35]. After that, a short interview was held with each of the test users. In this interview the test users were asked how they felt and thought about the different UI techniques. This was important in order to find out the reasons for the scores of SUS and AttrakDiff and to be able to determine specific difficulties using these UI techniques.

Procedure During the execution of each task the test users were observed to detect some usability issues and clarify potential ambiguities faced by the test users. To compensate the learning effects, the sequence of the different alternatives were randomized (within-subject design, see section 2.3.2). During the interview phase, the test users gave feedback and shared their experience with the different UI alternatives. Finally, the test users were asked to rate and argue about the preferred UI alternative for selecting patients.

26 Patients were displayed on the digital map when the application started. Moving and zooming of the map were disabled. The patients' locations and triage states were the same for each alternative. The test users were first asked to select each red patient and then each yellow one with each UI alternative. Each selected patient disappeared from this test scenario. The task was completed when no patient was left on the map.

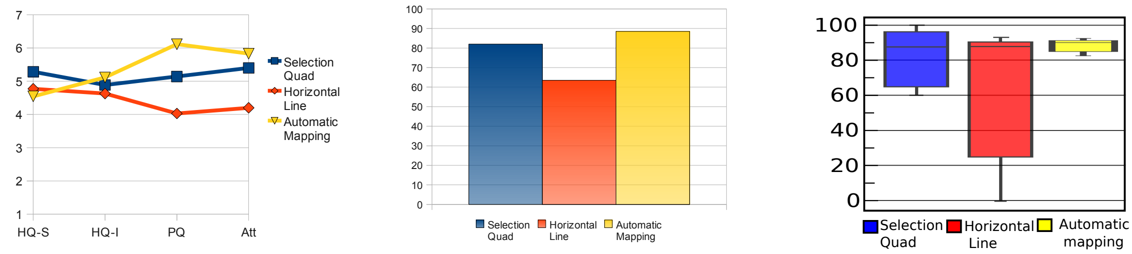
7.1.1.3. Results

In this section the results are presented and discussed for each UI selection alternative. In the first part, the results of AttrakDiff are recapitulated, while the following part summarizes the results from the SUS questionnaire. Finally, the outcome of the test users' feedback during the interviews are summed up and discussed.

AttrakDiff The results of AttrakDiff are separated into four dimensions (see Figure 7.4 (a)). The first two dimensions describe the hedonic quality of the UI and enclose stimulation and identity. For the stimulation part the Selection Quad (SE2) shows the best results while the Automatic Mapping has the highest score in the remaining three dimensions. The differences between UI alternatives are higher for the pragmatic quality (PQ) and the attractiveness (Att). Moreover, the order of the scores for the identity quality, pragmatic quality and the attractiveness respectively are the same. That means, the Automatic Mapping (SE3) has scored the best results in all three measurements followed by the Selection Quad (SE2) and finally the Horizontal Line (SE1). Nevertheless, the results are good for all alternatives since the score is never below four, but there are evident tendencies for the Automatic Mapping alternative.

SUS The results for SUS support the results from AttrakDiff (see figure 7.4 (b)). Actually, the usability of the Automatic Mapping alternative also obtained the highest score of 88.5 among the three alternatives. Figure 7.4 (c) shows clearly that the results of the Automatic Mapping alternative are very decisive since all test users rated this alternative in an interval between [82.5 ; 92.5]. Even though, the Selection Quad alternative obtained the second best score with a mean of 82.0, the test users rating was spread over a bit larger interval of [60.0 ; 100.0] compared to the Automatic Mapping alternative. Because of the number of test users it does not make sense to perform a significance test. Instead, we decided to qualitatively analyze the results. The reasons for those scores become clearer with the interview and the feedback of the test users.

Interview The interviews show that the reason for the low score of the horizontal Line (SE1) is that the line was too thin to intersect the patients easily. That is even more dis-



(a) AttrakDiff (b) SUS - Means (c) SUS - Boxplot

Figure 7.4.: Results for AttrakDiff and SUS

advantageous for the specific task to select all patients. However, if the specific task is to select one patient among a huge number of patients geographically, the horizontal Line (SE1) might score the best among the three UI alternatives. This also has been confirmed by one of the test users. The Selection Quad was not immediately understood by all users which make this alternative not as intuitive and practical as the Automatic Mapping. A better visualization of the functionality and the usage of the Selection Quad could improve its usability and practicality. Even though, the complexity of the Selection Quad is higher than the complexity of the Automatic Mapping, it also provides more flexibility to the user since scrolling and zooming can be integrated in a very compact way with no extra UI element. This benefit could not be seen by the test users during this evaluation because the scrolling and zooming feature of the Selection Quad was disabled.

7.1.1.4. Discussion of the First Iteration for Selecting Items on a Digital Map

During this evaluation, the Automatic Mapping was clearly rated as the best selection technique. But this result has to be regarded carefully, since this was an initial evaluation with a predefined task. This task is not representative for all possible real tasks in a real MCI, in our case the result of the evaluation is strongly bound to this task. Nevertheless, the evaluation points out important issues for all three alternatives. Investigating those issues and learning from the test users' feedback, can improve the proposed selection technique greatly.

7.1.2. Scrolling a Digital Map 1.0

We assume, that our users are used to different kinds of browsing through a digital map. A meanwhile well-known paradigm for touchscreens is to allow the users to drag the map by sticking the finger on the map. An advantage is that this paradigm is very intuitive, because the map sticks to the user's finger like a real paper-based map would. Additionally, the user has the full control of the speed and the direction of the movement. Thus, the user is able to move the map slowly and smoothly as well as fast and rough by moving the finger accordingly. However, this widely used method does not satisfy the requirement of interacting with the application only from the screen's bezel. In fact, the limited space at the bezel would require repeated short thumb movements to move the map. This would take too much time and will not only exhaust the users' thumb, but also lead to frustration after a while. Therefore, we invented three other promising techniques, considered to be appropriate for the thumb interaction. They are described in the following. This user study has been published in the proceedings of GMDS 2010 [19].

7.1.2.1. Concepts to scroll the digital map

The three investigated selection methods are: **Minimap Scrolling (SC-1.1)**, **Arrow Scrolling (SC-2.1)** and **Joystick Scrolling (SC-3.1))**.

Minimap Scrolling (SC-1.1) The first approach to browse the map was to put a miniature version of the map on the bezel of the screen. This is called Minimap and is shown in Figure 7.5. Tracked patients that are not inside the visible area of the map are indicated visually by a sign displayed on the corresponding side of the screen. This way, the user is immediately informed about the existence and the direction of these patients. However, the distance to each patient is not displayed on the real map to avoid information overload. For this reason, we make it possible to indirectly extract this distance from the Minimap. On the one hand the Minimap is visually comprehensive. The Minimap shows the whole area of interest and all patients. This way the user is automatically aware of the following information: The geographical position of each patient, the disposition of all patients, the triage states and the number of red, green, yellow and black categorized patients. Furthermore, the part of the real map which is currently rendered on the display is highlighted with a small square on the Minimap. The square helps for the orientation. On the other hand the Minimap is interactive. The user is able to jump to different positions with just one tap on the Minimap. Hence, a fast and rough positioning is possible. Moreover, the user is also able to move the map continuously while sliding the thumb inside the Minimap. Another advantage is the direct mapping of the finger's position on the Minimap and the position on the real map which makes it more intuitive. So there is a high chance that the users will feel comfortable with this approach. Finally, the Minimap needs less space on the screen. The Minimap concept was already known and has been used by some applications like games or browser map applications. The difference to the setup in this thesis is that the interaction will be done just with the thumb. The user does not have a high precision mouse-cursor combination. Fine movements with the thumb are challenging on touchscreens because their precision is even worse compared to the other fingers. We will learn during the evaluation whether the users are able to deal with the coarse movement or not.

Arrow Scrolling (SC-2.1) The second approach is straightforward. A button with an arrow is placed on each corner of the screen. For illustration, the upper two are shown in Figure 7.6. The arrow points the direction in which the map moves, if the user presses the button and holds it down. The map moves with a constant speed. Users can understand this method easily. But there are some disadvantages coming with this approach. The constant speed means that users are not able to determine the speed of the movement and then cannot move the map in arbitrary direction either. Only diagonal directions (45, 135, 225, and 315 degrees) are possible. Users have to zig-zag by switching between the diago-

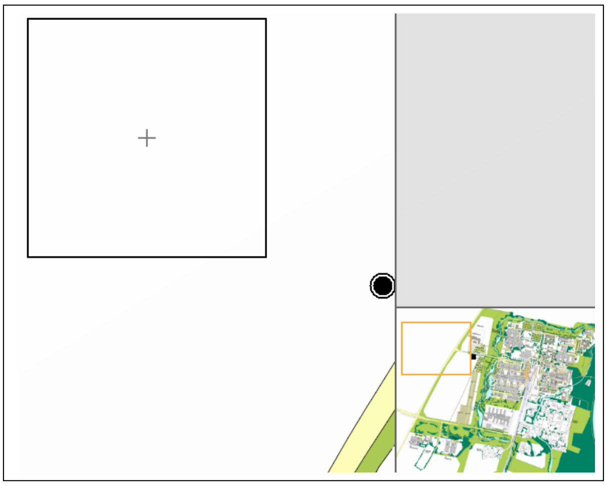


Figure 7.5.: Minimap Scrolling *(SC-1.1)*

nal directions to reach non-diagonal points. Since these concepts are designed for the use without multiple touch capability it is not possible to press two of them at the same time to define additional directions. Unfortunately, more than the four buttons would destroy the intuitiveness of this approach. A button which moves the map to the north has to be placed on the middle of the upper edge of the screen consequently. This is contradictory to the requirement that every UI element has to be reachable with the thumbs. Beyond that, the four buttons requires already a lot of space of the marginal interaction area down, so placing more simple buttons to the corners is not a good idea. Consequently, adding more buttons is not an option.

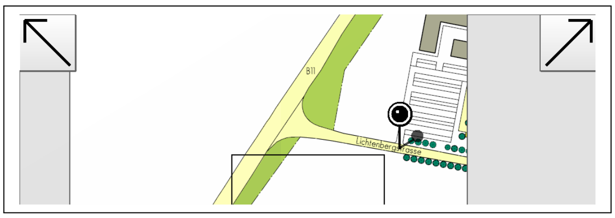


Figure 7.6.: Arrow Scrolling *(SC-2.1)*

Joystick Scrolling (SC-3.1) Our third approach to browse through the map is called Joystick Scrolling. A graphical joystick is used to control the movement of the map. Its functionality is basically the same as for a physical analogue joystick but there are some inherent differences. Visually, the graphical joystick looks like a real joystick from a bird's eye view (Figure 7.7). There is a graphical representation of the stick to give the user a visual feedback of the stick's current position. During Scrolling, the head of the stick will be occluded by the user's finger, but the stick itself will be observable. The functionality of this "softjoystick" is the same as the functionality of a real joystick and will be directly mapped to the movement of the map. When the users' thumb is on the upper-left side of the center of the joystick, the map is moved to the upper-left. The speed is mapped to the distance between the thumb's position and the center of the joystick. The radius of the joystick determines the maximum speed. This UI provides the full power of control for users since they can do both; control the speed and the movement to every possible direction. This functionality enables the users to move the map fast or slowly in any specific direction with a specified speed. Discrepancies to real joysticks lead to the following problem: The graphical stick does not provide haptic feedback. Like real joysticks, the stick bounces always back to the center when it is not held in the desired position. With a physical joystick, users become directly aware of this due to the tactile feedback. This is not the case for the virtual joystick.

Figure 7.7.: Joystick Scrolling *(SC-3.1)*

7.1.2.2. Evaluation - Formative User Study

Developing high quality user interfaces makes it essential to work together with potential users of the system. We arranged a user study to be able to rate the different alternatives and compare their usability. The first goal was to find out which of the presented alternatives is the most intuitive, efficient, attractive and accurate. Another goal was to find out potential usability issues and problems. Such knowledge enables us to reject, accept, or enhance the UI elements. This way, we can focus on the remaining elements for the following iteration.

The users need something to obtain a realistic imagination and feeling for the application and its tasks. To this end, the first step is to talk with them and define some basic requirements. We did this with people from the fire department TUM (Feuerwehr TUM) and found some special requirements, described in 5.4. After this discussion, the developer of the user interface was able to develop different alternatives which satisfy the requirements. At this point, the user is able to experience the different alternatives by using them in a specified task.

Participants We have evaluated the system with the same five test users described in Section 7.1.1.2. Two of them are female and three male and they were between 25 and 28 years old. Because of the reason described in Section 7.1.1.2 we decided to evaluate our system with people from different fields: Two biologists, a jurist, a political scientist and a sociologist. The optimized elements will then be evaluated with people from the rescue service in the next iteration. All of the five test users have low to average experience with computer interaction in general according to their own assessment. Additionally, they mentioned that all of them have low to average experience with touchscreens.

Procedure The procedure was the same which has been described in Section 7.1.1.2. The users also read the instructions before they started with the task. None of the elements were introduced to the users. The application logged the time each user needed to fulfill the task. Additionally, the accuracy was measured in case of the scrolling task. For each of the interface elements the users also filled out two SUS [14] and AttrakDiff [35]. After that a short interview followed.

During the execution of each task the subjects were observed by the supervisor to determine difficulties of the use and to guide the users if there are some ambiguities. A within-subject design has been used for this evaluation. To avoid biasing the results through learning effects the sequence of the different alternatives was permuted. After finishing a task with one of the tested alternatives, the users fill out two questionnaires: AttrakDiff and SUS. Afterwards, the subjects could write down their comments for each alternative. After using all elements they were asked to choose their favorite alternative.

The tasks of the users are described in detail in the following.

The Task Among others, we wanted to learn which of the introduced alternatives to move the map was the fastest one and which was the one with the highest accuracy. A quad was drawn at the center of the map and the goal was to move the map in a way that the currently shown patient was exactly in the middle of this quad as shown in Figure 7.8.

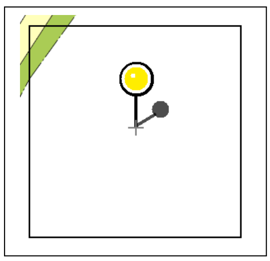


Figure 7.8.: Scrolling, perfect hit

The top priority of this task was to move the map as fast as possible in a way that the patient was positioned anywhere inside the black quad. The second priority was to place it exactly in the center. The application recorded the time users needed for each patient and the achieved accuracy as well. Seven patients appeared one after another in a predefined order. The position and ordering of the patients was the same for all alternatives and for all users to avoid differences in the results because of different positional distances between subsequent patients.

7.1.2.3. Results

The results of SUS for all alternatives are presented in Figure 7.9 as a box-plot on the right side. The left side is showing the results for the zooming part. As can be seen in that figure the results for zooming were very good for both presented alternatives (Buttons, Slider). Because of this, we stopped our efforts for developing zooming elements and focused on selection and scrolling. The results for AttrakDiff are shown in Figure 7.10. The results for the scrolling part is presented and discussed in the following. The collected data for the speed and the accuracy in case of the scrolling task are not mentioned here. The reason is that the users tried to place the map as perfectly as possible. Therefore, they didn't try to reach the patient as fast as possible which in fact was more important. Unfortunately, this misunderstanding falsified the results for speed and accuracy. We considered this issue in the case of the next iteration.

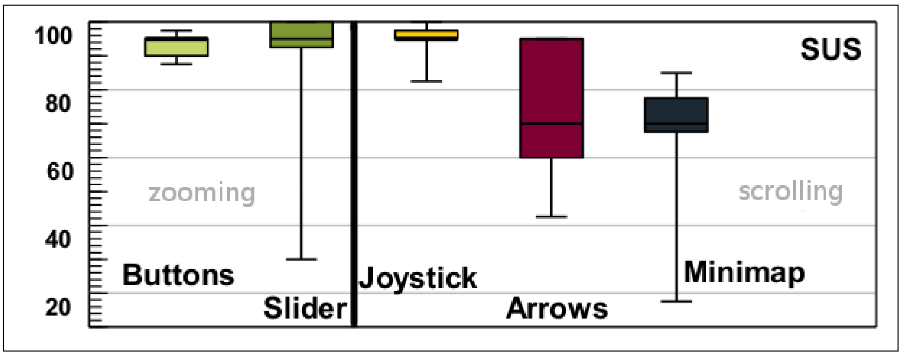


Figure 7.9.: System Usability Scale (SUS). The first two elements are those elements for zooming and the remaining three elements are those developed for selecting items.

AttrakDiff AttrakDiff results generally range from one to seven, with seven being the best score. The attractivity and the pragmatic quality of the joystick is rated very high (both above 6). The hedonistic quality is subdivided into stimulation and identification. For both parts the joystick has the highest score, too. Last but not least, four of the five test users preferred the joystick to scroll the map. One participant preferred the arrows. Although none of the participants chose the Minimap, all five test users mentioned, that they would like to use it if the positioning precision within the Minimap would be improved.

SUS When comparing the results of the SUS questionnaire regarding the three elements *Minimap (SC1)*, *Arrows (SC2)* and the *Joystick (SC3)* a clear tendency for the joystick alternative can be observed. The outcome of SUS ranges from 0 to 100, where 100 is the best score. Four of the five subjects rate the Joystick above 90 and the fifth subject's rating is still above 80. Consequently, the mean value of the joystick is very high with a score of 94.

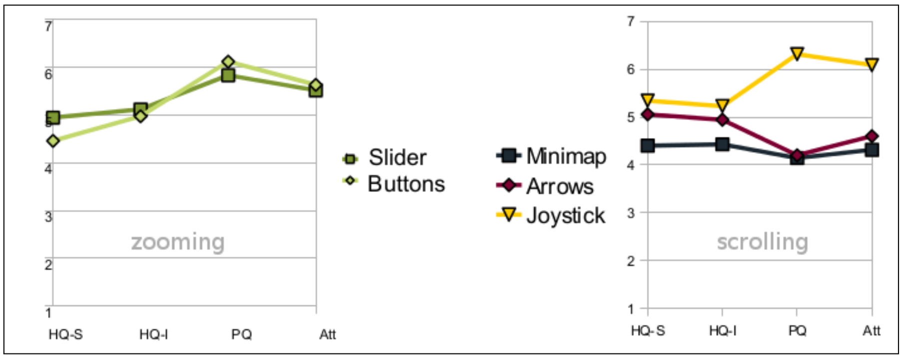


Figure 7.10.: AttrakDiff. The diagram on the left side shows the results for zooming and the diagram on the right side is showing the results for scrolling.

The results for the *Arrows (SC2)* and the *Minimap (SC1)* are more scattered even though the most of them are still high. The mean values are 72.5 and 63.5 respectively.

Interview The interviews with the users reveal the reason behind the quantitative results. The users were asked to place the map in a way that the displayed patient was positioned exactly in the center of the visualized square as fast as possible. The goal is to investigate the accuracy and the efficiency of each alternative. A picture of a perfect hit was shown to the users to help them to understand the task (Figure 7.8). Because of this picture the users payed more attention to accuracy than to speed. It made them try their best to move the map as perfect as shown in the instructions. For this reason, we decided to change this picture for the second iteration in a way that the patient was not perfectly centered, emphasizing that the test users should concentrate on the speed. Moreover, the participants need to be explicitly informed that the first priority is to execute the task as fast as possible.

Scrolling with the Minimap (SC-1.1) cannot be performed as precise with a thumb than with the mouse pointer. Because of this, the results of the Minimap are not as good as expected. But the users mentioned during the interviews that they like the overview feature of the Minimap. The arrows (SC-2.1) are declared to be very easy and intuitive to use. However, the lack of the possibility to determine the speed of the map's movement and the missing arrows to some of the directions was frustrating for four out of the five test users. The other one has not noticed that there are only the four diagonal directions to move the map. Since the joystick (SC-3.1) was the only element which provides the possibility to move the map fast and rough as well as slow and precise, the required task could be solved in a satisfactory way by using the joystick. The high value of the pragmatic quality (PQ) of the joystick in the AttrakDiff results confirms this assumption.

The results of the first iteration have to be investigated with diligence, because they are strongly coupled to the specific task which is not necessarily representative for all possible tasks in the case of a real MCI. As a logical consequence, more iterations and more evaluations are necessary to develop a UI which fits the tough requirements during an MCI. But although these results depend on the underlying tasks, this user study demonstrated that the users are able to understand the functionality of the joystick quite well, and that they are all willing to continue the scrolling task using the joystick to improve their results. In the following, some of the users' comments concerning the joystick are given:

- "good control and selection possible"
- "I had a lot of fun!"
- "Most comfortable of the used alternatives"

Even though the fun factor using the UI is not a primary goal of our usability research, it helps to increase the acceptance of the users, and therefore the use of the underlying system as well. The second iteration of the user-centered development procedure for the map interaction UI elements is given in Section 7.2.

7.1.2.4. Discussion of the First Iteration for Scrolling a Digital Map

This part of the first iteration explored three UI concepts for scrolling a digital map under the constraint to interact with a ruggedized tablet while holding it in both hands: Minimap Scrolling (SC-1.1, 7.1.2.1), Arrow Scrolling (SC-2.1, 7.1.2.1), and Joystick Scrolling (SC-3.1, 7.1.2.1). In this evaluation the virtual joystick was clearly rated the best scrolling method. The reason is that the users tried to precisely scroll to the target position as it was shown on an image printed on the task description (see Figure 7.8). The qualitative interviews revealed some weak points of each alternative as well as their strong points. The Minimap gives a great overview of the whole MCI area and gives the user the ability to quickly jump between different destinations. But it lacks the ability for precise scrolling. Using four arrows for scrolling was rejected by our test users due to its restriction of only four possible directions for scrolling. For this reason, it is not part of the following iterations. The Joystick got almost inverted results when compared to the results of the Minimap: It does not give any overview and does not allow users to quickly jump to a specific position on the map. But it allows precise and fast scrolling. For this reason, the goal of the next iteration is to improve both concepts in a way that their weak points are eliminated.

We evaluated the different UI alternatives with five participants that are not from the rescue service. The results of this evaluation proves that this decision was a good decision because weak points as well as the capabilities of all introduced UI elements could be identified. It hence enabled us to enhance the alternatives according to the results of this first iteration.

7.1.3. Conclusion of the First Iteration and Lessons Learned

In this section (Section 7.1) the first iteration of the user-centered development procedure of developing a UI for a map application has been described. This included different concepts for selecting items on a digital map and scrolling a digital map. For selecting, the Horizontal Line (SE-1.1, section 7.1.1.1), the Selection Quad (SE-2.1, 7.1.1.1), and the Automatic Mapping (SE-3.1, 7.1.1.1) were introduced. In our evaluation, the Automatic Mapping clearly outperformed the other two alternatives. But the analysis of the interviews revealed that the Automatic Mapping better fits the given tasks. Consequently, these results have to be regarded critically. Additionally, some weak points of the other two alternatives have been identified. For this reason, the goal of the second iteration for selecting items is to improve the Horizontal Line and the Selection Quad according to the test users feedback and to optimize the evaluation task to eliminate scenario specific advantages or disadvantages for the evaluated concepts. A more detailed description of the conclusion for selecting items is given in Section 7.1.1.4.

For scrolling a digital map, the following concepts have been introduced: Minimap Scrolling (SC-1.1, 7.1.2.1), Arrow Scrolling (SC-2.1, 7.1.2.1), and Joystick Scrolling (SC-3.1, 7.1.2.1). Scrolling the map with the four arrow buttons has been rejected by our participants. The quantitative results were better in the case of the Joystick. But the overview capability and the option to jump to specific positions on the map have been rated as very useful. An advantage of the Joystick is its ability to move the map smoothly and quickly. A disadvantage in comparison to the Minimap is that there is no overview of the map. Thus, in the next iteration, the weak points of both concepts needs to be eliminated and the evaluation task will be optimized according to the experiences made in this iteration. A more detailed description of the conclusion for scrolling a digital map is given in Section 7.1.1.4.

The results of this iteration can also be used for other applications and purposes. For instance, each introduced UI element may also be used for different applications on touchscreen devices such as tablets or smartphones which are intended to be used while holding them in both hands.

7.2. Second Iteration

In the previous iteration (described in Section 7.1) different UI alternatives for selecting items and scrolling a digital map have been developed, implemented, and evaluated with participants. All of the introduced concepts fulfill the requirements mentioned in Section 5. The goal of the second iteration therefore is to optimize the remaining concepts and to evaluate them with real users from the target group. The optimized concepts and the evaluation for selecting items are described in Section 7.2.1.The optimized concepts and the evaluation for scrolling a digital map are described in Section 7.2.2.

7.2.1. Selecting Items on a Digital Map 2.0

In the first iteration the Automatic Mapping introduced in Section 7.1.1.1 seemed to be the best alternative. It was the fastest concept and it also got the best subjective feedback from the participants. But these results need to be regarded carefully because of the following reasons:

- The previous evaluation was not performed with the real target group.
- The results of the previous evaluation presented trends but no significant results.

- The evaluation task of the previous evaluation was not representative for real tasks.
- Weak points of the other two alternatives (Horizontal Line and Selection Quad) were identified. Consequently, the results can change after eliminating the weak points of these alternatives.
- Weak points of the evaluation-task description itself could be identified and eliminated.

This user study has been published in 2012 in Coskun et al. [20].

7.2.1.1. Concepts to Select Items

The updated concepts and the changes to the previous iteration are described in the following section.

Horizontal Bar (SE-1.2) As a consequence of the first iteration the Horizontal Line has been changed to the Horizontal Bar. The test users of the first iteration mentioned that it was difficult to select patients with the Horizontal Line because it was too thin. The thickness of the new bar could now be defined by the user by touching and dragging a button labeled with symbols of arrows like shown in Figure 7.11. Other than that, the functionality remains the same like in the previous iteration: All patients inside the bar are added to the sub-list of selectable items and each patient got selectable by a button on the left interaction area. The bar could be moved up and down by moving the right thumb up or down while touching the display.

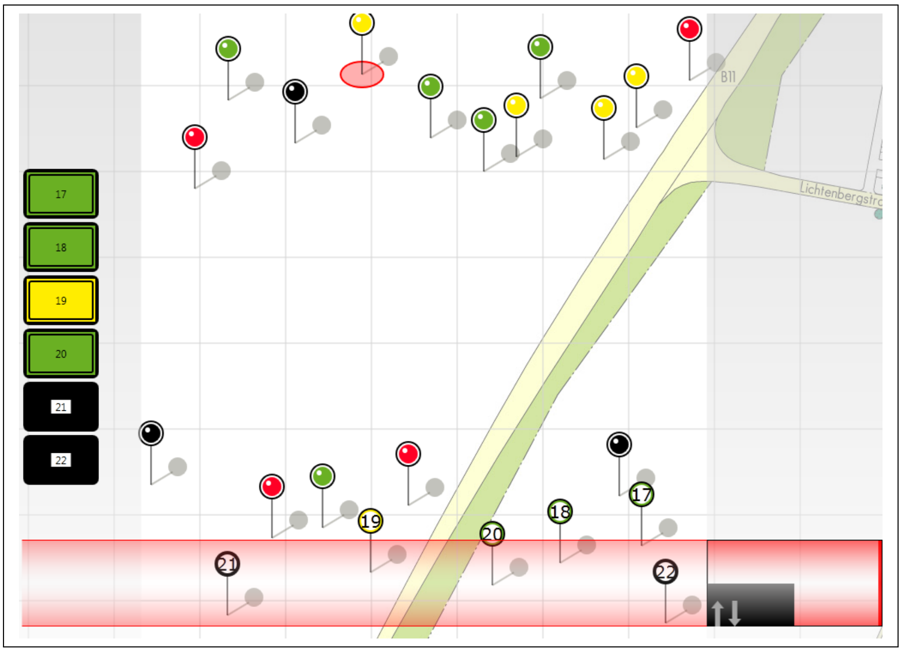


Figure 7.11.: Horizontal Bar (SE-1.2)

Selection Quad (SE-2.2) The Selection Quad was updated visually in the second iteration. The borders of the Quad have been changed to a decent black to avoid distracting the users from more important information on the application. The quad itself is now filled with a transparent light gray to emphasize that this is a specific area with a specific functionality. The functionality itself has not been changed. The quad is still controlled by a virtual joystick and the buttons are provided to let the user change the size of the quad. Each patient inside the quad is selectable. When the screen is not touched in the current interaction area, the quad simply disappears. The appearance of the Selection Quad in the second iteration is shown in Figure 7.12.

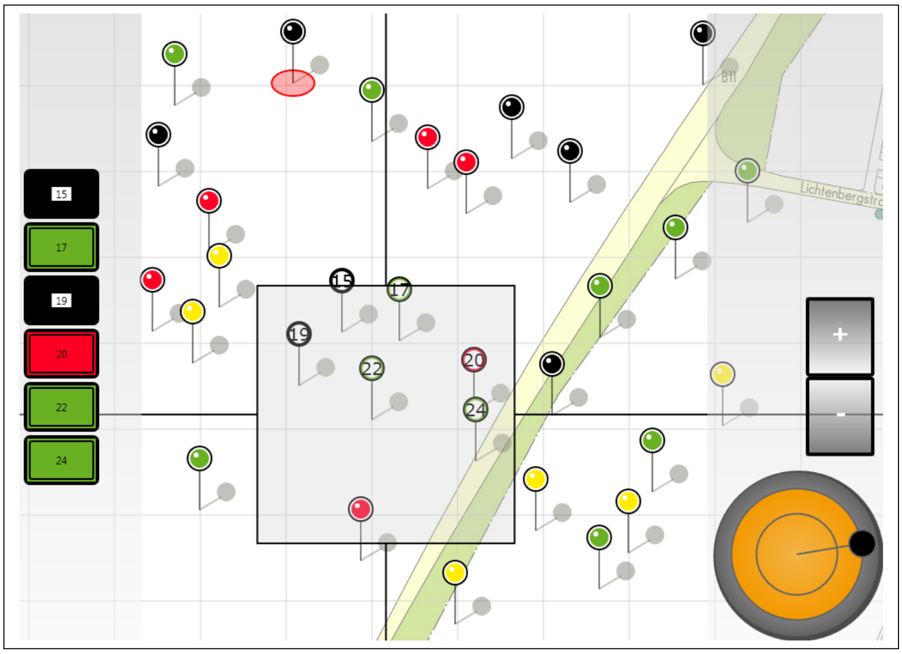


Figure 7.12.: Selection Quad (SE-2.2)

Automatic Mapping (SE-3.2) The Automatic Mapping has been rated as the best alternative among the presented selecting alternatives during the first iteration. For this reason, its functionality has not been changed for the second iteration. When pressing a button, all visible patients become selectable in either the left interaction area or the right interaction area. The user only needs to press the correct button. The version which has been used for the second iteration is shown in Figure 7.13.

7.2.1.2. Evaluation - Formative User Study

The previous evaluation revealed some weak spots in the concepts and showed some tendencies of the users but no absolute results. The Automatic Mapping was rated the best selection interface among the tested selection interfaces. However, the analysis and the interviews of the previous evaluation showed a clear influence of the defined task on the results. In fact, the Automatic Mapping was the best fitting selection alternative for the

Figure 7.13.: Automatic Mapping (SE-3.2)

given specific task and not the best selection alternative in a digital map in general. For this reason, we redesigned the tasks to highlight both: the advantages and the disadvantages of each selection alternative.

In the next paragraph, first, the design of the evaluation is described followed by the demographical data of the participants. Finally, the procedure of the evaluation is specified.

Evaluation Design The evaluation for selecting items in the second iteration is composed of the following three steps:

1. Preevaluation
   - Analyzing and discussing the results of the previous evaluation among a group of user interface experts.
   - Eliminating the shortcomings.
   - Improving the concepts and the evaluation design.
2. First Formal Evaluation
   - Evaluate the concepts with participants who are not members of the rescue service.
   - Optimize the task setup and the concepts according to the first obtained results.
3. Second Formal Evaluation
   - Evaluate the concepts with the **real target group (AIOs)**.

- Optimize the concepts according to the second obtained results.
- Learn from the feedback of the target group to improve the UI.

Procedure In order to determine usability issues and to be able to rate the different UI alternatives, the participants had to experience the UI elements. For this reason, we again defined tasks to be solved by the participants. We designed the tasks to deal with two aspects: 1) to represent a real task in the target group environment and 2) to accentuate as mentioned earlier the advantages and the disadvantages of each alternative.

The participants' task was to select geographically presented patients on a 2D map application with the introduced UI elements on the rugged tablet PC. This time each selection task was divided into 4 rounds. Each round was especially designed to emphasize the advantages and the disadvantages of each selection alternative. The participants had to solve each round with all the alternatives (within-subject design) in a random order to compensate the learning effects. The features to scroll or zoom the map were disabled to avoid influencing the results for the selection alternatives. The application automatically centered the map to the correct position for each round. The patients' positions were predefined for each round.

The user had to select only a subset of the presented patients for each round. The next patient to be selected next was marked with a red ellipse as shown in Figure 7.14. If more than one patient was marked, the participants could choose their own sequence. Thus, the user had to realize which patient to select using the current selection alternative and to add this patient to the selectable sub-list on the interaction areas of the tablet's bezels. The selection was then performed through soft buttons.

A message informed the participants whether a marked patient was selected or not. The task was resumed in both cases. In the failure case, the participant could immediately retry selecting a marked patient. Otherwise, the ellipse was removed and the participant could either continue with the next patient or with the next round. From the second round on, the participants had to execute multiple selection sub-tasks during each round. Those sub-tasks had the same positioning of the overall visible patients, but the distribution of the marked patients to be selected was either clustered or loosely distributed.

For each participant (in both formal evaluations), the application automatically logged the time needed to select each patient. The participants were also taped while solving the tasks on the tablet PC. This way, additional analysis could be conducted afterward. When a participant finished all four rounds with one selection alternative, they were asked to fill out the SUS questionnaire [14]. Finally, a recorded interview was conducted for each alternative.

The first round The participants had the opportunity to learn the respective UI element and the task to be solved. Four patients were randomly placed on the map current cutout.

Figure 7.14.: The patients' distribution for round 2 showing the active Automatic Mapping (Splitscreen)

The logging of the selection time duration was deactivated for this first round. We clearly informed the participants that the focus was not on the execution speed, but rather on understanding the selection alternatives. Once this was explicitly confirmed by the participants, the second round could be started. The participant could continue only if no misunderstanding remained.

The second round For the second round, a cluster of patients was placed on the upper left edge of the current screen's cutout as shown in Figure 7.14, depicting a realistic positioning of injured people in a real MCI. The main reason behind this setup was the goal to verify the assumption that the positioning of the patients influences the performance of the selection alternative. In the Automatic Mapping case for example, most of the visible patients will be placed on the left edge of the screen, since those patients are actually on the left half of the screen. This leads to a large number of soft buttons and hence the search task of the corresponding selectable patient button becomes complicated and its size becomes impractically small. As a result, the selection speed will suffer from this additional complication (*Assumption 1*). This is not necessarily the case for the Horizontal Bar and the Selection Quad, because of the user's capability to resize these elements. The evaluation will allow us to verify our assumption.

The third round For the third round the patients were spread out along the upper and lower edge of the screen as shown in Figure 7.15. The distribution of the patients on the left and right half was similar in terms of the patients' density. According to our experience from the previous evaluation and given these patients' distribution, we expected that the Automatic Mapping could be the most efficient alternative in this round (*Assumption 2*). A total of 22 patients were placed in round 3. Then, if the participant increased the size of the Selection Quad or the Horizontal Bar, a large number of the mapped buttons on the left side of the screen will be visible which again will introduce some additional complications. Based on our experience from the previous evaluation, we know that an automatic resizing of the Selection Quad limiting the selectable sub-list to a specific number, i.e. the patients' number within the quad or Horizontal Bar, confuses the users. Thus, in the current evaluation, manual resizing was enabled instead of the automatic one. We expect, that the resize feature of the Horizontal Bar and the Selection Quad would not be used frequently (*Assumption 3*).

Figure 7.15.: The patients' distribution for round 3 showing the active Automatic Mapping (Splitscreen)

As previously mentioned, the rounds were segmented into multiple sub-tasks. The positioning of the 22 patients was the same for each sub-task, but the marked patients to be selected by the participant were predefined to be either clustered or loosely distributed. The goal was to find out if the Horizontal Bar performs better than the Selection Quad in both distributions (*Assumption 4*). We expected that, because the Horizontal Bar moves in one dimension while the Selection Quad moves in 2 dimensions. This leads to the fact that the users only need to think about one dimension in the case of the Horizontal Bar which makes it less flexible but also less complex to use.

The fourth round The goal of the fourth round was to emphasize the disadvantage of the Automatic Mapping. In the fourth round of the first formal evaluation, 22 patients were loosely distributed on the screen, 11 were on the left half and the other 11 were on the right one. The first formal evaluation showed that this amount was not large enough to emphasize the disadvantage of the Automatic Mapping. For this reason, the number of patients was increased for the second formal evaluation to 32 (see Fig. 7.16). To be able to place a button for each patient on the screen edge, the size of the buttons was accordingly reduced. Thus, we expected that the Automatic Mapping will be the worst in terms of performance in this case compared to the other 2 alternatives (*Assumption 5*).

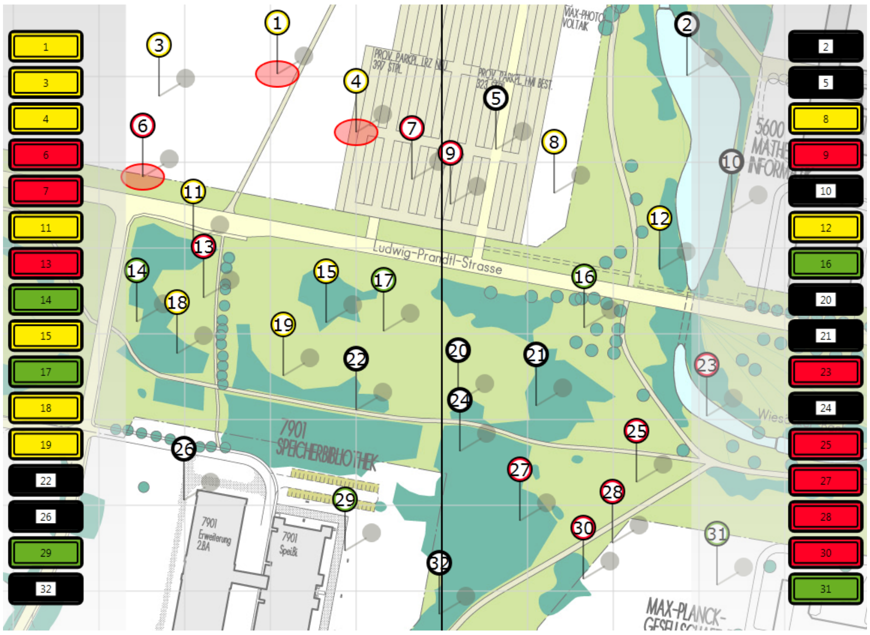


Figure 7.16.: The patients' distribution for round 4 showing the active Automatic Mapping (splitscreen)

Participants As already mentioned, this study consists of three evaluation steps. The participants used for each evaluation are described in the following paragraphs.

1. Preevaluation The informal pre-evaluation has been performed among 5 User Interface experts. The optimized UI elements were discussed as well as the new evaluation tasks. As a result of these discussions, the final task setup was defined and the application was ready for the first formal evaluation.

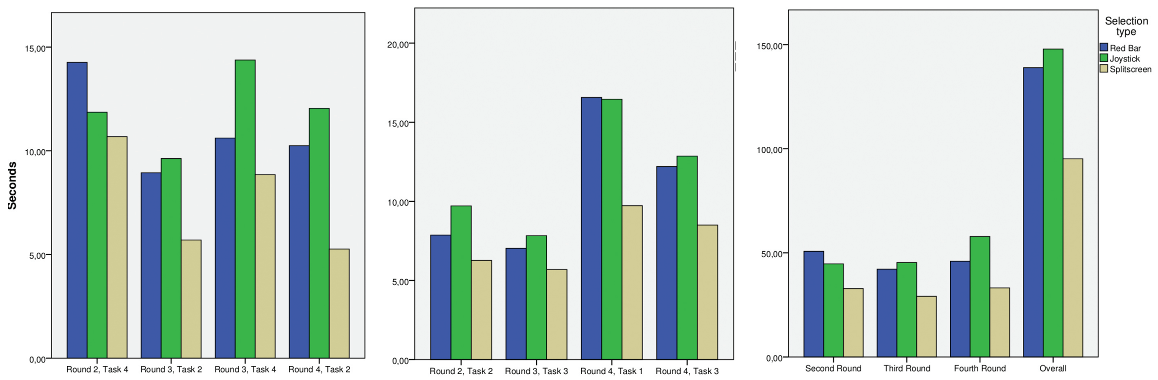
2. First Formal Evaluation To make sure that the evaluation procedure was well designed, we first performed the evaluation with 5 students (4 male, 1 female). All of them were between 23 and 25 years old and had experience in using touch screens on smart phones, and were right-handed. The goal was to evaluate the introduced UI elements independently of the rescue service to evaluate the general usability of the developed UI elements. Again a within-subject design was used.

3. Second Formal Evaluation The second formal evaluation was performed by five **A**mbulant **I**ncident **O**fficers (AIO) from the **A**rbeiter-**S**amariter **B**und (ASB) Muenchen, i.e. the target group, they are all males aged between 27 and 46 (36.4 in average) and right-handed. Four of them confirmed having experience with touchscreens. But we were more interested in their experience as an AIO. Consequently, the feedback of the second group is a prerequisite to develop a UI which increases the acceptance of the SpeedUp system. We used a within-subject design for this group too.

7.2.1.3. Results

The results can be classified in three categories: (1) Automatically logged data (speed and clicks), (2) Results from the SUS-questionnaire and (3) Subjective feedback from the interviews.

Logged Data The mean values of the time that the participants needed to solve the tasks are illustrated in Figure 7.17 for the first formal evaluation and in Figure 7.18 for the second one. In both figures, diagram (a) shows the mean values for the sub-tasks where the patients were clustered while diagram (b) represents the results in the case of loosely distributed patients. Diagram (c) depicts the mean values per round. The overall time duration for all tasks is then summed up in Figure (c). The logged data show that for almost all setups the Automatic Mapping (split screen) was the fastest selection alternative, independently of the patients' distribution. For instance, the AIOs needed a mean time of 160.1



(a) Clustered patients (b) Loosely distributed patients (c) Results per round

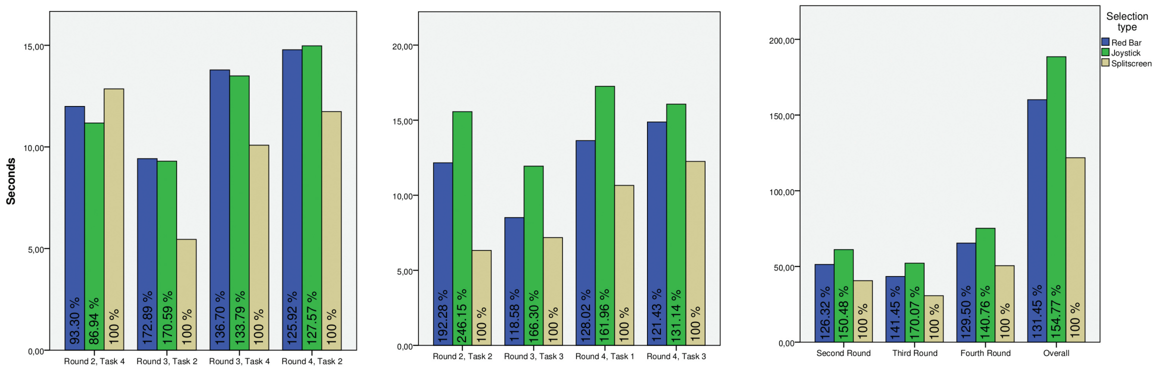
Figure 7.17.: Time Mean values (sec) of the different evaluation setups in the first formal evaluation

seconds and 120.2 interaction clicks to solve all tasks with the Horizontal Bar, a mean time of 188.5 seconds and 114.2 interaction clicks in the case of the Selection Quad (controlled with the joystick), and a mean time of 121.8 seconds and 83.4 interaction clicks in the case of the Automatic Mapping (split screen). The mean number of the interaction clicks per round can be obtained from Table 7.1. For all rounds and for both groups, the Automatic

Round	G1	G2	G1	G2	G1	G2
2.	29.4	38.0	35.4	33.6	25.2	26.6
3.	39.0	30.4	34.2	28.8	21.5	20.0
4.	34.6	51.8	41.2	51.8	25.6	36.8
$\sum$	112.4	120.2	110.8	114.2	72.4	83.4
	Horizontal Bar		Joystick		Split Screen	

Table 7.1.: The interactions (clicks) needed for each round and selection alternative. G1 = Group of first formal evaluation; G2 = Group of the second formal evaluation (AIOs)

Mapping had the lowest mean values among the three selection alternatives. Based on the logged data, the Automatic Mapping seemed to be the fastest selection alternative.



(a) Clustered patients (b) Loosely distributed patients (c) Results per round

Figure 7.18.: Time Mean values (sec) of the different evaluation setups in the second formal evaluation (AIOs)

SUS - Questionnaire The SUS scores of the first and second formal evaluation are illustrated in Figure 7.19 and Figure 7.20 respectively. For both groups the outcome of SUS for all presented selection alternatives is distributed mainly between 60 and 80 points. Our values are within the average of the SUS values studied by Tullis at al. in 2008 [97]. Considering the first group, the SUS values of the Automatic Mapping were spread out mainly above 90. According to Tullis at al. [97] a SUS score of 90 puts us in the top 5% of their SUS samples. This means that the users of the first group clearly preferred the Automatic Mapping. However, the SUS score of the second group, the AIOs, is smaller than the one of the first group. Each AIO rated the Automatic Mapping below 80. There even was 1 AIO, who rated it below 50. The joystick controlled Selection Quad and the Horizontal Bar scores were comparable in both groups.

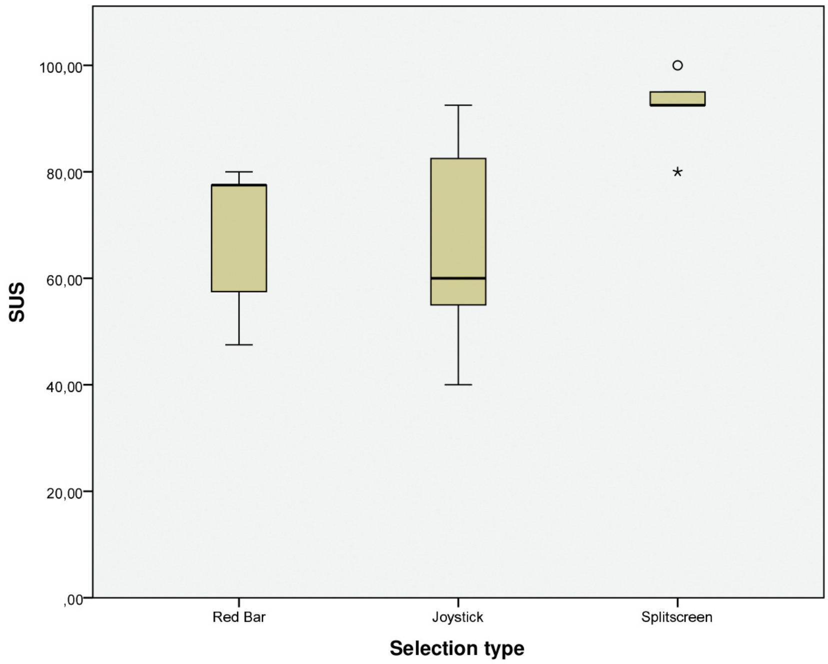


Figure 7.19.: SUS scores for the first formal evaluation - performed with students

Interview In the case of the Automatic Mapping, one participant of the first group suggested splitting the screen adaptively and not always at the half. The adaptivity should ensure that there were always as many buttons on the left as on the right side of the screen. Another suggestion was to visualize the mapping between the buttons and the corresponding patient for example with a linking line. One of our participants was confused by the appearing/disappearing-animation of the Horizontal Bar. An AIO hinted that for mental reasons he would prefer to have just 1 list of buttons instead of 2, as in the Automatic Mapping, since it required less mental effort to search 1 list instead of 2. For the same reason he preferred the Horizontal Bar: The bar was moved in one dimension which made it natural to sequentially process all patients from the top to the bottom. The last worth mentioning suggestion made was to develop a new selection alternative considering the operational sector of a real MCI. Thus, only the patients of the currently selected operational sector should be mapped to the sub-list.

7.2.1.4. Discussion of the Second Iteration for Selecting Items on a Digital Map

Our first assumption was that the performance of the Automatic Mapping decreases when the number of displayed patients increases. The quantitative results could not confirm that, since for both the clustered and the distributed patients with small (22) and large (32) number of patients, the Automatic Mapping performed best overall. The SUS values of the first evaluation group (computer scientists) show trend favoring to the Automatic Mapping. However, the second group (the AIO) was not completely satisfied with it. The interviews revealed that the reason for that was that the AIOs were associating the pre-

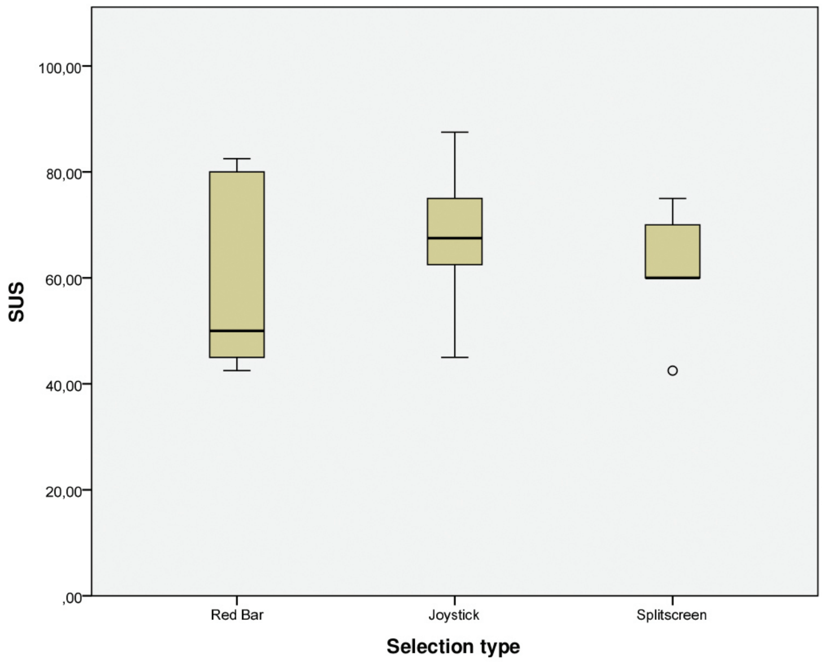


Figure 7.20.: SUS scores for the second formal evaluation - performed with our target group (AIOs)

sented selection alternatives with a real MCI. And that is exactly the motive for second formal evaluation. Although they needed less clicks and less time to solve the tasks with the Automatic Mapping, all of them agreed, that in a real MCI it is more likely that an AIO selects one patient to interact with, for example to add and collect some data from the patient, instead of selecting all of them consecutively. Overall, our quantitative results show that the Automatic Mapping was the most efficient selection alternative. Consequently, there seems to be a trade-off between efficiency and acceptance by the target group in this case. One AIO mentioned, that he prefers the Horizontal Bar for the following reason: Since all involved people during an MCI are in an extreme and dangerous situation, he prefers doing one task at one time and concentrate on that. With the Horizontal Bar it is possible to process the task step by step from top to bottom. This is like scanning the map vertically in a checklist. This may be a strong positive psychological impact of the Horizontal Bar.

Our second assumption was that the Automatic Mapping performs best in round 3. This was true for all of our results. Thus, the Automatic Mapping is very efficient if there are not too many patients. This confirms what we expected.

Our third assumption was that the feature to manually resize the Horizontal Bar or the Selection Quad will not be used frequently. The video analysis showed that half of our participants used the resize feature. But in most cases, they just set the size initially and then did not change it further. Our interviews and our observations could not point out any confusion due to the resize feature of the Selection Quad. Even though our assumption was true for this study, some of the participants mentioned that they did need this feature.

Assumption 4 was that the Horizontal Bar performs better than the Selection Quad, when the patients were clustered or loosely distributed. This assumption was indeed true in the second group (AIOs) for clustered patients. But the differences between the results of the Selection Quad and the Horizontal Bar were too small to confirm this assumption. For the first group this assumption was not always true. Concerning the loosely distributed patients, even though the results of the second group would confirm this assumption, there were no significant results for the Horizontal Bar, neither through the objective results nor through the subjective interviews and observations.

The last assumption was that the Automatic Mapping will perform worst in round 4, since both edges of the screen were full of buttons and the participants had to spend some effort finding the corresponding button that represented the patient to be selected. Although we increased the number of patients between the first and the second formal evaluation to stress test the worst-case scenario for the Automatic Mapping, this assumption could not be confirmed through the quantitative results. However, the feedback from the AIOs on the Automatic Mapping highlighted some mental effort and work-flow shortcomings: searching 2 lists instead of 1 increase the mental load of the user and should be avoided.

As far as efficiency is concerned, which in fact is a crucial requirement for electronic devices that are intended to be used in MCIs, the Automatic Mapping seems still to be the best choice among the presented alternatives. However, efficiency is not decisive if the target group does not accept the UI. This also confirms the fact that a user-centered development design is very important, since the results from the target group can differ entirely from the results of any other group. We had further discussions with 3 of the AIOs to be able to find a solution for this trade-off. The outcome of this informal discussion is an additional assumption: If the evaluation task includes more than just UI elements for selecting items (scrolling and zooming), the advantages of the Automatic Mapping could be reduced while the advantages of the Selection Quad might be emphasized. This assumption is based on the fact that the Selection Quad is actually more than just a feature to select items. It additionally allows to scroll the map, center the map to a desired position and even to zoom it in or out. These features were excluded in the current study, since we focused on the selection part to avoid any influences and to be able to have the same conditions for all selection alternatives.

7.2.2. Scrolling a Digital Map 2.0

In the previous iteration, the following three UI alternatives have been developed and tested: **a)** 4 Simple buttons on each edge of the screen, **b)** an interactive minimap and **c)** a virtual joystick. In summary, the virtual joystick performed best. However, by critically analyzing the qualitative interviews, it was clear that the reason for this good performance was the low precision of the Minimap. While Minimaps work great for applications which are controlled with the mouse and keyboard, they do not perform well on touchscreens. This is especially the case in our setup, since only the thumbs are free to interact with the Minimap. But our users also mentioned a disadvantage of the joystick compared to the Minimap: The joystick does not offer an overview visualization of the patients' position. One simple solution might be to provide both, a joystick to interact with and a Minimap to visualize important data and their positions on the map. But since there is were limited interaction space on the bezels of the screen we decided to eliminate the weak points of both concepts and conduct another evaluation. The new concepts and the changes are described Section 7.2.2.1. This is followed by the description of the evaluation in Section 7.2.2.2. The results of this evaluation are presented and discussed in Section 7.2.2.3. The details of this user study have been published in the proceedings of GMDS 2012 [26].

7.2.2.1. Concepts to Scroll a Digital Map

In this section the enhanced concepts for the Minimap and the Joystick are described.

Minimap 2.0 The disadvantage of the Minimap according to our previous evaluation is the low precision. Consequently, we added the following feature to the Minimap: By touching and holding the thumb on the Minimap (200 ms) the behavior of the Minimap changes such that it works similar to the joystick. The initially touched position is the center of this joystick functionality. The position of the thumb on the Minimap is then no longer mapped 1 to 1 to the real map. Instead, the map scrolls exactly the same way, as it scrolls when using the joystick. The same happens if the user touches the rectangle inside the Minimap, which indicates the current position of the map section. This is logical, because there is no need to jump to the new position because the new position is close to the current position. If the user touches an area inside the Minimap but outside this rectangle, the Minimap keeps its original behavior and jumps immediately to the new position. To reduce the risk that users will not understand this multi-functionality of the Minimap, we adapted the visualization: As soon as the Minimap swaps to the joystick mode, four arrows appear which are scaled individually according to the thumb's distance to the initial touch position (see Figure 7.21 second row, right).

Radar-Joystick 2.0 The weak point of the joystick was that there was no visualization showing the patient's positions. For this reason, we added a visualization feature to the joystick which works similar to a radar. The relative positions of the patients are shown inside the background of the joystick. This way, patients who are not inside the current map section and therefore not visible on the map are visible inside the Radar-Joystick. The missing visualization was mentioned as the main disadvantage of the previous version of the joystick while the usability was rated quite high. Figure 7.22 shows both the graphical representations of the Radar-Joystick when it is not touched and when the user is interacting with it in the left and the right figure, respectively. The interaction itself was rated as intuitive and good in the first iteration. For this reason, no changes have been made to the way of interacting with the joystick. There are 2 areas inside the joystick: The inner precise area and the outer fast area. Those 2 areas are visible in figure 7.22. The scrolling speed

Figure 7.21.: The Minimap not touched on the left side and touched on the right side

increases with the distance of the thumb to the center of the joystick. While it increases 1:1 inside the inner area it increases 1:2 inside the outer area.

New features for both concepts We also faced some problems that originated from the ruggedness of the used device. The touchscreen is not as sensitive as modern home tablets nowadays are. Sometimes, the touchscreen loses the contact to the finger even though the finger is still touching it. For this reason, we added a short time delay of 50ms in which touch up events are just ignored. Thus, if the system loses the contact to the finger for a short period of time, the user does not even notice this. On the other side, as another consequence, the map now keeps scrolling for 50 ms, when the user intends to release the finger. But since this is almost not noticeable, we kept this feature in the current version of our concepts.

Another outcome of the first evaluation was the well-known occlusion problem on touchscreens. For this reason, we added the clone-feature to both concepts: As soon as the user interacts with either the Joystick or the Minimap, a cloned view pops up above the UI element. This cloned view is still visible while the UI element itself is occluded by the finger (see Fig. 7.21 and 7.22).

7.2.2.2. Evaluation - Formative User Study

In the following paragraphs, first the selection of the participants of this study is described. This is followed by the explanation of the procedure of the study. Afterwards, the evaluation apparatus is presented.

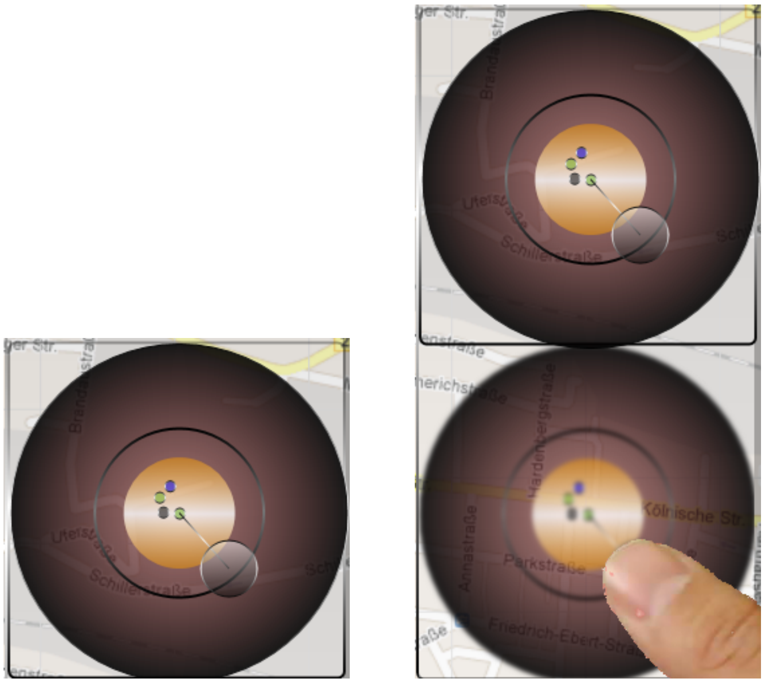


Figure 7.22.: The Radar-Joystick not touched on the left side and touched on the right side

Participants As for the second iteration of selecting items, this time 4 people from the target group (AIOs) participated in the evaluation of the second iteration for scrolling a digital map when using the thumbs only while holding the tablet in both hands. All of the test users were male and right-handed with a work experience of 26.25 years in average. They were between 35 and 52 years old. Three of them used a touchscreen and a map application before and one had no experience with touch screens at all.

Procedure The evaluation setup consisted of two rounds. Each round was partitioned into three sessions. The sessions were designed to emphasize the advantages and the disadvantages of both concepts. Each session consisted of multiple patients and a different distribution on the map. The goal was to scroll the map to the shown patient using one of the two concepts. Once the map was centered to the patient, the next patient appeared until the system reached the end of the test round. The participants were explicitly asked to scroll to the patient **as fast** as possible and **not as accurately** as possible. To keep the accuracy fixed, the entire evaluation was divided into two rounds dealing with different accuracy settings. Both rounds had to be solved by our participants in each session involving both concepts. Thus we had 12 executions of rounds per participant (Round 1-ABC and Round 2-ABC with both concepts). A semi-transparent circle positioned at the center of the screen was shown to the user. The goal was to position the map such that the patient is inside this circle. In the first round, the participants had to perform the three sessions with a large circle (very low accuracy) and in the second session the size of the circle was reduced, thereby increasing the needed accuracy to position the patient inside this circle. Since all participants used both concepts, we randomized the sequence of the concepts for each user to discount learning effects (within-subject design).

The following paragraphs describe and explain the differences between the three sessions. The distributions of all three sessions are shown in Figure 7.23. However, this is just to illustrate the distribution of the patients in this dissertation. The participants never saw the overall distribution. Just the next patient which should be found by the user was shown either inside the Minimap or inside the Radar-Joystick. After completing a session with one UI concept, the participants were asked to fill out the standardized SUS questionnaire [14]. Additionally, a short guided interview was conducted to get qualitative data and professional feedback from the AIOs. Afterwards, they repeated the same procedure with the other concept. The results of both our qualitative and quantitative results are presented in Section 7.2.2.3.

Session A Task A consists of seven patients which are positioned closely to each other (clustered patients). The reason behind that is to emphasize the advantage of the Radar-Joystick. Based on our experience in the first evaluation we expect the joystick to perform best in this session during both precision settings (The rounds). We also expect that the result of the Minimap gets worse in the second round since the needed accuracy increases (smaller target circle). Because of that, we expect that the performance of the Minimap suffers more in Round 2 than the performance of the joystick. The distribution of the patients in Session A is shown for both concepts in Figure 7.23 on the left side.

- A1: Joystick performs better than Minimap in Session A in both rounds.
- A2: The Minimap performs worse in the second round.
- A3: The performance of the Minimap suffers more when more precision is needed than the joystick.

Sessions B For Session B the distribution of the seven patients was increased. Thus, the included visualization of the patients' positions inside the Radar-joystick and the Minimap has to be used in this case to be able to find the patients. We expect that there will be no relevant difference between both concepts in this task. This distribution is shown for both concepts in Figure 7.23 at the center.

- A4: There is no relevant difference in speed for both concepts in Session B.

Session C In Session C the whole map was used to distribute the seven patients. Thus, to fulfill the task, the participants have to scroll the map from one edge to the other for each patient. Because of this fact, we expect that the Minimap will perform faster in Task 3, even for Session 3 in which more accuracy is needed to complete the task. This distribution is shown for both concepts in Figure 7.23 on the right side.

- A5: The Minimap will perform better in Session C for both rounds.

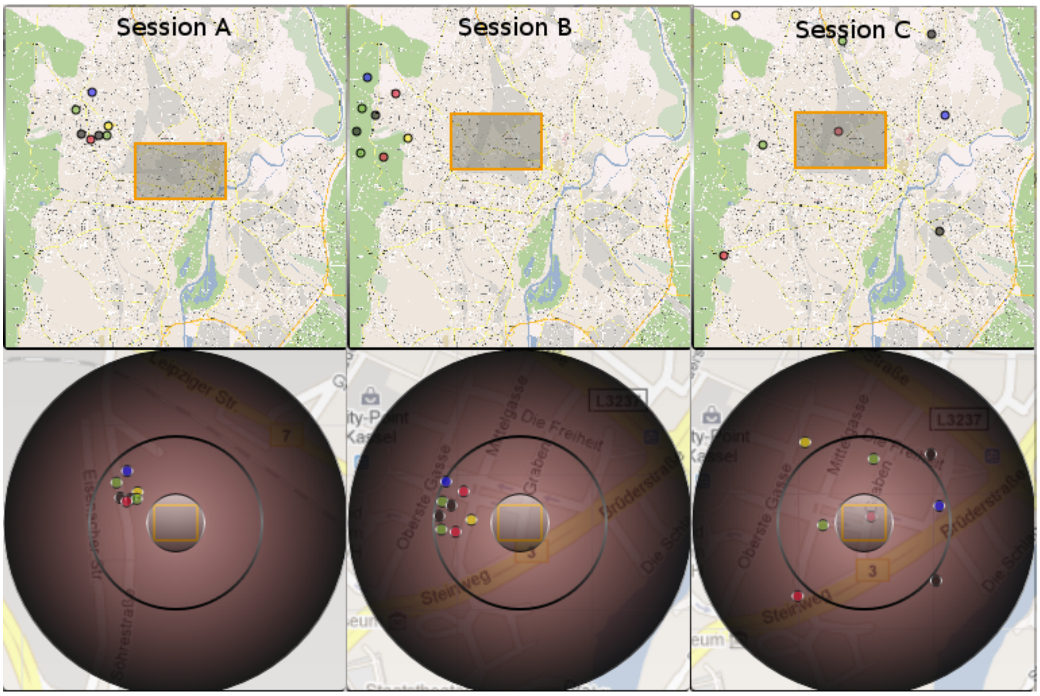


Figure 7.23.: Both concepts and their patient visualization in all three sessions.

Apparatus As a hardware, the convertible rugged tablet PC Getac V200 has been used. As being a convertible rugged notebook the weight is even higher than the weight of standard rugged tablets. Another disadvantage of the device is its width of 5cm. But in the same time, compared to other tablets of the time in which this study has been conducted, it has a high performance equipped with an Intel I7 processor and 8 GB of RAM. Another advantage is that the touch display is much more reliable and sensible than other rugged tablets like the XPlore 104, which we used in the previous evaluation. Still, the display is still not as reliable and sensible as the tablets which are used at home. During our evaluation, Windows 7 was running on the Getac V200 and the whole application has been written in WPF and C#. Figure 7.24 shows the device.

7.2.2.3. Results

The results can be classified into three different main categories:

1. Automatically logged data (process-oriented objective data),
2. Results from the SUS-questionnaire and preferred interaction metaphor (subjective quantitative data) and
3. Subjective feedback from the interviews (subjective qualitative data).

In the following, these three categories will be described separately.

Figure 7.24.: The rugged convertible notebook during the evaluations: Getac V200

Automatically Logged Data To be able to compare the efficiency of both concepts the application logged the time needed to fulfill each task for each participant.

The first diagram in Figure 7.25 compares the speed of the Joystick and the Minimap for all sessions in Round 1. In this round, the participants did not need to be very precise when scrolling to the target. In Session A the participants needed approximately the same time with both concepts (Joystick 3% faster than Minimap). In Session B the distances between the targets were increased and the participants did need 15% more time to solve the whole session with the Minimap (85,43s) than with the Joystick (74.06s). This tendency changed in Session C, in which the scrolling targets were distributed over the entire map. In this session, the Minimap (93.60s) was 17% faster than the Joystick (113.11s).

The second diagram in Figure 7.25 shows the same data for Round 2. The participants had to be more precise. For this reason, the participants generally needed more time with both concepts to fulfill the same tasks as in Round 1. However, the Minimap suffered more than the Joystick. This can be seen for example in Session B: In Round 1 the Minimap needed 15% more time than the joystick. The difference increased in Round 2 - Session B to 37% (Minimap: 113.91s, Joystick: 83.41s). Again, the Joystick outperformed the Minimap.

The third diagram compares the sums of all means for all sessions per round. This means, the diagram shows the results when the amount of short and long scrolling distances are mixed. The differences in speed for both concepts reduce in this case to 2% and 4%, respectively.

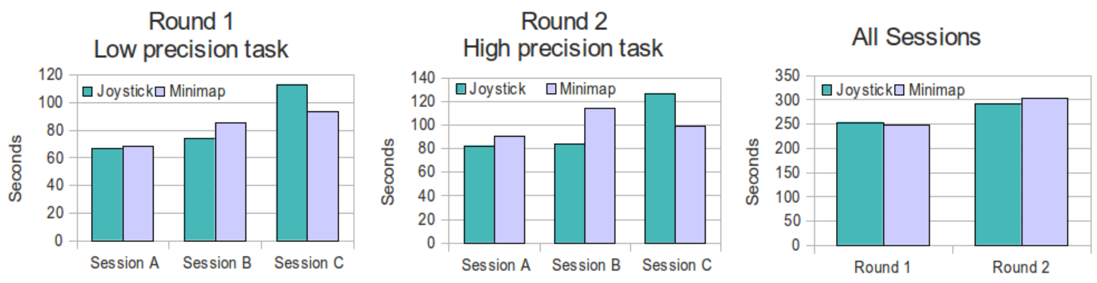


Figure 7.25.: The needed time in seconds to solve each session in Round 1, Round 2, and for all sessions per round

ID	Assumption	Result
A1	Joystick performs better than Minimap in Session A in both rounds	yes
A2	The Minimap performs worse in the second round	yes
A3	The performance of the Minimap suffers more when more precision is needed than the joystick	yes
A4	There is no relevant difference in speed for both concepts in Session B	no
A5	The Minimap will perform better in Session C for both rounds	yes

Table 7.2.: Our initial assumption and whether they are fulfilled or not.

Table 7.2 summarizes our assumptions and whether they are confirmed by our quantitative data.

SUS-Questionnaire and Preferred Interaction Metaphor The SUS scores of the two different selection alternatives (Joystick vs. Minimap) of our formal evaluation are illustrated in Figure 7.26 on the left side. The mean SUS score of the Joystick is 75.0 and therefore higher than the SUS score of the Minimap with a mean of 55.6. According to Tullis et al. [97] the score of the Radar-Joystick is positioned at the upper bound of the average SUS scores which appear in literature. Thus the usability of the Radar-Joystick can be considered as high. The SUS score for the Minimap is below the lower bound of the average SUS scores studied by Tullis et al.. Thus the usability of the Minimap is not good enough to be suggested as a concept for interaction purposes on touchscreens which are used with the thumb and in which the participant cannot handle precisely while interacting. Additionally, our participants were asked to rate the two concepts with a six item Likert-Scale (see Figure 7.26 right side). A value of 0 means: “I would never use the concept in future” while a value of 6 means “I would definitely use this concept in future”. The Minimap mean value considering this question is also lower than the value of the Radar-Joystick, thus supporting the results obtained by the SUS questionnaire. However, it is still interesting that although the SUS-score of the Minimap is quite bad, the Likert value of the Minimap is still above three. If people rate the usability of a concept badly but also say that they might use this concept in the future, this could mean that they liked at least some aspects of the concepts but had difficulties to use it. The results of our qualitative interviews, given in Section 7.2.2.3 also try clarifying these reasons behind the subjective results.

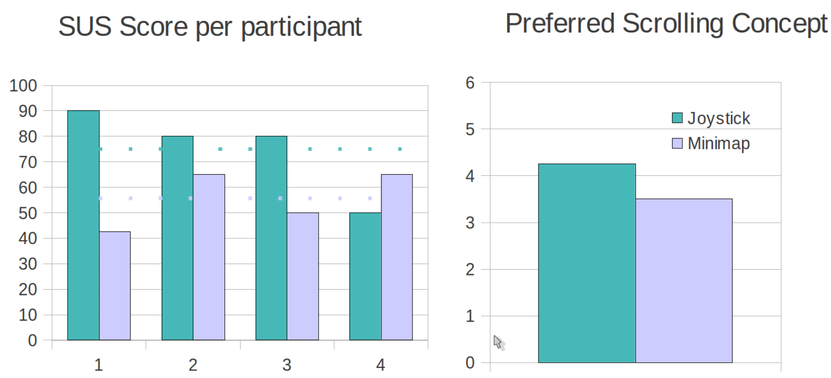


Figure 7.26.: System Usability Scale (SUS) for each participant on the left side and the subjectively preferred concept on the right side

Qualitative Interviews To be able to understand the reasons for the scores of the concepts and to learn which features worked well and which are problematic, qualitative interviews with the participants are indispensable. Overall, the qualitative interviews confirmed the results of the SUS questionnaire. The participants argued that the Radar-Joystick was much easier to use and more intuitive. In contrast to the Minimap, the participants did not need further explanation to understand its functionality. The major reason why they needed help understanding the Minimap functionality was the added dual functionality: If the user holds the finger on the screen instead of just tapping it or if the thumb initially touches the Minimap inside the rectangle showing the current cutout of the map, the Minimap switches to a Joystick mode to be able to do fine adjustment when scrolling. This feature was added to overcome its precision problem. However, this feature was not immediately understood by our participants without further explanation. Thus, the intuitivity clearly from this added feature. The usability also suffers from the duality, because the participants sometimes activated the fine adjustment feature although they wanted to perform a tap and vice versa. This was the main source of frustration in our evaluation which consequently led to the low SUS score of the Minimap. The reason for the participants expressing the will to still use the Minimap in the future can be due to its visualization part and its ability to quickly jump to the desired position on the map by tapping somewhere outside the rectangle. The visualization clearly gives them an overview of the whole incident area, the distribution, and triage status of the found patients. And the quick-jump feature enables them to quickly jump to any position.

7.2.2.4. Discussion of the Second Iteration for Scrolling a Digital Map

The outcome of our second iteration is clearly that the Minimap helps getting an overview of the whole situation and significantly increasing the speed of scrolling. However, our attempts to enhance the Minimap to also be able to fine adjust the scrolling confused our participants. Thus, the duality presented in our Minimap 2.0 concept should not be used. Instead, we suggest enabling only the feature to immediately jump to the desired position on the map by tapping the Minimap at the wanted position. But we also suggest including the Radar-Joystick. Since in the suggested setup both UI elements are provided, the size of the visualization of the Radar-Joystick can be reduced in order to safe rare screen space. It does not need to visualize the distances to all patients, since this can also be seen on the Minimap. Instead, the Radar-Joystick should only visualize the distances to patients, who are close to the current position of the tablet PC user. But this setup would use more space on the edge of the screen, which is actually another disadvantage. For this reason, another outcome of the second iteration is to equip the Minimap with the Joystick. The Joystick then fades when not touched by the user. Just a small circle will be visible to the user indicating that there is further functionality. Once this small circle is touched by the user, the Joystick will be opened and immediately enabled to be used by the user. And as soon as the user releases the finger from the Joystick/screen, it automatically collapses or fades again. In our next development, we will provide this setup and compare it with setups, in which both concepts are just provided separately. If the performance with the integrated and the separated concepts are approximately the same, the merged version can be used without slowing the user down. This is described in Section 7.3 which deals with the third iteration of the development procedure.

7.2.3. Conclusion of the Second Iteration and Lessons Learned

On contrary to our expectations, the Automatic Mapping (Section 7.2.1.1) again outperformed the other two presented alternatives when analyzing the quantitative results. However, the target group (AIOs) rejected the Automatic Mapping. The reason for that is that the Automatic Mapping overloads the screen with too much information which is an essential disadvantage. This could not be seen by the participants who were not from the target group in the previous iteration. Additionally, the AIOs mentioned that although the Automatic Mapping is faster than the other two alternatives single items will not be selected that often. This means, that small differences in speed might be less important, since delays do not accumulate so much when the feature is not frequently used. Nevertheless, it is an important feature to have, but it is more important to focus on comfortability, intuitivity, and to provide a structured and high quality overview about the available MCI related information.

The following qualitative advantage of the Horizontal Bar (Section 7.2.1.1) has been mentioned by one of the AIOs: With the Horizontal Bar the task can be solved from top to bottom which is like scanning the map vertically such as in a checklist and therefore this participant preferred the Horizontal Bar. This may be a strong positive psychological argument for the Horizontal Bar. However, the following advantage of the Selection Quad could not be seen by our participants: The Selection Quad also allows scrolling and zooming the map with one single UI element. Therefore, it saves limited screen space and leads to less movements with the hand when all features are activated. Because we focused on these features (scrolling and selecting) separately this advantage was not visible. Thus, the preferred concept could change if all features (selecting, scrolling, and zooming) are activated making the hidden advantages of the multi-feature Minimap visible to the user. For this reason, the goal of the third iteration is to perform an evaluation with tasks in which all features are enabled together and are all necessary to solve the given task. A more detailed description of the conclusion for selecting items can be read in Section 7.2.1.4 and for Scrolling in Section 7.2.2.4.

7.3. Third Iteration

In both previous iterations we focused on one of the standard map interaction features (selecting, scrolling, and zooming) while the remaining features were deactivated to avoid confounding the results due to too many options for achieving the given tasks. On the one hand, this was important to ensure that the results were not influenced by other factors. On the other hand, it also hid some aspects which could have changed the qualitative results of the UI elements. For example, the Selection Quad was intended to be a multi-purpose UI element which could be used for all of the three standard map-interaction features together. This makes it a powerful and space-saving UI element. But since it was only used to select items in the previous iterations, our users were not able to observe that advantage. The previous evaluation setup made sense in order to focus and rate the single UI elements but an integrated evaluation in which all features are enabled and needed to solve the evaluation task is essential to learn how the UI elements perform when they are used together.

7.3.1. Concepts of the Third Iteration

The third iteration consists of two contrasting setups: **A - Separated UI Elements:** The UI elements which have been rated best in the previous iterations have been chosen to be part of this first version of the integrated setup; one UI element for each map-interaction feature. **B - One Combined UI Element:** The best aspects of each alternative are chosen and integrated into a new solution which combines all advantages that have been identified in the previous iterations into one powerful and space-saving UI element.

7.3.1.1. Separated UI Elements

The idea of the first concept of the third iteration is to provide the most intuitive interface by choosing the UI elements which have been rated best by the target group in the previous iteration. Consequently, there is one UI element for each map-interaction feature. The whole interface is shown in Figure 7.27.

To **select** items, the Horizontal Bar (number 1 in the Figure) (introduced in Section 7.2.1.1) has been integrated into this alternative. The Horizontal Bar is only visible when

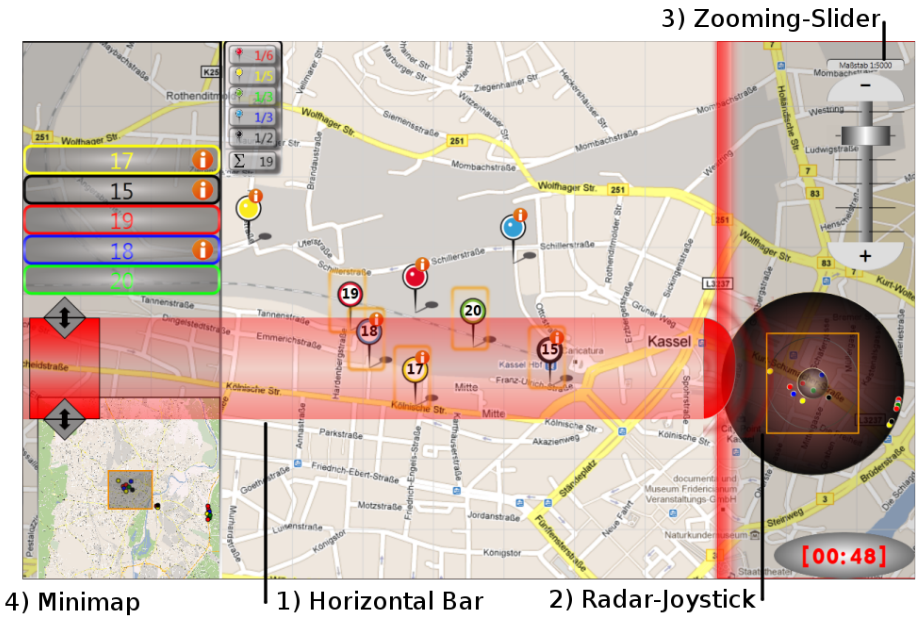


Figure 7.27.: Integrated Alternative 1: Separated UI elements

the user is touching the red vertical rail on the right bezel of the screen. The Automatic Mapping has been rejected by the target group as being inappropriate in the context of a real MCI. The Selection Quad moves in two dimensions while the Red Bar only moves in one dimension which has been rated as being less complex. For those reasons, the Horizontal Bar is the logical choice.

To **scroll** the map the Minimap (number 4) **and** the Radar-Joystick (number 2) have been chosen. The overview of the Minimap has always been rated as being important and the target group also affirmed that. Unfortunately, the idea of integrating two modes into the Minimap, one to jump to locations and one to precisely scroll the Minimap with an integrated joystick-function confused the users in the second iteration and it did not work well on the rather insensitive screen of the ruggedized tablet. For this reason, the scrolling-mode of the Minimap was disabled. But the user could still use it to jump to specific locations of the map. When precise scrolling was needed, the joystick could be used.

To **zoom** the map, the slider (number 3) of the first iteration was chosen. Both UI elements for zooming were accepted well by the users of the first iteration. Thus, we simply chose the more powerful one which is obviously the slider.

The advantage of this first alternative is that each integrated element has exactly one function which makes them less complex to use when compared to a multi-function element. Additionally, each UI element received good quantitative as well as qualitative results in the previous iteration. The probability that this interface will be understood immediately and can be used efficiently therefore is high. But these advantages are obtained at the cost of screen space. Thus, it makes sense to develop also a multi-function UI element (Combined element) and to compare it to the alternative with separate controls. This is explained in the following section.

7.3.1.2. One Combined UI Element

As mentioned earlier, the separate elements need a lot of space. This also means, that not all elements can be placed in a location which can be comfortably reached with the thumbs without the need to move the hands up or down. When using a combined element which can be used to select items, to scroll the map, and to zoom it in or out, this element can be positioned ergonomically optimized in order to be used with one thumb. The solution used for the third iteration as the second combined interface concept can be seen in Figure 7.28.

Figure 7.28.: Integrated Alternative 2: Combined UI element

To support **selecting** items, the concept of the Selection Quad (number 1 in the Figure) has been integrated in the combined element. This quad can be moved by using the Radar-Joystick (number 2). Once the Selection Quad reaches an edge of the screen it starts **scrolling** the map in the direction given by the joystick position. The Radar-Joystick is vertically squeezed to reduce the size needed by the element when being not active. When touched, the Radar-Joystick becomes active and is vertically scaled up to become a circle again. At the same time, the Minimap (number 4) and the buttons (number 3) are moved upwards to avoid being occluded by the Radar-Joystick. The elements moved upwards then are no longer reachable with the thumb. But this is good, because since the user has only one thumb on one hand, it is not possible to interact with the Radar-Joystick and another element on the same side simultaneously. When the user releases the Radar-Joystick it is squeezed again and the upper elements move down. Then the Minimap can be used again to jump to specific locations and the buttons are also reachable again. This context-change is animated. When the Radar-Joystick becomes active, the animation is very fast (100ms) to avoid slowing down the user. When it becomes inactive the animation starts very slowly and after 200ms it accelerates to be closed in the next 50ms. The reason is that the touchscreens of both ruggedized tablets which have been used sometimes lose track of the touching finger although it is still touching the screen. If this happens while using the Radar-Joystick, and if it squeezes too fast, the Minimap could be touched leading to a quick jump to the touched position inside the Minimap. Because the user actually didn't want to stop the scrolling with the Radar-Joystick in that context, this would have led to frustration by the user which has to be avoided. The buttons (number 3) are used to center the map on the current position of the Selection Quad, to **zoom** the map out to a bird-view of the whole MCI area while centered to the position of the Selection Quad, and to zoom the map out or in in that order. They can be easily reached with the thumb when the hands are positioned on the horizontal line going through the center of gravity.

7.3.1.3. General Features for both UI Alternatives

In addition to the feedback related to the presented UI elements, the AIOs of the previous evaluation mentioned that they missed an overview of the number of patients of the different triage categories. This is included in both alternatives in the upper left part of the visualization area. Another general feature which was added to both alternatives is the icon attached to those patients with additional relevant information labeled with an "i".

7.3.2. Evaluation - Formative User Study

In this section, the evaluation of the third iteration is described. First the participants' background is given, followed by the evaluation procedure. This consists of the evaluation design, the used questionnaires and the logged data. Afterwards, the tasks to be solved by the participants are explained in detail.

7.3.2.1. Participants

Seven participants took part in the evaluation of the third iteration. All of them were from the rescue service. But this time, in addition to five people from ASB, we also included one person from the Johaniter Bund and one from the Malteser Bund, which are also rescue service organizations in Germany. The reason for this was our goal to include also different cultures of different organizations. They aged between 21 and 52 years in age (average 35). Six of them were right-handed and all of them were male. Additionally, all of them were experienced with the handling of an MCI through regularly practiced real exercises. In opposite to the previous iterations, all of them were also experienced with the interaction on touch screen devices. This difference is due to the rapid change in the mobile market in the last few years. This fact increases also the expectations of the users and therefore also the subjective results.

7.3.2.2. Procedure

The evaluation procedure of the third iteration is presented in this section. First, the evaluation design is given. Next, the questionnaires are explained, followed by a description of the interviews with the participants. Afterwards, a description of the automatically logged data is presented. A within-subject design has again been used for the third iteration due to its more efficient use of the rare participants (AIOs). At the beginning of the evaluation a written introduction was given to the participants to ensure that each participant gets the same initial information. Afterwards, each participant had the opportunity to try out the concept that the specific participant should start with. They were observed during this part to learn about potential problems using the different interaction metaphors and to eliminate problems in understanding the UI elements. Then, the tasks consisting of three sub-tasks followed. Before each task, an explanation of the task to be solved was given to the participants as well as a description of the MCI situation of the specific task. When all tasks were finished using the first concept, the same steps were repeated with the second concept. As soon as a participant finished the tasks with the second concept he[1] was asked to fill out the SUS and NASA-TLX questionnaire. As another subjective rating, the participants were asked to rate all elements separately from 1 (best score) to 6 (worst score). At the end, an interview was conducted with each participant to get qualitative user-feedback. This time the interview guideline from Mieg et al. was used [69].

In addition to the subjective user ratings, objective interaction data was automatically logged by the implemented system. An important objective value is the distance traveled on the map with both concepts. The tasks are the same, consequently these distances should be similar too. If not, this indicated that the concept with a higher traveled distance is less efficient than the other one. The combined alternative is more complex, thus if there is a negative influence on the distance traveled it should be the combined version. However, because the power of both concepts is the same, we expected that there would be no significant difference. Another important objective value is the time needed to solve a task with both alternatives. The faster one will be simply the more efficient one. We expect no significant difference between both concepts in that point.

7.3.2.3. Apparatus

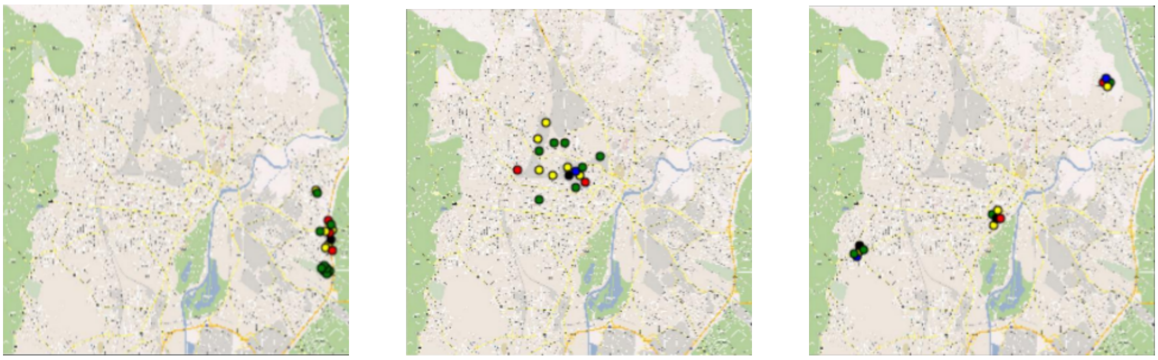
For hardware, we used the same convertible rugged tablet PC Getac V200 as in the second iteration. Details about the device is given in Section 7.2.2.2.

7.3.2.4. The Task

The task which should be solved by the participants using both alternatives consisted of three subtasks. As an important difference to the previous iterations, more attention was payed to the realism of the task by discussing the tasks in detail with an AIO of the rescue service. All of these subtasks were repeated three times with a different distribution of the patients in each repetition. The subtasks are described in the following:

[1] All of the participants were male

1.) Find Additional Patient Information As mentioned earlier, "info" icons were added to those patients who were tagged with relevant information (see for example Figure 7.28). In discussion with the AIO, the following three properties were chosen as additional relevant information: a) This patient is a child, b) This patient is poisoned, and c) intracranial injury. These properties are important for the AIO in case of a real MCI because they have a direct influence on the decisions of the AIO. For example, in case of a poisoned patient, a toxicologist is needed while patients with an intracranial injury need to be transported to a hospital with special resources to treat such a patient. The task of the test participants is to select all patients which are labeled with additional information. Figure 7.29 shows the three distribution-settings used for the first task.

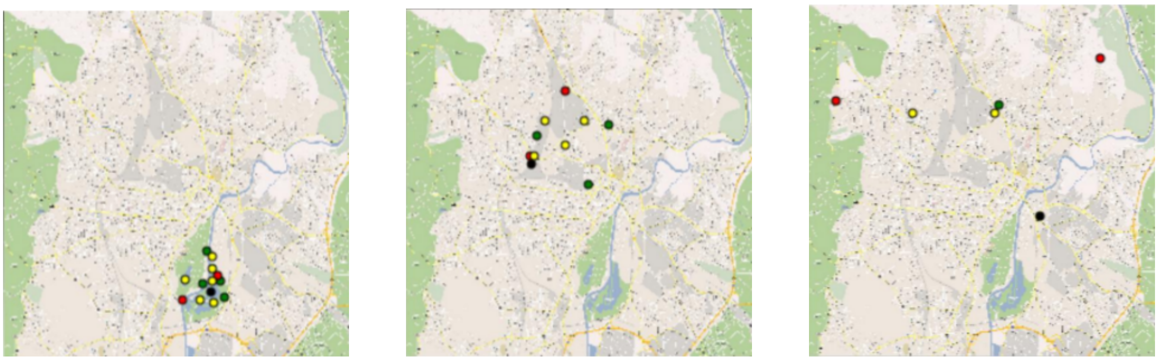


(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

Figure 7.29.: Task 1: Find additional patient information

2.) Tag Patients in Transport Order The second subtask is to select each patient on the map in the correct transport order which depends on the triage state. The goal of this task is to select each red triaged patient as soon as possible and to ensure that each triaged as red patient is selected for transport before starting with the yellow patients. The relevant subjective data were the time needed to finish the task and the number of errors made while solving the task. It was expected that less errors would be made when using the combined UI element, because the attention of the user was focused on one element while it moved from element to element in the case of the separated alternative. Figure 7.30 shows the three distribution-settings used for the second task.

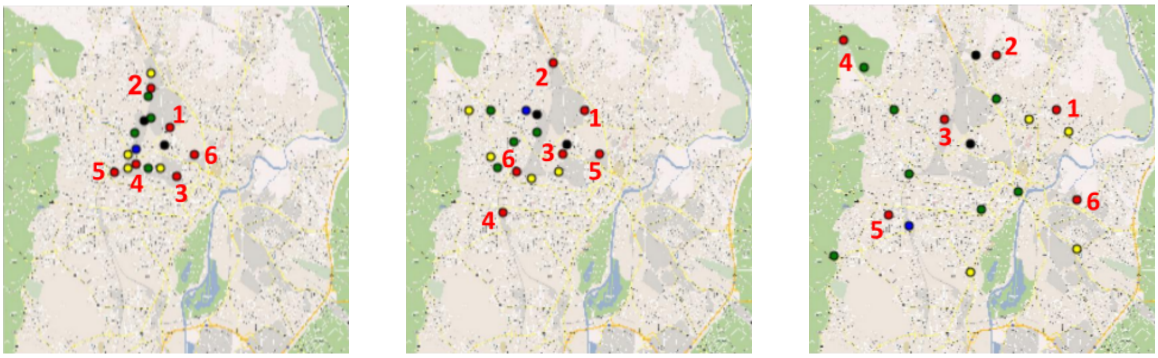
3.) Select Spontaneously Appearing Red Patients Subtask 3 simulated the triage phase of the MCI. The AIO observes new patients appearing on the map as soon as they are triaged by the relief units. For the evaluation, this is predefined and simulated. Thus, no real relief units are executing the triage and the patients do not exist. The task was to select each red patient as soon as the patient appeared on the map. In each distribution, 6 red patients appeared. While in subtask 2 the participant decided which of the red patients to select first, in subtask 3 the sequence was strictly ordered by the system. This way, the traveling distances could be compared independently of the user decisions. Figure 7.31



(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

Figure 7.30.: Task 2: Tag patients in transport order

shows the three distribution-settings used for the third task.



(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

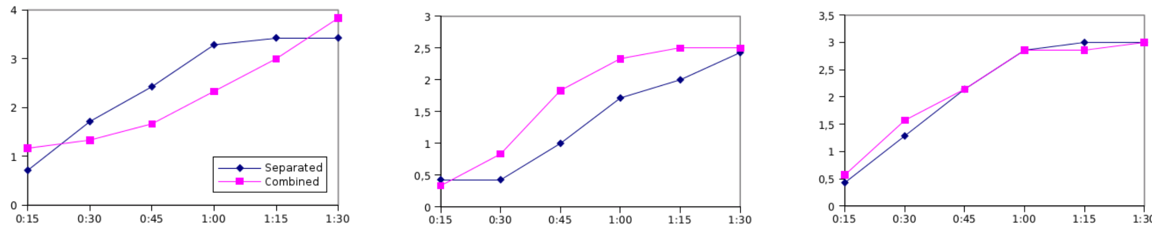
Figure 7.31.: Task 3: Select spontanously appearing red patients

7.3.3. Results

In this section the comparative results of the evaluation of the third are presented. First, the quantitative objective results gathered from the logged data are given in Section 7.3.3.1. Second, the quantitative but subjective results of the questionnaires SUS and NASA-TLX are shown in Section 7.3.3.2. Last but not least, the qualitative results of the interviews are summarized in Section 7.3.3.3.

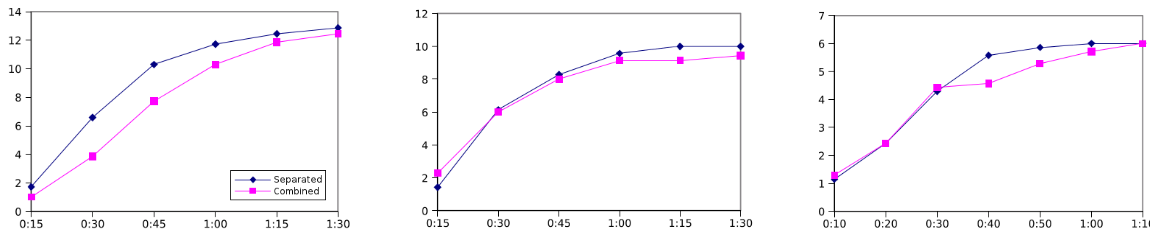
7.3.3.1. Quantitative Objective Results: Logged Data

The figures 7.32, 7.33, and 7.34 show the results for the three different distribution settings of the first, the second, and the third task respectively. The horizontal axis represents the time in minutes and the vertical axis represents the number of patients selected. No significant differences considering the average speeds for all subtasks with all distributions could be found. This is what we expected, since both setups provide the same features to the user.



(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

Figure 7.32.: Average time of Task 1. The horizontal axis represents the time and the vertical axis the number of patients selected.



(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

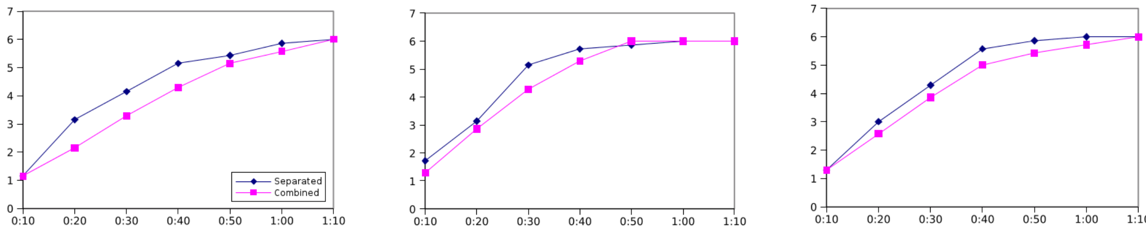
Figure 7.33.: Average time results of Task 2. The horizontal axis represents the time and the vertical axis the number of patients selected.

The distance traveled on the map is 54% higher in the case of the combined solution. The reasons for these objective ratings are discussed in Section 7.3.4.

7.3.3.2. Quantitative Subjective Results: Questionnaires

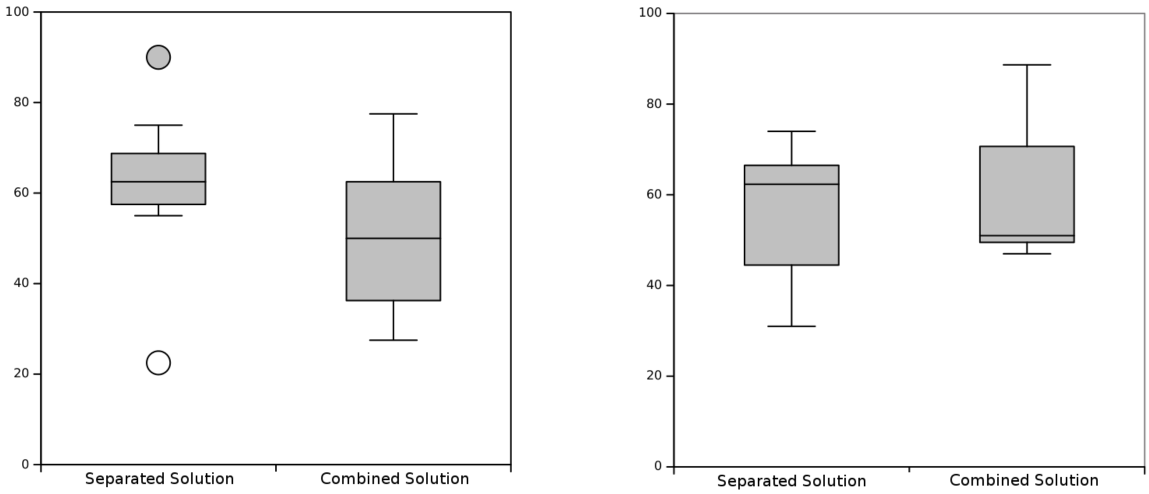
Figure 7.35 shows the results for the SUS questionnaire (Figure a) and the results for the NASA-TLX questionnaire (Figure b). The median value of the SUS score in case of the separated solution is 61.1 while the combined solution got a score of 50.4. This means, regarding the usability the separated solution seems to be the better one since the higher the SUS score the higher the usability.

In the case of the NASA-TLX score, the separated solution got a median of 62.0 while the combined solution got a value of 51.0. The lower the NASA-TLX score the less stressed the user is while interacting with the interface. But overall the boxblot of the combined solution ranges from 47 to 89 and the boxplot of the separated solution ranges from 31 to



(a) Narrow distribution (b) Moderate distribution (c) Large area distribution

Figure 7.34.: Average time results of Task 3. The horizontal axis represents the time and the vertical axis the number of patients selected.



(a) SUS (b) NASA-TLX

Figure 7.35.: Boxplots for the SUS and NASA-TLX scores.

74. Thus, in this case the median might not be the optimal value to compare the NASA-TLX scores for each solution. Consequently, more people felt more stressed while using the combined alternative then they felt when using the separated alternative. The reason behind is discussed in Section 7.3.3.3.

The average results for the subjective ratings of the single UI elements used in this iteration are given in Table 7.3. The Horizontal Bar for selecting items got results from 1 to 4 with an average of 2.6. The Zoom-Slider got results from 2 to 4 with average of 2.9. The Radar-Joystick's scores ranges from 2 to 6 with an average value of 3.6. The Minimap as a single element scores from 1 to 3 with an average of 2.0. The combined element scores from 3 to 6 with an average of 4.1.

UI element	Average
Horizontal Bar	2.6
Zoom-Slider	2.9
Radar-Joystick	3.6
Minimap	2.0
Combined element	4.1

Table 7.3.: The subjective ratings for each UI element used in this evaluation. 1 is the best score and 6 the worst.

7.3.3.3. Qualitative subjective results: Interview

To understand the reason behind the quantitative results, an interview was conducted with each participant after the usage of both presented alternatives. The results of the analysis of the interviews is summarized in the following and ordered by the main topics mentioned during the interviews.

Radar-Joystick The Radar-Joystick was not accepted by the participants and was rated worse than in the previous iteration. This is the case although exactly the same Radar-Joystick was used for the third iteration as was used in the second iteration. The reason is that the expectations of the users changed since the last iteration. While a touchscreen, even when not working perfectly, was an interesting and innovative new technology in the first iteration, it was already quite well known during the second iteration and it was commonly known for the third iteration. Modern smartphones and tablets increased the expectations of the users. The ruggedized tablets which have been used in the context of this dissertation, introduced in Section 5.4, could no longer compete with these increased expectations during the third iteration. Modern home-tablets are not only less heavy, the user experience is much better due to the high responsivity and sensitivity of modern touch screens. The influence of the worse quality of the touchscreen of the ruggedized tablet was even worse for sliding gestures. When sliding, it happens that the system loses the track of the finger, even though the user is still touching the screen. In case of the Joystick this led to small stops while using it for scrolling. This was frustrating for the participants. The radar functionality was rated as positive.

Minimap The Minimap got very good results in the third iteration considering the subjective ratings. With an average score of 2.0 (Table 7.3) it got the best score among all presented alternatives. But the users missed the option to scroll precisely with the Minimap. As a solution, one participant suggested adding a second mode to the Minimap to be able to scroll precisely when touching the yellow rectangle. But this is exactly what had been tried out in the second iteration and was rejected by the participants because of the weak touchscreen. Consequently, the idea of the second iteration to enhance the Minimap with a second mode seems to be a reasonable idea but the activation mechanism must not rely on precise interaction. The user should be able to enter the second mode without focusing on the Minimap.

Combined Element The weak touchscreen of the ruggedized tablets frustrated the participants even more in the case of the combined element. The reason for that was that when the screen lost the track of the finger, the joystick part of the combined element got squeezed down again. When the system did not regain track of the finger quickly, the joystick collapsed and the Minimap moved to its original position. But since the user actually did not remove the finger from the screen, the finger then was touching the Minimap without the user's intention. When the system then regained the track of the finger, the system considered this as a new touchdown event which led to a jump of the map to the touched position. This was extremely frustrating for the participants and led to the bad average score of 4.1 shown in Table 7.3. This also explains the larger traveling distance with the combined element. Whenever the map jumped to another position, the participant needed to restart the search for the patient and scroll back to the initial location. As an advantage of the combined element, the participants mentioned that there was no need to replace the hand to reach an element on the upper or lower corner of the screen because each element was always placed on the optimal location to be reached with the thumb. This is also the reason, why the zooming slider of the separated alternative was rated worse than the zooming buttons of the combined element.

Selection Metaphors The direct manipulation of the Horizontal Bar was judged as being more intuitive than the indirect manipulation of the Selection Quad used in the combined alternative. The Bar appeared exactly at the touching point of the user and followed the thumb when it was moved up or down. The Selection Quad required more attention from the user because it consisted of two elements, the joystick to move the quad and the quad itself.

The Tasks The participants also gave us feedback for the tasks themselves. According to that feedback, green patients should not appear on the digital map because they could walk, by triage definition. They are usually sent to a collecting point where they are under a physician's care. In Task 2 all patients should be marked for transport in the correct order. This included green and black patients which would not be the case in a real MCI because green and black patients did not need transport to a hospital. It was also mentioned that including vehicles to the MCI scenario would increase its authenticity.

Future Suggestions Gathered Through the Interviews Although not being the focus of this thesis, general feedback to the map application could also be collected through the interviews. For scrolling, it was suggested to integrate a feature to jump to different operational areas which could be defined beforehand or right in time. This is like a shortcut button which can be defined by users. The overview visualization of the map on the upper left edge (Figure 7.28) was rated as being very useful. But it would have been preferred to position it horizontally attached on a bar on the upper edge of the screen in which additional useful information could be placed, such as:

- Estimated number of patients not triaged yet
- The number of patients which have been transported already

- Weather information like the strength of wind and the temperature

In addition to the map itself the following views should be integrated in MCI applications:

- A view in which recently triaged patients are listed. This can be enhanced with a slider representing the time.
- A view with all patients listed which have additional information.
- A view in which the available resources are shown.
- A note-view

7.3.4. Discussion of the Third Iteration and Lessons Learned

In the third iteration, two integrated concepts have been evaluated with AIOs as participants, our target group. The first concept consisted of those UI elements which had been rated best in the previous iterations. The second concept represented a combined element which could be used for all three standard map application interaction features: Selecting, Scrolling, and Zooming. As an outcome of the third iteration the separated version was rated as being more intuitive and less complex to use. Quantitatively, there was no relevant difference in both concepts but the qualitative results revealed that the separated solution was preferred when used with the Durius V200. Consequently, as long as there is enough space for multiple separated elements we suggest providing the most intuitive UI element for each of the studied interaction features. Frequently used elements should then be at the ergonomically optimal position as discussed in Section 6.2. Elements which are less frequently used should instead be positioned at the corners of the screen when there is no space left directly at the thumb interaction areas. This way, hand movements can at least be reduced. If there is not enough space for providing all UI elements, the combined alternative might still be a solution. The main reason for the low results for the combined element is the low sensitivity of the screen of the ruggedized device. Not intended quick-jumps as a consequence of interacting with the Minimap were the source for the frustration of the participants while interacting with the interface. A solution for devices with such touchscreens is to restrict Minimap jumps to double-taps with the thumb. As a consequence, the frustrating origined by not intended quick jumps would disappear. Another solution could be introduced by the improved hardware. Currently, many ruggedized Android devices start to appear with much better touchscreens by still fulfilling the same tough security requirements, being less heavy, and which are equipped with more powerful processors at the same time. The results of the third iteration might differ on those new devices.

8. Conclusion & Discussion of Interacting with a Digital Map on a Ruggedized Tablet

In this part of the thesis our iterative and user-centered development procedure of a digital map application has been presented. The digital map application is intended to be used by AIOs during MCIs, in which the environment can be unstable and dangerous. For this reason, special requirements related to the hardware as well as to the graphical UI were collected in addition to standard UI requirements which each UI should fulfill. These requirements were presented in Section 5 in full detail. Each device which is intended to be used in the critical situation of an MCI needs to be 100% reliable. For this reason, people from the Garching TUM fire service told us that the device needed to be a ruggedized tablet device. Due to the weight of those ruggedized tablet devices, it made sense to develop the UI of the map application such that it could be used while holding the tablet with both hands. We therefore focused on the three main interaction features of a digital map application, selecting, scrolling, and zooming under special constraints. Different UI solutions for those requirements were introduced and improved according to the results of each iteration.

In the context of this thesis, three iterations have been performed. In particular, different alternatives to select items on a digital map and to scroll a digital map were developed based on literature research and on interviews with the target group. The first and the second iteration focused on one feature (selecting or scrolling) at a time and disabled the others. That way, it could assured that there were no influences because of using a different way to solve the task than the intended one, reducing the risk of biasing the results. The advantages and disadvantages of each alternative could therefore be identified and the individual concepts could be optimized according to the outcome of the first and second iteration. The third iteration differed from that approach because scrolling, selecting, and zooming were combined and it was up to the user to decide which one to use at a specific point in time. We presented two alternatives in the third iteration: Separated UI elements, one element for one function and a combined version which integrated the same features as the interface with the separated UI elements, but in a single compact UI element. The combined element consisted of an adapted version of the Minimap, simple buttons for zooming, and an integrated squeezed joystick (see Figure 7.28). The Joystick could be used to move the square as it is the case for the Selection Quad. But this time, once the square reached one edge of the screen, the display started to scroll to that direction. The integrated Minimap still provided an overview and the ability to quickly jump to a desired position. The same functionalities were also provided by the separated alternative (see Figure 7.27). But for selecting, the Horizontal Bar was used and for zooming a simple zooming slider was implemented. For scrolling, both, the Minimap and the Radar-Joystick were provided. Interestingly, the set of separated elements outperformed the combined element, although the users needed to move their hands more often than when using the combined element and although the separated version took a lot of scarce screen space. A reason for this might be that some not intended quick jumps with the integrated version of the Minimap led to some frustration of the users. Those unintended quick jumps occurred when the touchscreen lost contact to the finger although it was still touching it. When this happened while the user was using the integrated joystick, the system "believed" that the user wanted to stop interacting with the joystick and it squeezed down again. Although a time delay was built in to alleviate this problem, it still happened to our participants. This led to the mentioned quick jump although the user intention never was to use the Minimap. This is a critical misinterpretation of the user's actions. The source of this critical misinterpretation was the low reactivity and sensitivity of the touchscreens of both ruggedized tablet devices that we used for testing in the context of this thesis. Therefore, one conclusion of the last iteration was to use one element for one feature as long as the available screen space allowed it. Additionally, it should be avoided using interaction metaphors based on sliding gestures and taps at the same time on a single element. Another conclusion of this part of the thesis was to always provide a Minimap to the AIO because of the important overview it provided to the user. But another way to precisely move the map was also necessary. The virtual Radar-Joystick worked very well for that purpose. For selecting items on a digital map, the Red Bar tended to be the best choice. Although the Automatic Mapping was faster and the Selection Quad was more flexible, the Red Bar got more positive feedback than the other two alternatives. The speed advantage of the Automatic Mapping was rated as less important by the participants from the Arbeiter-Samariter-Bund München and the Selection Quad was rated as more complex than the Red Bar and was preferred for that reason.

The conclusion of this work has also been presented as invited talks at the Joint Research Center (JRC)in Ispra, Italy [22] and at the Search and Rescue conference in Oradea in 2012 [23].

9. Future Work

As discussed in the previous section, the interaction design of the UI elements is restricted to the limitations of the ruggedized tablet devices used in the context of the thesis. The results of the evaluations presented in this part are strictly coupled to the usage with those heavy devices. However, tablet devices and touchscreens evolved a lot in recent years. Thus, as future work two kinds of approaches should be investigated.

First, iteration two and three should be repeated on modern tablet devices. This way, it is possible to learn whether the presented elements suffered more or less from the limitations of the used touchscreens. This approach also makes sense, because the prices of those devices dropped a lot in recent years and the prices will continue to fall in future. Consequently, these devices can be used as simple tools while the important data is stored and synchronized by e.g. a cloud in real-time. When such a tablet device falls down and gets damaged the AIO simply pulls out the next device. For this approach it is important to have a reliable and perfect ad-hoc network at the location of the MCI. Stable and reliable ad-hoc networks are research topics on its own. Once it can be assured that a stable and fast network is also available during such an incident, the described approach is a promising one.

A second approach is to use high-end ruggedized tablet devices just appearing in the market which are also much more comfortable to use while still being ruggedized. The usage of those modern ruggedized tablet devices might eliminate the limitations of the ruggedized tablet devices used in the context of this thesis.

As soon as iteration two and three has been repeated with more modern tablet devices, an evaluation with many more participants makes sense in order to statistically prove significant differences in the results for the developed UI elements.

Part III.

Mobile Text Input

10. Introduction

It has been become evident during the requirement analysis of the map application interface that mobile text input is an important requirement for MCIs. Therefore, we have investigated different alternatives for mobile text input compatible with the requirement of being usable just with the thumbs while holding a tablet in both hands. Thus, our first step was to conduct an initial study in which we adapted and compared different text input concepts from literature. This initial study is described in Section 13.1. Although the users of this first study reached a reasonable speed with the split keyboard introduced in that study, the restriction to the thumbs slowed the user down when compared to a text input concept in which the users could use all 10 fingers simultaneously. Because of that, we focused on eliminating this restriction by enabling the user to interact with their 2x4 fingers on the back of the device. For this, a touch sensitive surface was needed on the rear side. Assuming that such devices will be ubiquitous in the near future, we developed a text input concept based on finger gestures and the 10-Finger-System which enables the user to blind-type on the back of a device by using the eight fingers. However, because this concept is completely new and there is no prior experience in using finger dependent gestures for all fingers simultaneously, we first developed a stationary version for tabletops or large tablets. In this stationary concept the fingers were not occluded and the QWERTY layout could be used directly without any rotation of the layout. This was not the case for the back-touch version in which both hands were rotated -90° and 90° and therefore the QWERTY layout had to be adapted accordingly. Because this mobile version was more complex, the finger gestures for the mobile version are more challenging. Thus, it makes sense to first develop and evaluate the stationary version to discuss and improve the concept itself and its implementation and then to start with the development of the mobile version afterwards.

Because the principle concept is based on **gest**ures and it basically is a ke**yboard**, we called our concept **Gestyboard**. The three iterations of the **stationary version** of the Gestyboard are described in the chapters 13.2, 13.3, and 13.4. The single steps in each iteration follow the same iterative user-centered procedure which is used for the map application development shown in Figure 3.1 and consists of the requirement analysis, development and optimization of the concept, the implementation, the evaluation, and the discussion of the results. Afterwards, the first iteration of the **mobile version** of the Gestyboard, called **Gestyboard Backtouch** is given in Chapter 13.5.

Our own requirements on which the concept is built are listed and explained in Chapter 11. Afterwards, related work in this field is presented in Chapter 12. This includes other mobile and stationary text input concepts, similar text input concepts, and basic text input evaluation techniques. The conclusion and discussion is given in Chapter 14 and ideas to further improve the Gestyboard concept are then listed and described in Chapter 15.

11. Requirements

We first defined requirements which a text input concept should fulfill to be accepted by the users and to compete with classical hardware keyboards. Because the most powerful, known and accepted keyboard is still the classical hardware keyboard we collected important properties, advantages as well as disadvantages, and extracted our requirements from those properties. Figure 11.1 summarizes them. Since the goal of the Gestyboard is to compete with the classical hardware keyboard, we deduced requirements from those properties and believe that a keyboard concept which fulfills all of those requirements can become a real competitor to the classical hardware keyboard.

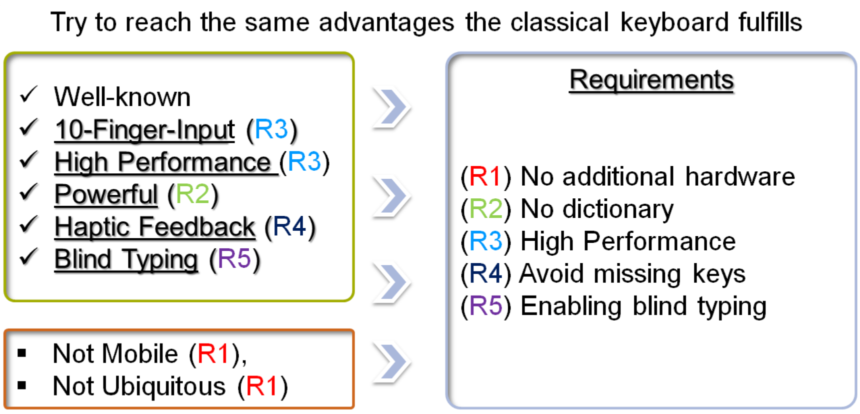


Figure 11.1.: Our Requirements which are derived from the properties of the classical hardware keyboard

The main disadvantage of the classical hardware keyboard is its lack of mobility and ubiquity. This is actually the reason why many new mobile text input concepts have been developed in recent years for smartphones. But because touchscreens become more and more ubiquitous a text input concept which does not require other hardware than a touchscreen can become ubiquitous too once it is available on all those displays. Consequently, our first requirement (**R1**) demands to avoid the need for additional or special hardware. **R2** defines the requirement not to use a dictionary or a language model. While it is well known that a dictionary increases the performance it also reduces the expressive power of the text input, as words unknown to the system can no longer be conveniently entered. This is not the case for the classical hardware keyboard. Because of that, the user is able to switch the language while chatting, to shorten words if wished, to type names the system does not know yet and to use individual words which do not appear on dictionaries without adapting the dictionary first. This also avoids wrong corrections of the text input concepts which can change the whole meaning of the sentence. Additionally, a language model can always be added on top of the initial keystroke input at the application level in order to improve the performance. **R3** formalizes the requirement to enable the user to reach a speed around 60 WPM since trained users are able to reach this speed with the classical hardware keyboard. The fourth requirement (**R4**) demands enabling the user to avoid missing keys. This is not fulfilled by touchscreen keyboards because of the missing haptic feedback. In contrary to that, the haptic feedback of the classical keyboard provides this feature. For this reason, the activation mechanism of each keystroke has to be unique to remove the need for haptic feedback in our concept. This is related to our last requirement (**R5**) which ensures that the text input concept supports blind-typing. This means that the user should be able to type text while focusing on the written text without looking at the keyboard.

Table 11 gives an overview of some text input concepts which have been developed during the last 20 years and checks which of our requirements they fulfill. The classical keyboard is also listed. Except for the classical keyboard, none of the text input concepts fulfill all of the the requirements from **R2** to **R5**. Consequently, the probability is high that a concept which fulfills all of those requirements can become a real competitor to the classical keyboard. A concept which additionally fulfills **R1** becomes automatically ubiquitous and, in consequence, this special concept might replace the classical keyboard in future.

Table 11.1.: Several text input concepts and their relation to our requirements. In this table a requirement is fulfilled if we found an evidence for it. **R3** is true, if a speed of 60 WPM could be reached.

Concept	**Ref.**	**Year**	**R1**	**R2**	**R3**	**R4**	**R5**
Classical keyboard	[29]	1878	no	yes	yes	yes	yes
Unistrokes	[32]	1993	yes	yes	no	yes	yes
TCube	[98]	1994	no	yes	no	no	no
Quikwriting	[80]	1998	no	yes	no	no	no
Dasher	[101]	2000	yes	yes	no	no	no
EdgeWriting	[106]	2003	no	yes	no	yes	yes
Graffiti	[15]	2008	no	yes	no	no	yes
Bader keyboard	[8]	2008	yes	yes	no	no	no
Microsoft Split	[70]	2009	yes	yes	no	no	no
Fitaly	[31]	2010	yes	yes	no	no	no
LiquidKeyboard	[86]	2011	yes	yes	no	no	no
Swype	[94]	2011	yes	no	no	no	no
1Line	[56]	2011	yes	no	no	no	no

12. Related Work

With mobile computing become more and more ubiquitous the development of new text input concepts has also rapidly evolved. Despite this, the performance of touchscreen based keyboard concepts still suffers, primarily due to the lack of tactile feedback [58]. This chapter first introduces in Section 12.1 related studies which focus on enhancing a touchscreen with some kind of artificial haptic feedback. This is followed in Section 12.2 by an overview of mobile text input concepts which have been developed in recent years. Afterwards, text input concepts which are designed to be used with 10 fingers simultaneously are introduced in Section 12.3. Last but not least, in Section 12.4 the most important basics related to text-entry evaluation techniques are given.

12.1. Enhancing a Touchscreen with Artificial Haptic Feedback

Many research projects tried to solve the problem of the missing tactile feedback by enhancing the touchscreen with some kind of tactile facility. Hoggan et al. investigated the effectiveness of tactile feedback for touchscreen smartphones [38] with a finger-based text input system. According to their results, the performance of the finger-based text entry can be significantly improved by providing tactile feedback. To simulate the tactile feedback, the built-in smartphone vibration actuator was used. In 2010 Bau et al. introduced a way to provide tactile feedback for larger devices, like multi-touch tables: the so called Tesla Touch technology, based on the electro-vibration principle [9].

But even though adding tactile feedback to a device improves the performance of the text entry, in our opinion the speed and the accuracy of the classical keyboard cannot be attained as long as the tactile feedback is not as precise as allowing the user to clearly distinguish the edges between the keys by sensing them without activating them. Additionally, large multi-touchscreen displays that provide tactile feedback are infrequent and expensive. Therefore, an alternative is to develop completely new concepts for touchscreen text input systems with no need for tactile feedback. Kölsch et al. [48] conducted a survey on virtual keyboards in 2002 and classified them into four main categories: 1.) speech recognition, 2.) handwriting recognition, 3.) sign languages and 4.) touch typing. The last category encapsulates all touch based text input systems. Therefore, the virtual keyboard is a sub-category of touch typing. However, Kölsch et al. did not distinguish between touch typing with or without requiring tactile feedback.

12.2. Mobile Text Input Concepts

Our first approach to develop a text input concept to be used by thumbs only while holding a tablet in both hands was to study related mobile text input concepts and to adapt them such that they can be used with the thumbs. In this section we introduce and rate those mobile text input concepts.

12.2.1. Unistrokes (1993) and Graffiti (1997)

Goldberg et al. [32] developed a text input concept in 1993 called Unistrokes based on simple gestures. These gestures were designed to satisfy three major criteria: simplicity in learning, simplicity in execution and uniqueness. It has been patented in a Google-Patent in 1997 [33]. The execution of the gestures does not rely on tactile feedback, this makes blind-typing possible. Figure 12.1 shows the alphabet of Unistrokes.

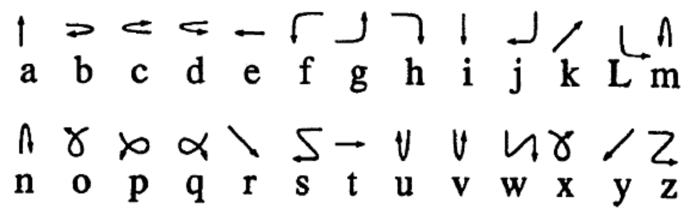


Figure 12.1.: The Unistrokes alphabet (original source [33])

The unique gestures are simple gestures as shown in the figure. They are easy to remember and also easy to execute without focusing on the gesture. However, because there is no real relation between the gesture and the associated letter, the gestures were not intuitive and had to be learned by heart. Because of that, the PDA manufacturer PALM adapted the Unistrokes alphabet and created the Graffiti alphabet in which single gestures are designed to be closer (in shape) to the associated letter, thus being easier to learn and easier to remember. Figure 12.2 shows the adapted alphabet.

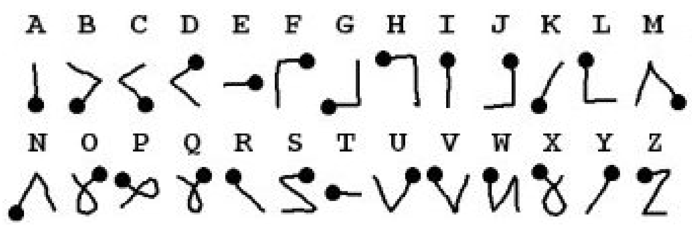


Figure 12.2.: The Graffiti alphabet (original source [64])

Castellucci et al. [15] conducted an empirical evaluation between both alphabets in 2008. In a longitudinal evaluation consisting of 25 sessions the participants reached an average speed of 11.4 WPM for Graffiti and 15.8 WPM for Unistrokes. Both alphabets had high correction rates, but while the correction rate of Graffiti remained relatively consistent at 26% in that study, the correction rate of Unistrokes decreased from 43% to 16%.

12.2.2. T-Cube (1994)

In 1994, Dan et al. developed T-Cube [98]. Intended to be used with a pen, T-Cube is a mobile text input concept which does not require a lot of screen space - an important requirement for mobile text entry. It introduces an alphabet of "flicks". A flick is basically a sliding gesture consisting of two parameters: The starting point of the gesture and the direction of the movement of the pen. Figure 12.3 shows the original conceptional figure of T-cube.

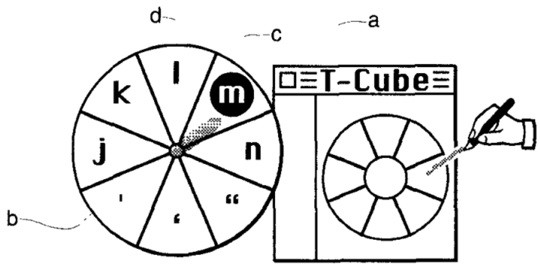


Figure 12.3.: T-Cube: a) shows nine possible initial targets for the pen, b) shows the pie menu which opens once a target is activated, c) the highlighted character, and d) the highlighted character is typed once the pen is lifted (original source[98])

First, one of the nine shown targets has to be chosen by the user (labeled with a in Figure 12.3). Once a target is entered with the pen, a pie menu is shown to the user, showing those characters which can be typed with that target. By moving the pen, the user can choose one of those characters and the highlighted character is typed as soon as the pen is lifted. Because this technique can be adapted such that it works with the thumbs instead of with a pen, we chose this concept as one alternative for our initial text input study as described in Section 13.1.

12.2.3. Quikwriting (1998)

In 1998, Perlin developed the stylus-based text entry concept called Quikwriting [80]. The specialty of Quikwriting is that the stylus never needs to be lifted, thus leading to a continuous styling motion. Figure 12.4 shows an example how to write the letter 'f' on the left, and the word "the" on the right.

A stroke always starts at the center of a visualization. Then a sliding gesture has to be executed in the direction in which the desired letter is positioned. The letter 'f' is located in the upper right corner of the the visualization in that figure. Consequently, a sliding gesture to the upper right has to be executed. At this state, the letter 'n' is selected. To get the letter 'f' which is positioned to the left side of the letter 'n' the sliding gesture has to be continued in the left direction. Then, the letter 'f' is selected [1] The letter 't' is located at the right field of letters). Once the desired letter is selected, it is typed when re-centering the stylus. The example on the right side demonstrates how to write the word "the": Again,

[1] In Figure 12.4 the letter 'f' looks like the letter 't', but this is due to the font.

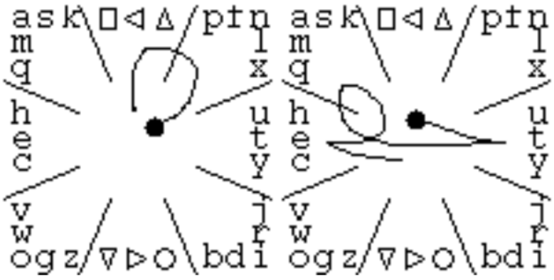


Figure 12.4.: Quikwriting: An example of writing the letter ’f’ and the word “the” [80]

the stroke starts at the center. Then, a sliding gesture to the right side is executed to reach the letter ’t’. Without lifting the stylus the gesture has to be continued with a movement in left direction. To get the letter ’h’ a movement to the top direction follows. Afterwards, the movement continues in the left direction again to finally get the letter ’e’. By returning to the center, the word “the” is typed. This leads to the mentioned continuous movement while writing a word. This concept was initially designed to be used with a stylus, but efforts to adapt it to be also usable with a finger have been done for example by Isokoski et al. [40], [41]. Consequently, this concept also is a candidate for our initial text input evaluation.

12.2.4. Swype (2010)

A commercially successful example is Swype [94]. It is designed to work across a variety of devices. According to the developers of Swype, a speed of 40 WPM can be reached with that keyboard. Like most modern smartphone keyboards it is based on a dictionary. The user does not type the single letters of a word directly, instead a “swype” gesture is executed on a QWERTY layout containing all of the letters which are part of the word in the correct order. There is no need to perfectly hit the letters on the way. Nevertheless, most of the suggestions are correct, even if some of the letters were not hit during the sliding gesture.

12.2.5. SwiftKey (2011)

Another commercially successful example is Swiftkey [93]. While Swype changes the way of interacting with a softkeyboard from taps to continues sliding gesture, Swiftkey keeps the standard way of interacting with the keyboard by taps. But like for Swype, the user does not need to be very precise with the taps, because SwiftKey is also based on a dictionary and tries to guess what the user wanted to type instead of accepting what the user actually typed. Many softkeyboards are actually enhancements with a similar concept to get rid of the need for precise interaction which is known to be a challenge for touchscreens, especially for small touchscreens. But the specialty of SwiftKey is that is does not only guess what the user intended to type, it also predicts what the user wants to type next. This is achieved by analyzing the users’ personal text messages like SMS, E-Mail or any other text of the user. For that, the application first asks for permission to access this data which is available on most smartphones by an Internet connection and the locally stored SMS cache. By guessing what the user wants to type, even if the user did not finish the word yet and by even predicting the next word, the efficiency of typing text can apparently be improved.

12.3. Text Input Concepts designed for 10 Finger Interaction

After of our initial text input study related to the interaction with the two thumbs, we developed a concept which allows to use all 10 fingers simultaneously by still allowing the user to blind-type. Because of that, we also studied different text input concepts for touchscreens which also allows to type with 10 fingers simultaneously. Some examples are given in this section.

12.3.1. A Virtual Touchscreen Keyboard Adapted to the Users Hands (2008)

An early attempt to develop a text input concept for 10 finger usage on touchscreens has been performed by Patrick Bader in 2008 [8]. The layout of this keyboard is designed such that it ergonomically fits to the users hands. Figure 12.5 shows the layout of this keyboard which is based on the QWERTY layout. The fingers rest on the home row keys and a keystroke is performed by tapping the desired keys or buttons.

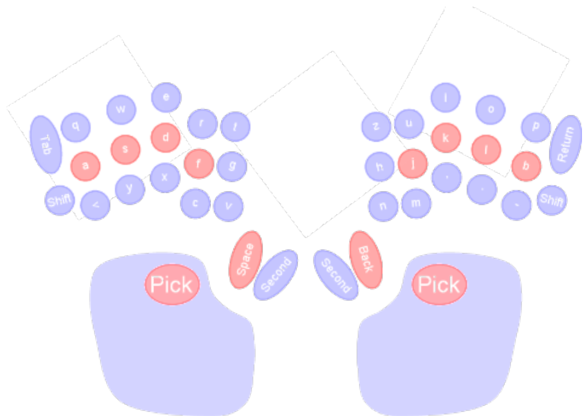


Figure 12.5.: The layout of the touchscreen keyboard from Patrick Bader

12.3.2. Microsoft's Split Keyboard (2009)

Mircosoft published the *US patent 2009/0237361 A1* in 2009 covering a virtual touchscreen keyboard which automatically adapts to the user's hands [70]. To do so, the keyboard is split in two parts, one for the left hand and one for the right hand as shown in Figure 12.6. If one hand moves or rotates while it is still touching the surface of the touchscreen, the corresponding part of the keyboard follows it.

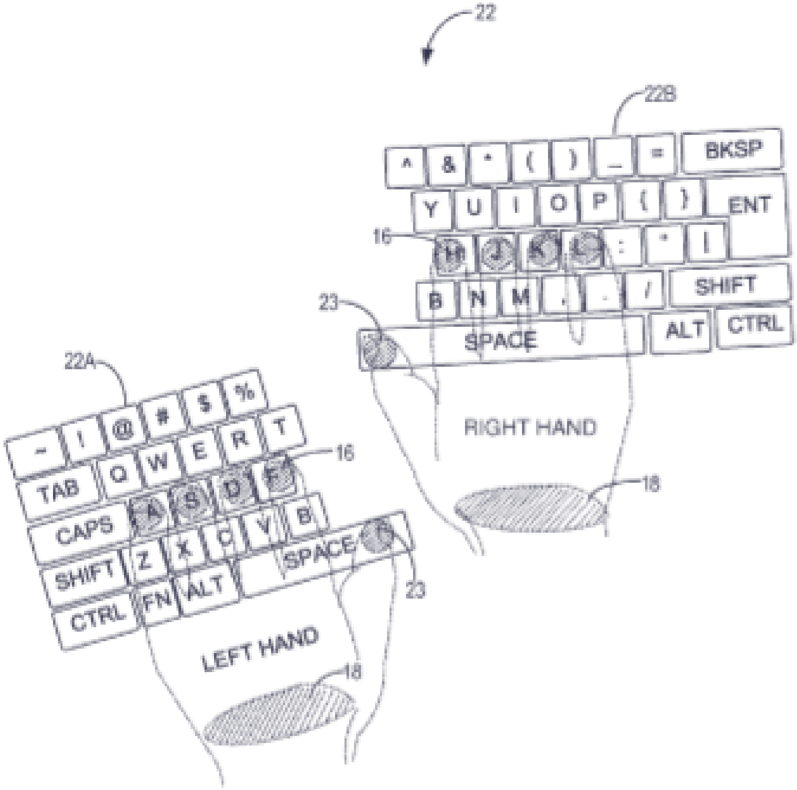


Figure 12.6.: The layout of the touchscreen keyboard from Microsoft.

12.3.3. LiquidKeyboard (2011)

Another virtual keyboard concept for touchscreens, the LiquidKeyboard, has been introduced by Sax et al. in 2011 [85], [86]. The keys of the virtual keyboard are placed at the position of the user's fingers when the display is touched. The standard QWERTY layout is used to define the position of the keys. When moving a finger on the screen, the assigned group of virtual keys follows it. This way, the keyboard automatically adapts to the user's hand physiology. The key activation method is to tap the keys. Two versions of the LiquidKeyboard were developed and empirically tested. As of now, there are no user performance related results available, but this is planned for future research. Figure 12.7 shows the LiquidKeyboard when touched with 8 fingers.

12.4. Text Input Evaluation Techniques & Basics

A variety of different techniques exist in the literature for evaluating text input concepts. This section presents relevant basics concerning the evaluation design (12.4.1), relevant metrics (12.4.2), and the writing task (12.4.3).

Figure 12.7.: The LiquidKeyboard developed at the University of Technology, Sydney (UTS)

12.4.1. Evaluation Design

In this section important properties of the evaluation design for a text input concept are explained.

Validity It is important to ensure that the results obtained by the evaluation have a high degree of internal as well as external validity. A high degree of internal validity means that the results are not falsified by external parameters. For example, when developing a new layout for the classical keyboard it is difficult to reach a high degree of internal validity when comparing the new layout with the standard QWERTY layout. The reason for this is the users' prior knowledge and training with the QWERTY layout. Consequently, the QWERTY layout will get better results than the new layout even if the new layout is better. To reach a high degree of internal validity in this case the users need to get used to the new layout first. A high degree of external validity means that the results are also valid for other setups than the specific evaluation setup. In the case of text input, this means that the evaluation setup should be designed such that it fits to the normal writing behavior of the participants. This is called the "unconstrained text entry evaluation paradigm" [61].

Number of participants MacKenzie and Tanaka-Ishii [62] suggest to use a similar number of test users to what has been used in literature for similar text input techniques. For longitudinal evaluations five [66] to twelve [51] test users are common.

Learning Behavior An important property for innovative text input techniques is the learning behavior. If the new concept is easy and fast to learn, the probability that the new concept will be accepted and used by the users is increased. The learning behavior is different for the speed and the error rate of a new text input concept ([39]). In the first period of learning, the error rate usually decreases until the user starts accepting a specific number of errors. After that point in time, the error rate remains more or less constant.

If the new technique is better than the existing one, then the test users' performance of the new technique should at some point exceed their performance of the existing technique. This point is called crossover. If this point is not reached by the new text input technique, one possible reason is that the new technique actually is not better than the existing one. Another possible reason is that the new concept can be further optimized. But it is also possible that the period of the time to learn the new technique was not long enough [66]. Figure 12.8 illustrates this behavior.

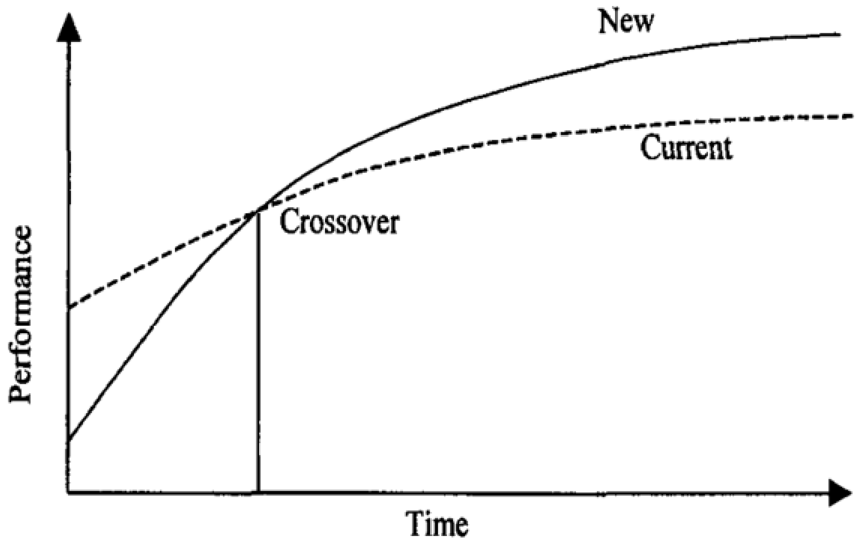


Figure 12.8.: The Crossover of two different text input concepts [66]

Initially, the measured performance follows the power law and it keeps following the power law long enough to be a good approximation to the real performance if the sessions had been continued [39]. Equation 12.1 represents the power law in which r_n represents the speed of the n-th repetition. x represents how fast the typing speed changes and is estimated through the measured values so far.

$$r_n = r_1 n^x \tag{12.1}$$

Novice Users The writing performance of novice users is an important factor for text input concepts. It has an impact on the acceptance of the new concept. It is therefore important to conduct an evaluation which focuses on the writing performance of novice users when developing a new text input concept. For this reason, a common evaluation design starts measuring the performance of the user right after a minute of trying the new concept out and again after five minutes of typing. Another design starts measuring after the first sentence has been typed by the participant [65].

Advanced Users Another important factor for a text input concept is the performance of advanced users. For this, longitudinal evaluations consisting of multiple sessions are usually conducted to allow the participants to learn the new concept from session to session and to improve their performance while doing so. The performance is measured in each session thereby allowing to evaluate the learning behavior of the concept [62]. Usually 10 to 20 sessions are performed (see for example [15], [17], [51], [27]).

Expert users Even when conducting a longitudinal evaluation, the time each user spends with the system is usually not sufficient to become an expert [63]. But because the typing speed follows the power law it is possible to extrapolate the future potential performance of each user. Examples in which this has been done are Clarkson et al. [17], Costagliola et al. [27], and Wigdor et al. [102].

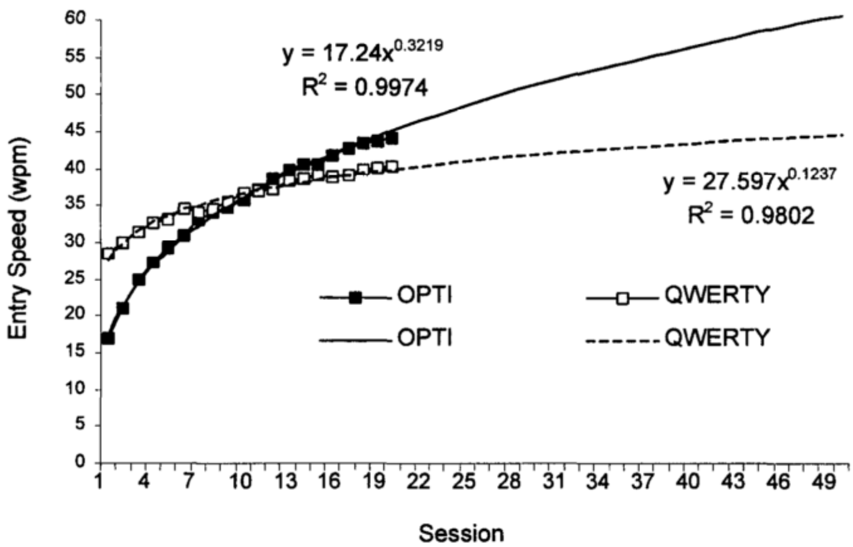


Figure 12.9.: An example of the learning behavior for two text input concepts with 20 sessions extrapolated to 50 sessions [66].

Figure 12.9 shows an example of the learning curves for two text input concepts of a real evaluation. 20 sessions have been conducted in that evaluation. This is extrapolated to the 50th session. The closer the value of R_2 to 1 (100%) the higher the prediction quality [66]. Another approach to estimate the expert performance is called *Accelerated Learning*. For this, the participants repeat typing the same sentences again and again to simulate the expert performance [59]. For that kind of study the choice of the sentences which should be typed is important. MacKenzie et al. [59] for example use the pangram *"The quick brown fox jumps over the lazy dog"* for this purpose. As being a pangram, this sentence uses every letter of the alphabet and is therefore more difficult to type than an average English sentence. Therefore, the results should be more conservative compared to typical English according to the authors.

12.4.2. Relevant Metrics

In this section, the most important metrics are summarized as being important basics for each text entry experiment. First, the typing speed is described in Section 12.4.2.1. This is followed by the definition for the error rate in Section 12.4.2.2 .

12.4.2.1. Typing Speed

Typing speed is usually measured in words per minute (WPM). One word is defined as a sequence of 5 characters including spaces and signs [3]. Equation 12.2 shows the formula to compute the WPM. S is defined as the time elapsed from the first keystroke to the last keystroke. $|T|$ is the length of the transcribed text. $|T| - 1$ is consequently the length of the text without the first letter. This is divided by 5 due to the average number of letters in a word in the English language including spaces and signs and it is multiplied by 60 to get the words per minute. Letters or symbols which appear in the input stream but not in the original text are excluded from that computation [3].

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \tag{12.2}$$

12.4.2.2. Error Rate

There are two relevant different approaches to analyze the error rate of text input concepts. One is to classify the errors into different types of errors to find out the sources for specific problems. The second is to focus on general error rates and to the typing speed achieved by the participants [46].

Error Classification Errors are usually classified into the following three main categories: **A) Insertion**, **B) Omission**, and **C) Substitution**. An insertion error is an error in which the user typed a letter which is not part of the original text. An omission error is an error in which a letter which should have been typed is missing in the input stream and a substitution error is an error in which a wrong letter has been typed instead the intended one. These main classes can be further split into multiple subcategories. For example, Kano et al. [46] splits the substitution class into those substitutions in which neighboring keys have been typed instead the intended one or substitutions in which a letter which should have been typed in capital has been typed in lowercase.

Total Error Rate The Total Error Rate gives an insight into how error-prone a text input concept is in general [105]. This is usually used when two text input concepts are compared against each other.

12.4.3. Writing Task

For text entry evaluation, different types of writing tasks exist. In the following, the most important basics related to the writing task are described. Additionally, it is explained why for most text input evaluations the transcription is usually used.

12.4.3.1. Text Transcription vs. Text Generation

According to MacKenzie and Soukoreff [60], there are two main classes of writing tasks: text generation and text transcription.

Text generation allows users to type what comes to their minds while executing the task. This is how users type in their normal life when generating text like E-Mails or other documents. This increases the external validity of a study. But as a side effect, the results can be affected by factors which cannot be controlled in the study. For example, if the user needs to think about what to type next, this has a direct influence on the writing speed. Another problem attached to this kind of writing task is that it is difficult to analyze the error rate of the text input concept, because it is not always clear whether users typed what they intended to. For example, if the user mistyped a word the evaluator has no way to determine whether this error was made due to a problem with the text input concept or whether the user simply thought that the word is spelled like he/she wrote it. Another problem with this technique is, that the written text does not need to reflect a representative text for the used language. For this reasons, a transcription is usually preferred.

Text transcription requires the user to transcribe a presented text. All of the previously mentioned problems with text generation does not apply to text transcription. A representative text can be chosen for transcription (for example TEPS [61]). The users do not need to think about which sentence to type next, and it is known what the user should have typed. But this also introduces a negative side effect: The users are not able to focus on the written text only as it is possible for text generation. They additionally have to look at the presented text to be able to transcribe it.

Focus Of Attention (FOA) is a variable introduced by MacKenzie and Soukoreff [60]. It describes the required attention by a text input evaluation which uses the transcription technique. There are three different sources which can require the attention of the user, depending on the users skills and the text input concept itself: The first FOA is the written text. The second FOA is the presented text and the third FOA is the text input concept itself. Expert users in the case of the classical keyboard, for example, can type without focusing on either the written text or the keyboard. They only need to focus on the presented text which they should transcribe. Consequently, this is a One-FOA-Task. Less trained users have to additionally focus on the written text to be able to see whether they typed what they intended to type and to correct letters if necessary. This is a Two-FOA-Task. If the user additionally needs to focus on the text input system itself, it is a Three-FOA-Task.

12.4.3.2. Text Presentation

MacKenzie and Soukoreff [60] also introduced ways to reduce the increased need for attention for transcription tasks. One way is to present the text acoustically instead of visually. But because participants in the past felt exhausted and could not always follow the spoken text this way is rarely used. Another way is to place the presented text directly below the written text in a way, that each written letter appears directly above the corresponding letter in the presented text. Another technique is to first show the participant a short and memorable sentence. Once they know the sentence by heart, they start typing it without any presentation. One known technique here is to hide the shown text as soon as the participant typed the first letter. MacKenzie and Soukoreff [90] compared the effect of this technique with the same technique when the text is shown all the time. When the text is hidden, users tend to type faster. Yet, at the same time more errors are not corrected by those participants. When presenting the text all the time, users are a little bit slower on average but they do correct more errors. The reason behind this is that the shown text simplifies the correction itself because the differences are visible all the time. Kristensson and Vertanen confirmed that result [52]. Additionally, they revealed that the time required by the writing task increases in the case of the hidden text, although the users tend to be faster with it. This is because the users need additional time to learn the sentences by heart before they start typing.

12.4.3.3. Conditions for Error Correction

Measuring the error rate when the correction of errors is allowed is not trivial. Soukoreff [91] gives an example to illustrate that:

Presented Text: the quick brown fox

Transcribed Text: the quicxk brwn fix

When comparing both sentences "naturally", it is obvious that three errors have been made by the user. The accidentally inserted 'x', the missing 'o' and the mistyped 'i'. The first approach to automate the measuring of the error rate was to simply compare both sentences letter by letter. This approach measures six errors although some of the counted letters were entered correctly [91]. The simplest solution is to restrict the writing task itself in a way, that errors are simply not accepted by the system. This means, that those errors do not appear and the transcribed text continues as soon as the participant typed the correct letter. In fact, this is what we did in the case of the first (Section 13.2) and second iteration (Section 13.3) of the Gestyboard. This is reasonable for the first evaluations but once a text input concept should be used in real-life, the efficiency of the backspace key is especially important. Soukoreff and MacKenzie [89] conducted an experiment in 2003 in which the natural typing behavior of 4 typists has been recorded and observed for approximately a month and analyzed afterwards. According to the results of that experiment, 31,4% of all keystrokes do not appear on the visible text. Those keystrokes are, for example, modifier keys, the arrow keys and also the backspace key. In fact, the backspace key was the second most frequently used key of all keystrokes. Consequently, when a text input concept is still in an early step of development, it makes sense to restrict the evaluation task to focus on special aspects of the keyboard. But to prove that the keyboard is efficient for the real usage in life and to increase the external validity of the results, more advanced algorithms have to be used to measure the error rate when allowing the user to correct errors: "The text entry process is really the editing process, involving much more than the perfect linear input of alphanumeric symbols. By forbidding participants to correct their mistakes, researchers are missing an extremely important part of the text entry process" [91].

12.4.3.4. Text Selection

Kristensson and Vertanen defined criterion for the selection of the presented text [52]. To achieve a high degree of external validity, the text needs to be representative for the used language to reflect common text entry behavior. To achieve a high degree of internal validity, the sentences need to be memorable and should not use unusual punctuation characters. Another factor influenced by the choice of the text is the comparability to other text input evaluations. Because of this, different phrase sets for text entry evaluation purposes have been developed and published. The most frequently used phrase set has been published by MacKenzie and Soukoreff in 2003 [61] and is called **T**ext **E**ntry **P**hrase **S**et (TEPS). It consists of 500 short and memorable sentences which were chosen as being as representative as possible to the English language. There are no punctuation characters or special characters included in that phrase set. The reason for that is that although the external validity increases when using punctuation characters MacKenzie and Soukoreff argue that the internal validity can suffer when including punctuation characters. They suggest excluding those characters for text entry evaluations if their activation does not differ much from the activation of the other letters: "For text entry, the mostly significant point of differentiation is the basic mechanism to enter letters, words, and phrases. If the techniques under investigation include the same mechanism to enter punctuation and other characters, then it is best to exclude these characters from the interaction, because they do not serve to differentiate the techniques. Instead, they represent an additional and undesirable source of variation "[61]. When using the algorithm of Mayzner and Tresselt [67] this phrase set reaches a correlation of 95% with the English language. However, Peak and Hsu [79] argued that the letter frequencies were collected from newspapers, magazines, and books published prior to 1966 and might not reflect today's style of writing.

Aside from the commonly used TEPS other phrase sets for text entry evaluation purposes also have been published. Kano et al [45] published a phrase set for children in 2006. The sentences do not contain offensive or complex words. Another example is the Enron Mobile Mail Dataset which has been created for the special purpose of mobile text entry [99]. Here, a database of real E-Mails, written and sent with Blackberry phones, have been analyzed.

13. Development of Text Input Concepts

This chapter summarizes the most important development steps related to text input conducted in the context of this dissertation. First, our initial study of developing or adapting text input concepts from literature to be used with the thumbs is described in Section 13.1. Then, the description of the three iterations of the stationary version of the Gestyboard is given in the sections 13.2, 13.3, 13.4. In Section 13.5 the development of the Gestyboard Backtouch, the mobile version of the Gestyboard, is explained.

13.1. Initial Study: Thumb-Based Text Input Concepts

In Section 12.2 we introduced a selection of interesting text input concepts developed in the last 20 years: Unistrokes and Graffit, T-Cube, Quikwriting, Swype, and SwiftKey. Swype and SwiftKey were not available at the time of our initial study. The remaining concepts were rated and compared by different criteria. The criteria are described in Section 13.1.1. Then this is followed by a section for each chosen and adapted text input concept (ThumbCube 13.1.1.1, Quikwriting 13.1.1.2, Unistrokes 13.1.1.3). At the end of this section our own concept is explained which is basically a simple split softkeyboard (Section 13.1.1.4). This study has been executed in collaboration with Daniela Korhammer in the context of her bachelor thesis [50].

13.1.1. Selection of Text Input Concepts

The following paragraphs describe the criteria on which the selection of the presented text input concepts from Section 12.2 were based:

Reachability and Scalability Because only a limited area can be reached with the thumb while holding the tablet in both hands, the reachability is an important criterion for the selection of text input concepts in our case. This is also related to its scalability. If the concept does not work below a certain visual and functional scale of the UI element used for text entry and if at the same time this low scale is necessary to be reachable by the thumbs at the bezel of the tablet, then it is not an appropriate text input concept for our setup. For example, a full QWERTY softkeyboard positioned on a bezel of the tablet needs to be very tiny to fit into the interaction area. This makes hitting the softbuttons almost impossible. Thus, the concept needs to work on a low scale.

No dictionary Some text input concepts are especially optimized for a special context in which the text input concept is extremely restricted - for example with respect to the available space. Although there is not a lot of space in our case either, because of the flexibility of the touchscreen and because there are two interaction areas, the limitation in our case is not as strict as it was the case for example for the first mobile phones with hardware buttons. To be able to use the same buttons for dialing as for text input, the multi tap text input technique was commercially introduced. For that, a button needed to be pressed one to four times depending on the button and the letter which should be typed. An optimization of multi tap was called T9 and used a dictionary to allow the user to only tap a button once. Similar to Swype, the system then chose the intended word from the dictionary. In a comparison, the users were much faster and made less errors with T9 [42]. But using a standard dictionary is not the ideal choice regarding the context of the MCI. The reason for that is there are many rescue related abbreviations and names for rescue resources which have to be introduced to the dictionary before they can be typed. In a discussion with the fire department at TUM we decided to go for a solution in which the user is able to type anything without depending on a dictionary. However, this does not mean that a dictionary is a bad choice in the context of the rescue service in general. A dictionary especially prepared for that context might also be a good solution. This needs to be tested in future studies.

Independence of the Environment Because the goal of this study is to find or develop a text input concept which fits to the two-handed interaction requirement, the environment influences the requirements for the text input concept. The concept is intended to be used in a mobile outdoor scenario. Because of that, the concept should not rely on sound feedback or on speech recognition (the environment can be noisy) and it should not rely on the orientation of the tablet like it is the case for example for TiltText [102] in which the tablet needs to be tilted to type letters.

13.1.1.1. ThumbCube

The first alternative chosen and adapted for our evaluation is ThumbCube which is based on T-Cube explained in Section 12.2.2. While for T-Cube the first step was to choose one of the 9 segments of a circle, we adapted the layout to better fit the interaction by thumb. In ThumbCube the sections are single quadrants placed in the interaction area of the dominant hand of the user.

A group is selected by simply touching it with the thumb. In our evaluation setup this element was 17mm wide and 92mm high, and thus reachable with the thumbs. We additionally decided to allow four directions (horizontal and vertical movements) instead of the eight used direction for T-cube. The reason for that decision was the assumption that the additional diagonal gestures would be too difficult to be executed with the thumb. Thus we simplified this technique for the special requirements related to the thumbs.

13.1.1.2. Modified Quikwriting

We modified Quikwriting also in a way to better fit the requirement to be used with the thumb. We also considered the low sensitivity of the ruggedized tablet devices (see Section 5.3 for details) used in our studies. Gestures requiring long continued touches with the screen should be avoided because the users tend to release a little bit of the pressure

Figure 13.1.: The initial visualization of our adapted version of T-Cube called ThumbCube

while sliding and this frequently leads to the system's loss of the tracking of the finger. Because T-cube is designed to especially support continued sliding gestures we modified it to additionally allow for releasing the thumb from the screen to confirm the selected letter. In T-cube a letter was only confirmed when the stylus was re-centered. Other than that the functionality remains the same as for T-cube.

13.1.1.3. Unistrokes

For Unistrokes, we implemented a simple gesture recognition system. No modifications were made to the original concept. An explanation of the original concept can be read in Section 12.2.1.

13.1.1.4. Split Softkeyboard with the Fitaly and Alphabetic layout

Additionally to the mentioned adapted text input concepts, we also implemented a standard softkeyboard which can be used by tapping softbuttons. Because a keyboard containing each letter in the alphabet does not fit into one bezel we split the softkeyboard: The left half of the keyboard is placed on the left bezel while the right half of the keyboard is placed on the right side. But we decided to use the Fitaly ([31], [88]) layout instead of the well-known QWERTY layout for this split keyboard. There are two reasons for this decision. First, using the QWERTY layout would give this concept an unfair advantage when compared to the other concepts because the QWERTY layout is well-known. That would falsify our results because a known (just modified) version system would be compared against text input concepts not known by our participants. With a different layout we could avoid this falsification. Another reason is the chosen layout itself. This layout is especially optimized for usage with one finger. It is designed to reduce the distances between frequently needed keys. Figure 13.2 shows the split softkeyboard based on the Fitaly layout.

(a) Left Bezel (b) Right Bezel

Figure 13.2.: The Split-Keyboard with the Fitaly layout

As another alternative, we also used the same split softkeyboard with an alphabetic layout. For an alphabetic layout, people still have to spend more time searching letters when compared to the QWERTY layout. But because it uses the alphabetic system we expect our participants to find the letters faster as for the Fitaly layout. But once people are trained with both concepts, the Fitaly layout should be much more efficient due to its one-finger-usage optimization. However, in the context of an MCI one cannot expect that each person interacting with the interface has prior training with the text input concepts. This is the reason why we also integrated the alphabetic layout in our evaluation setup.

13.1.2. Evaluation

In this section the evaluation details are given. First, in Section 13.1.2.1 the participants of this study are described. Next, in Section 13.1.3 the procedure of this evaluation is given and this is followed by a brief explanation of the apparatus used for this evaluation.

13.1.2.1. Participants

Five test users participated in this study (2 female). All participants were students aging between 22 and 23 years at the time of the evaluation. Three of the test users had a technical background (mechanical engineering and bio informatics), one participant studied medicine and one studied sports science. All participants were right-handed. Consequently, except for the split keyboard each alternative was positioned on the right bezel of the tablet. The split keyboard needed both, the left and the right bezel anyway. This study was conducted in 2010 when smartphones were not that ubiquitously available as they are nowadays. For this reason none of the participants had prior regular experience with the usage of touchscreens. Their experience was limited to the use of automated teller machines (ATMs) and similar public systems equipped with touchscreens.

13.1.3. Procedure

Considering the error rate, we decided to use the simplest option: When a wrong letter was entered by a participant, the error counter was incremented and the wrong letter did not appear. The text continued only when the user entered the correct letter. This way, there was no need for the participants to correct their errors. As described in Section 12.4.3.3, this did not represent the natural way of writing and also did not allow a detailed analysis of the sources of errors. However, it was not the goal of this evaluation to get a detailed analysis of the error rate but rather to compare the performance of the different text input techniques when using them for the first time.

The participants typed four randomly chosen sentences from TEPS [61] and additionally the pangram "The quick brown fox jumps over the lazy dog." followed by two German sentences with each presented text input technique; seven sentences in sum. The reason for this short number of sentences was that we were interested in the immediate performance of the text input techniques chosen for this study. This followed the requirement to enable even those users in an MCI to interact with the MCI-application that have had no experience with the software. A within-subject design was used for this evaluation.

13.1.4. Apparatus

We used the xplore IX 104 since this was the target device which was intended to be used in a real MCI. Figure 5.2 A) shows the device. For this initial study, we developed a simple tool to log the performance (error rate, speed) and to present the text to be transcribed by the participants. Figure 13.3 shows the UI of this tool. Field **(1)** presents the text to be transcribed and field **(2)** shows the text entered successfully. The button labeled with **(3)** could be used to pause the evaluation. Field **(4)** showed a preview of the next letter and field **(5)** showed the current WPM and the number of errors made so far. The button labeled "START" **(6)** started the evaluation.

After each typed sentence, the evaluation was paused automatically and got reactivated as soon as the participant started typing the next sentence. Each participant was informed that feedback could also be given during the evaluation but only when it was paused to avoid falsifying the results. When a participant finished the evaluation with each alternative a short interview followed to also get qualitative feedback. For this interview, a guideline for conducting qualitative interviews in the context of UIs developed at the Technische Universität München was used [73].

13.1.5. Results

In this section the results of this initial text input evaluation are summarized. Section 13.1.5.1 presents the comparative results for the speed and Section 13.1.5.2 presents the error rate for each text input technique. Afterwards, in Section 13.1.6 discusses the qualitative results obtained from the interviews for each technique.

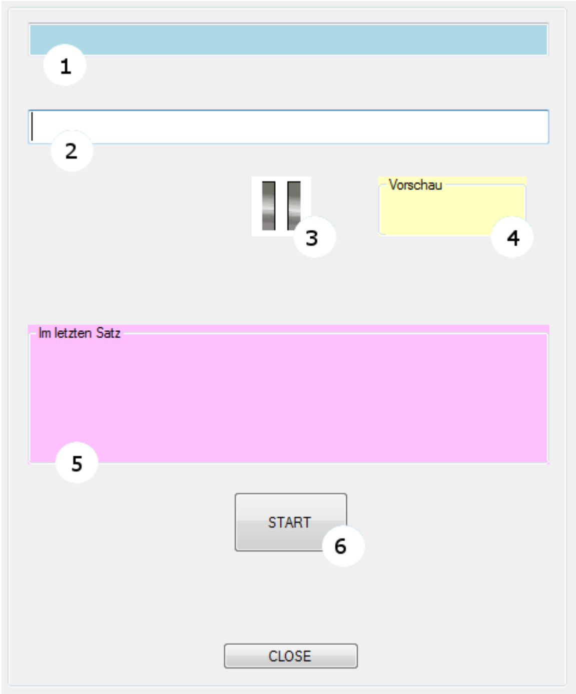


Figure 13.3.: The UI presented to the participants during the evaluation

13.1.5.1. Speed

Figure 13.4 shows the results for the average speed of the participants for the five English and the two German sentences. Additionally, the speed of the fastest and the lowest typers are visualized by the black line on the bars. The German sentences additionally included special characters. Because of that, it took more time to type the German sentences than the English sentences on average. The participants were much faster with the split standard keyboard with an alphabetic layout than with the other alternatives. With a speed of 7.5 WPM, the slowest typer of the alphabetic layout was still faster than the fastest typers of Quikwrite and Unistrokes. As expected, the split keyboard with the Fitaly layout was also slower than the alphabetic layout but it was still faster than any of the gesture based text input concepts tested in this evaluation. The fastest concept among the gesture-based concepts was the ThumbCube.

13.1.5.2. Error Rate

The ticker in our system only continues when the correct letter is typed. This reduces the type of errors which can be made by the user. This depth of analysis is sufficient for the initial evaluations steps. But for a detailed analysis of a text input concept the errors should be classified like it has been described in Section 12.4.2.2. This approach has been chosen for the third iteration of our Gestyboard text input concept in Section 13.4. Figure 13.5 shows the median of the error rate for the English and German sentences of this initial development step.

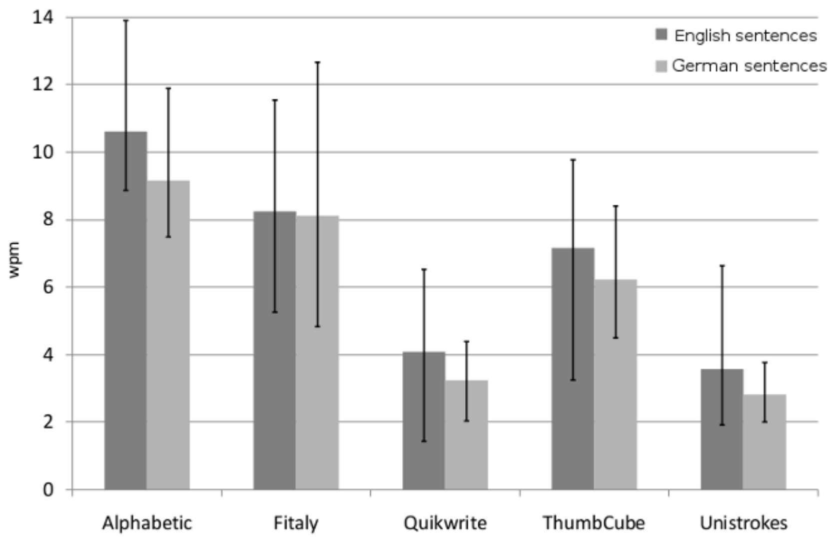


Figure 13.4.: The average speed of the English and German sentences in WPM

The fewest number of errors, less than 2,5%, were made with the two concepts which are based on softbuttons and a split layout: alphabetic and Fitaly. Much more errors was made with the gesture based text input concepts. The text input concept with the most errors was Unistrokes with an error rate of more than 30% for English sentences and almost 50% for German sentences. The reasons for these results are discussed in Section 13.1.6.

13.1.6. Conclusion and Discussion of the Initial Text Input Evaluation

The analysis of the conducted interviews revealed the reason behind the presented results. Considering the speed, the classical touchscreen keyboards were more efficient than the gesture based concepts. One reason for that is that the gesture based text input concepts presented in this evaluation need a learning phase before users can start typing with them efficiently. In this initial evaluation we focused on the immediate usage of those concepts. In the case of Unistroke, it has to be mentioned that many gestures were not correctly interpreted by the system. The implementation of the gesture recognition algorithm was not reliable enough. Thus, the results for Unistrokes do not reflect its real capability. But still, when considering the intuitivity and the capability to be used without any prior knowledge, the simplest solution is to split the standard keyboard. Additionally, the layout can then be changed to the standard QWERTY layout to further increase the typing speed of the users.

13.2. First Iteration: Gestyboard 1.0 - Stationary Text Input

As mentioned in the introduction of this part of the dissertation, the Gestyboard concept is based both on gestures and the 10-Finger-System. It was designed according to the requirements given in Section 11. The Gestyboard concept and the development steps related to

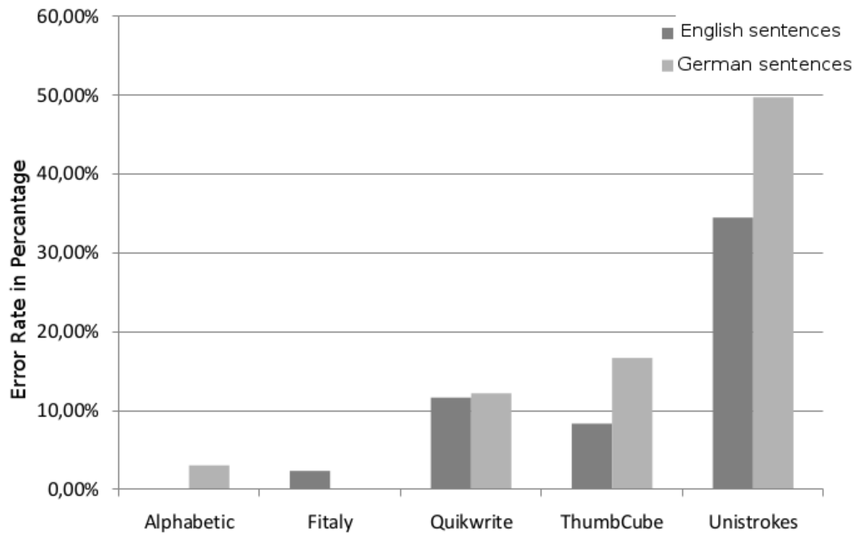


Figure 13.5.: The median of the error rate of the English and German sentences

the Gestyboard have all been performed in the context of this dissertation. This section explains in detail how the stationary version of the Gestyboard works and presents the evaluation and the results of its development iterations. This first iteration of the Gestyboard development has been published in [25] as a poster and in [24] as a technical report.

13.2.1. Gestyboard 1.0

This section introduces the initial Gestyboard concept. First, the activation gesture of the Gestyboard text input concept is explained in Section 13.2.1.1. This is followed by the explanation of single keystrokes in Section 13.2.1.2.

13.2.1.1. Activation of the Gestyboard

To start the text entry process, an activation state is needed initially for the Gestyboard. When the user touches the multitouch surface with all 10 fingers the keyboard is activated and a visual feedback is displayed on the touchscreen (see Fig. 13.10). The central row of keys (‘a’ ‘s’ ‘d’ ‘f’ and ‘j’ ‘k’ ‘l’ ‘;’) are placed at the position of the user’s fingers with a QWERTY layout and are activated by tapping them with the associated finger according to the 10-Finger-System. By sliding a finger horizontally, vertically or diagonally, the user is able to select and activate the direct neighboring keys of the central ones. Hence, a group of keys is assigned to each finger (a central key plus two or five direct neighboring keys).

13.2.1.2. Keystroke Activation Gestures

Each finger can activate one and only one key group assigned to it, based on the 10-Finger-System. Figure 13.7 shows the conceptional layout of the keys. When the keyboard is activated, the keys ‘a’ ‘s’ ‘d’ ‘f’ are assigned to finger 1 to 4 and and the keys ‘j’ ‘k’ ‘l’ ‘;’

Figure 13.6.: The numbering of the fingers in order to be clearer by referencing them.

to fingers 7 to 10, respectively. The space key is assigned to the fingers 5 and 6 (thumbs) and appears at their position in the visualization. It is important to mention that all these keys are not virtual buttons, i.e. they are not activated by a finger touch. In fact, their only purpose is to visualize the QWERTY layout to guide the users while typing if they do not master the 10-Finger-System.

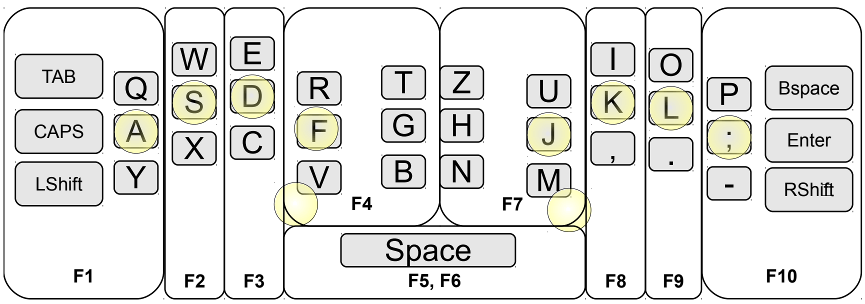


Figure 13.7.: The conceptual layout of the Gestyboard. Fx : Finger x

The Fingers' Gestures for the Home Row Keys Tapping with a finger always activates the central key assigned to that finger, whether or not the tap occurs within the corresponding square of the visual representation, e.g. a tap of finger 1 always activates the key 'a'. Even if a user taps on key 'q' with finger 1, key 'a' will still be activated because the tap represents a unique gesture for this finger. The same holds for key 's' and finger 2, key 'd' and finger 3, and so forth.

The Fingers' Gestures for the Neighboring Keys : Upper and Lower keys The upper keys ('q' 'w' 'e' 'r' and 'u' 'i' 'o' 'p') are activated by performing a sliding-up gesture with the corresponding finger. For example, to activate the key 'q', the corresponding finger, finger 1 in this case, has to be slid upwards on the screen. If the finger displacement exceeds a predefined threshold in the upper direction, the letter 'q' will be activated and hence typed. The same procedure is used for the activation of the lower keys ('y' 'x' 'c' 'v' and 'n' 'm' ',' '.'). Again, there is no need to physically hit the corresponding visual representation of the desired keys, the sliding gesture upwards or downwards with the associated finger defines a unique keystroke activation.

The Fingers' Gestures for the Vertical Central Keys The keys 't', 'g', 'b', and 'z', 'h', 'n' are special cases, since they add extra gesture overhead to fingers 4 and 7. Finger 4 has to be slid to the right and finger 7 has to be slid to the left to activate the key 'g' and 'h' respectively. To activate the keys which are positioned diagonally, a diagonal finger sliding to the corresponding direction has to be performed.

The Fingers' Gestures for Space, Enter, Backspace and Shift Performing a tap gesture with finger 5 or 6 (thumbs) activates the space key. The 'Enter' key is activated by a sliding gesture to the right with finger 10, while the 'Backspace' key is activated by sliding the same finger diagonally to the upper right. Lastly, the 'Shift' key can be activated with either finger 1 (diagonal sliding to the lower left) or finger 10 (diagonal sliding to the lower right). An overview of all the gestures is given in Figure 13.8.

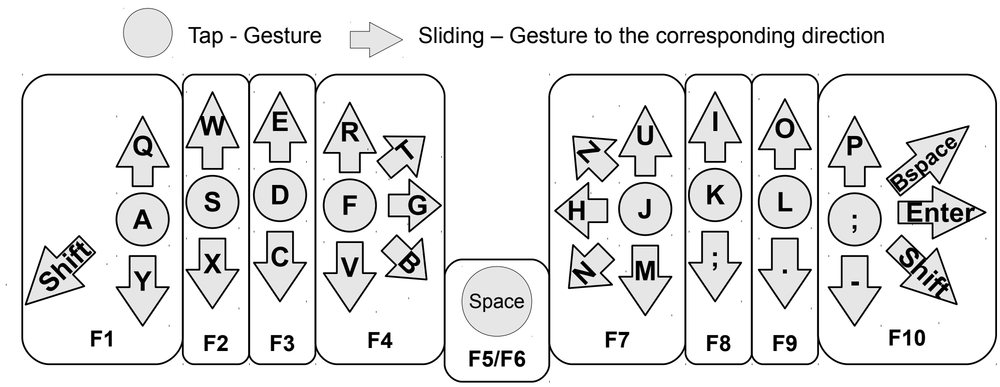


Figure 13.8.: The overview of the gestures. Fx : Finger x.

13.2.2. Added Value of the Gestyboard Concept

The rigorous requirements that are followed in the Gestyboard design (see Section 11), are the main advantages of the concept. **R1** is fulfilled, since no additional hardware like cameras or digital gloves is needed. Additional hardware is expensive, reduces flexibility, and in some cases requires additional effort for calibration. **R2** is also fulfilled, since the Gestyboard is not based on any dictionary or database. We consider that an advantage, because the user should be able to type everything, even mixed languages or slang, just as on a classical keyboard. Of course, text entry performance can be dramatically improved with a dictionary and with word prediction. However, this is more appropriately done by the applications themselves. **R3** cannot be verified with the current user evaluation results, since the users did not have as much training with the Gestyboard as they had with the classical keyboard. In order to attain high speeds with the Gestyboard, users need time to get familiar with the new finger gestures. We believe, that people can get used to these gestures, seeing that piano players are able to learn and perform much more complicated finger movements. R4 is however fulfilled by the concept itself. This is due to the unique finger gesture key mapping system. If the user for example wants to activate the key 'q', finger 1 has to be slid upwards. In the previous example, the upward movement does not need to be perfectly vertical, since there is no risk to activate another nearby key with the same gesture. Typing errors can of course still occur if the wrong finger is used, or by moving multiple fingers at once accidentally. Fulfilling R5 might be one of the most important advantages and an expected outcome of the 10-Finger-System: We expect that it is possible to type text blind (touch typing) once a user gets used to the finger gestures. For beginners, the visual representation of the keys is quite important to actually guide them while typing. We also expect that the longer the users use the Gestyboard in their daily life, the higher the chance will be that they do not rely on the visualization of the Gestyboard keys anymore.

13.2.3. Special Challenges

This novel approach also leads to some new special challenges. The users first have to understand the new input mechanism. To help them, we provided visualization for the fingers' gestures always representing the current state of the Gestyboard. This is one of the 10 usability heuristics of Nielsen [75] that each UI should fulfill. Another challenge is the human fingers' ergonomics. Sometimes it is not easily possible to move one finger independently from the others, whether intentionally or not. When one finger is slid back to the center again, the other fingers of the same hand will also slightly deviate from their centered position.

13.2.4. The First Prototype of the Gestyboard

To get the first impression of the Gestyboard concept, prototype version 1.0 was developed using Microsoft Windows Presentation Foundation framework (WPF). In this version, the basic gesture detection algorithm was implemented as a state machine for each finger which determines whether a sliding or a tap gesture is performed. The direction of the gesture was not yet integrated in the state machine. Instead, a hit event was triggered once a key visual representation was touched as a way to determine the direction of the sliding gestures. A desired key activation with a sliding gesture occurred when the following conditions were satisfied: 1.) a sliding gesture was detected; and 2.) a hit event was triggered; and 3.) the correct finger id according to the 10-Finger-System was used. If all three conditions were satisfied, the desired unique finger gesture-to-key mapping was triggered. This allowed for fine-tuning the gesture-recognition visually by positioning the keys in an optimized way, according to the results of the first evaluation. When a finger was released from the screen, its last position was stored. When the finger touched the screen again, its new position was compared with the previously stored one. If the distance between those 2 points is less than a specified offset, then the system considers this finger to be the one released earlier. Otherwise, the finger was simply ignored by the system in this first prototype. The users had to try to reposition their fingers within an offset from their initial position on the screen. Those offsets were visualized as rings and were visible at all times on the screen as long as the Gestyboard was active (see Fig. 13.9).

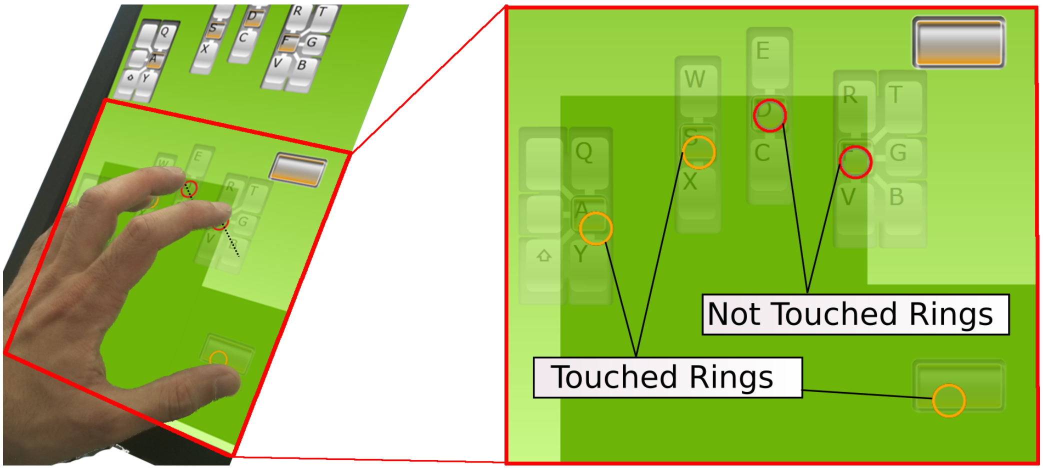


Figure 13.9.: Prototype 1.0 of the Gestyboard - The rings visualize the last fingers' positions known by the system. Fingers 1,2 and 5 are touching the screen, while 3 and 4 are hovering above it.

13.2.5. The Visualization of the First Prototype

The visualization of the "Gestyboard" included both: the direct key visualization and an offset visualization. The direct key visualization was placed directly below the user's fingers and might be more intuitive, since a one to one mapping was used here requiring less mental overload of the users compared to the offset visualization. The latter shows the same version of the Gestyboard shifted by a y-offset. In this case, users have to map their finger movements to the offset visualization of the keys, which might require more mental effort. Yet as an advantage of the offset visualization, the user's hands do not occlude the keys visualization. We decided to provide both visualizations to the user to find out their favorite.

The direct visualization nearly is transparent (alpha = 0.2) making it less pronounced. Except for the different level of transparency, both visualizations render exactly the same thing; as demonstrated in Figure 13.10.

To let the user know where the fingers are relative to the Gestyboard, a slider is displayed with each group of keys. The slider represents the current position of the fingers within the Gestyboard. Thus if, for instance, an 'h' is typed the slider is placed above the key visualization of the letter 'h'. The key label is still visible, because of the transparency of the ring slider (see Fig. 13.11).

There is a slider for each finger. Consequently, multiple sliders can be moved simultaneously, similar to pressing multiple keys on the classic keyboard. The letters are also typed in the same order as they are activated by the user. The only difference to the classical keyboard is that there is no auto-repeat feature for keys when holding a finger on a key. This will be added in future work. For our studies with the Gestyboard, this feature is not needed. Another important note considering the visualization is that the concept itself does not rely on the visualization. In principle it is possible (and even intended) to use the Gestyboard without any visualization. However, the visualization might be very

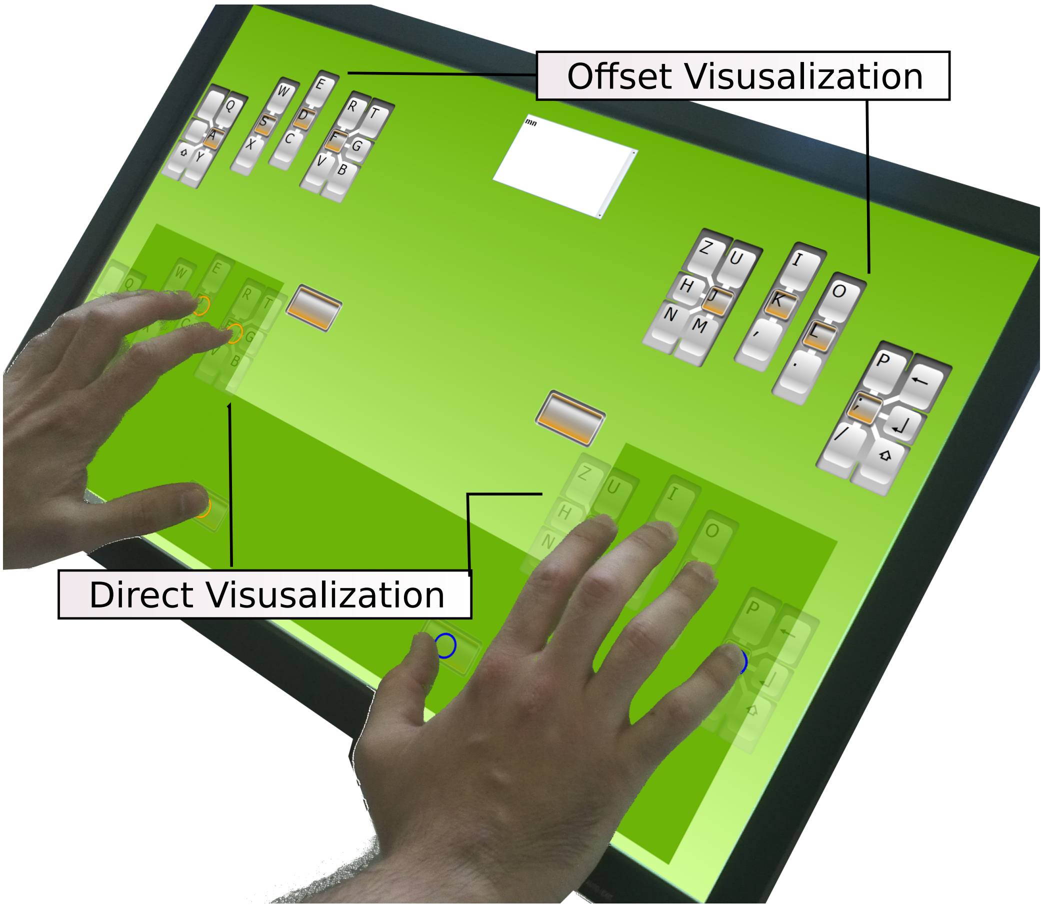


Figure 13.10.: Two visualizations of the Gestyboard are provided to the user. Direct: Appears directly below the user's fingers. Offset: An additional visualization with a y-offset to 1.) avoid occlusion issues; and 2.) find out which visualization is preferred.

important especially for beginners to understand how the system works.

13.2.6. Evaluation

We performed a within-subject evaluation to compare our Gestyboard (G) with both a virtual touchscreen keyboard (T) and a classical hardware keyboard (C).

13.2.6.1. Participants

We evaluated our first Gestyboard prototype with 41 pupils from a junior high school, 26 males and 15 females, all right-handed. All of them had been taught the 10-Finger-System in school on a standard computer with a classical hardware keyboard for at least 2 years. It was important that all our participants knew the 10-Finger-System as this was an essential prerequisite skill to use the Gestyboard efficiently. We evaluated the system with pupils

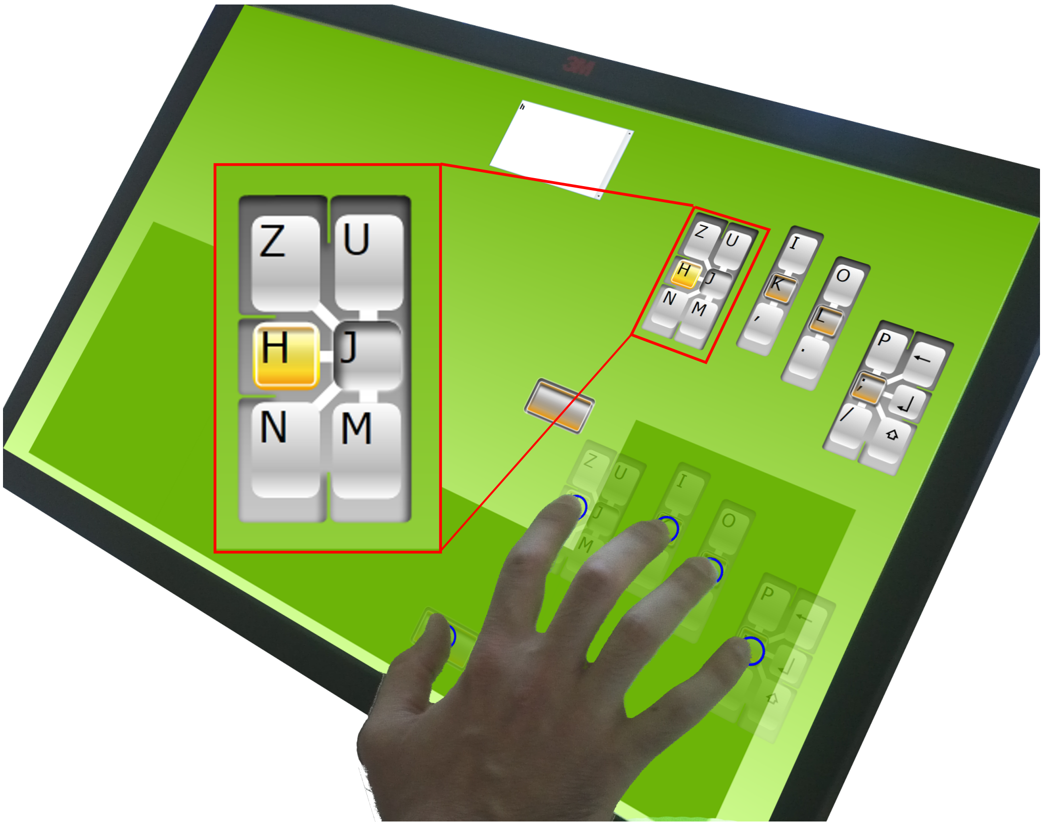


Figure 13.11.: The current position of each finger relative to the Gestyboard is visualized by a slider (see Letter 'H').

from different class levels to see whether there was any difference in the performance between beginners and more experienced typists. The youngest among them had been taught typing at school for 2 years, these were 21 pupils aged between 12 and 14 (group 1). The school requires that the pupils of this level have a typing speed of 80 characters per minute (16 WPM). Additionally, 6 pupils had a learning experience of 3 years (age 14-15) with a typing speed of 90 characters per minute (18 WPM) (group 2) and the last 14 had an experience of 4 years (age 14-17) with a typing speed of 100 characters per minute (25 WPM) (group 3). We selected participants with various typing skills in order to have a large variety in terms of experience.

Of the 41 participating pupils, 29 owned a touch device such as a mobile phone in 2011. Any such kind of touch devices was used at least once a week by 25 pupils, 9 pupils used these devices rarely (less than once in a week) and 7 had never used any touch device at all. 29 pupils used computers daily and 34 typed a text more than once a week. Text messages were written by 21 of the pupils on their mobile phones at least once a day, 4 had never written any text messages.

13.2.6.2. Apparatus

The tests were conducted during normal class hours at the school in a separate room next to the normal class room. We had six different stations for six different tasks. These included three typing stations in order to compare the three different keyboards: The hardware keyboard station consisted of a normal computer monitor and a standard hardware keyboard with German Layout (positions of 'Y' and 'Z' are swapped). For the virtual touchscreen keyboard we used a ruggedized tablet PC with dual touch input. The Windows 7 on-screen keyboard was used for text input on the tablet PC. The Gestyboard was shown on a 3M multitouch 22" monitor and a multitouch capability of more than 20 fingers at a time. As a software we used Tipp10[1], which is an open source typing tutor. We created our own typing lessons in Tipp10 - one for trying out the Gestyboard which consisted of two sentences and another one for the actual evaluation. The latter lesson consisted of seven English sentences and an English pangram, containing every letter of the alphabet. The complete text sums up to 248 characters. The seven sentences were chosen from MacKenzie's phrase set for text-entry evaluation purposes (TEPS) [61]. The frequency of each letter in each sentence represents the frequency of the letter in the English language in general. Although the current prototype allows use of the shift key by combining the shift key gesture with any other key gesture, uppercase letters were removed from the sentences in order to focus on the gestures to be activated with just one single finger gesture. To type uppercase letters, the user would had to perform both the gesture for the shift key and the gesture for the desired letter. This feature was kept out of the first evaluation to avoid the additional complexity. It is planned to investigate two-level gestures in details in future work.

Tipp10 presented the text to type in a continuous text ticker. The speed of the ticker was controlled through typing. The ticker stopped when a wrong character was pressed and proceeded again when the expected letter was typed. We measured the following data with Tipp10: the duration to type the eight sentences, the overall number of errors and the number of errors per character. As a consequence of this evaluation design, there was no need for the participants to correct errors. Although this does not reflect a natural scenario, it made sense to focus on the typing task itself in this first iteration. Errors were then allowed later in the third iteration described in Section 13.2.

13.2.6.3. Procedure

In order to make the evaluation as efficient as possible, we tested pupils in groups of three - each supervised by a different instructor. The following six different tasks were executed by each participant:

1. Introduction + demographic data: We introduced the participants to the evaluation environment and collected demographic data
2. Ergonomic test: A test to evaluate the user's ability to move single fingers separately
3. Text entry with the standard hardware keyboard

[1] `http://www.tipp10.com/`

4. Text entry with the virtual touchscreen keyboard
5. Learning phase of the Gestyboard with two practice phrases only, followed by the evaluation phase of the Gestyboard
6. Interview and Feedback: Qualitative feedback about the preferences and the problems encountered with the different keyboards

Each task was explained beforehand to the participants. We demonstrated first how to type with Gestyboard. A learning phase followed, where they wrote two practice sentences.

In summary, the design of the study was as follows:

3 techniques x 248 characters = 744 characters per person;

744 characters per person x 41 persons = 30504 characters entered in total;

Additionally 2 practice phrases with 62 characters x 41 persons: 2542 characters

13.2.6.4. Hypotheses

Our hypotheses before the evaluation were the following:

- **(H1)** The hardware keyboard is faster in terms of typing speed than the virtual touchscreen keyboard, and the virtual touchscreen keyboard is faster than the Gestyboard. We expect this due to the fact that the hardware keyboard is the one with which most people are familiar, the virtual touchscreen keyboard has the disadvantage of lacking haptic feedback and Gestyboard is a completely new input mechanism.
- **(H2)** 'a', 's', 'd', 'f', 'j', 'k', 'l' are the characters with the lowest error rate. We predict this because the fingers are initially placed on these characters in the initial typing position.
- **(H3)** The characters 'z', 'h', 'n', 't', 'g', 'b' are most error-prone due to their position on the keyboard. Only both index fingers (finger 4 and 7) are in charge of typing these 6 letters; each index finger has to move vertically, horizontally and diagonally to type these letters.
- **(H4)** The letters 'n' and 'b' have a higher error rate than 't' and 'z', because the index finger has to perform a diagonal movement down to type 'n' and 'b'.

13.2.7. Results

The typing speed, the total error rate and the error rate per character were calculated using the values which were measured with Tipp10. A correlation analysis was performed in order to find out whether for example, the typing speed of the touch keyboard correlates with the typing speed of the Gestyboard. Our results of the the evaluations are presented below.

13.2.7.1. Typing Speed

Although in literature WPM are usually used to measure the typing speed, we calculated the typing speed as characters per minute (CPM) through this formula: $\frac{total\ characters}{total\ time\ [in\ minutes]}$. The reason behind this was that the pupils worked with CPM in their school and Tipp10 also showed the current and average speed in CPM.

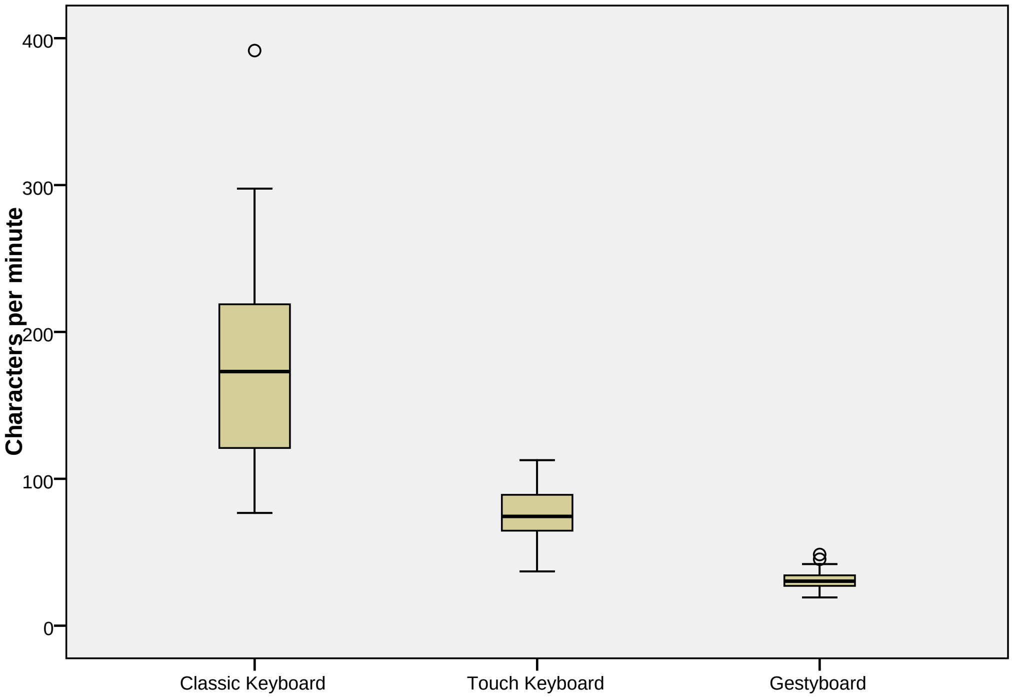


Figure 13.12.: Characters per minute with 3 different keyboards. Whiskers represent 95% confidence interval.

Our results show that all of the pupils had a faster typing speed with the classical hardware keyboard than required from the school (group 1 had an average typing speed of 137.66 CPM (required is 80 CPM), group 2: 168.84 CPM (required is 90 CPM), group 3: 225,47 CPM (required is 100 CPM)). The overall average text entry speed is: 174.41 CPM with a $STD = 68.75$ for the classical hardware keyboard, 75.79 CPM with $STD = 17,51$ for the virtual touchscreen keyboard and 31.03 CPM with $STD = 6.81$ for Gestyboard, see also Figure 13.12.

The classical hardware keyboard is significantly faster than the virtual touchscreen keyboard with $t(36) = 10.07$, $p < 0.01$ and the virtual touchscreen keyboard is significantly faster than Gestyboard with $t(36) = 14.20$, $p < 0.01$. Hence, H1 is verified.

13.2.7.2. Total error rate

Errors are counted per character and not by the number of wrong inputs. If the user typed more than one wrong character in sequence, it was counted as one error for the specific character. For example if, for the correct sentence:

see you later alligator

the user typed:

*se**df**e **as**you lat**w**er alligator*

These are in total 3 errors (2 errors for letter *'e'* and 1 error for letter *'y'*).

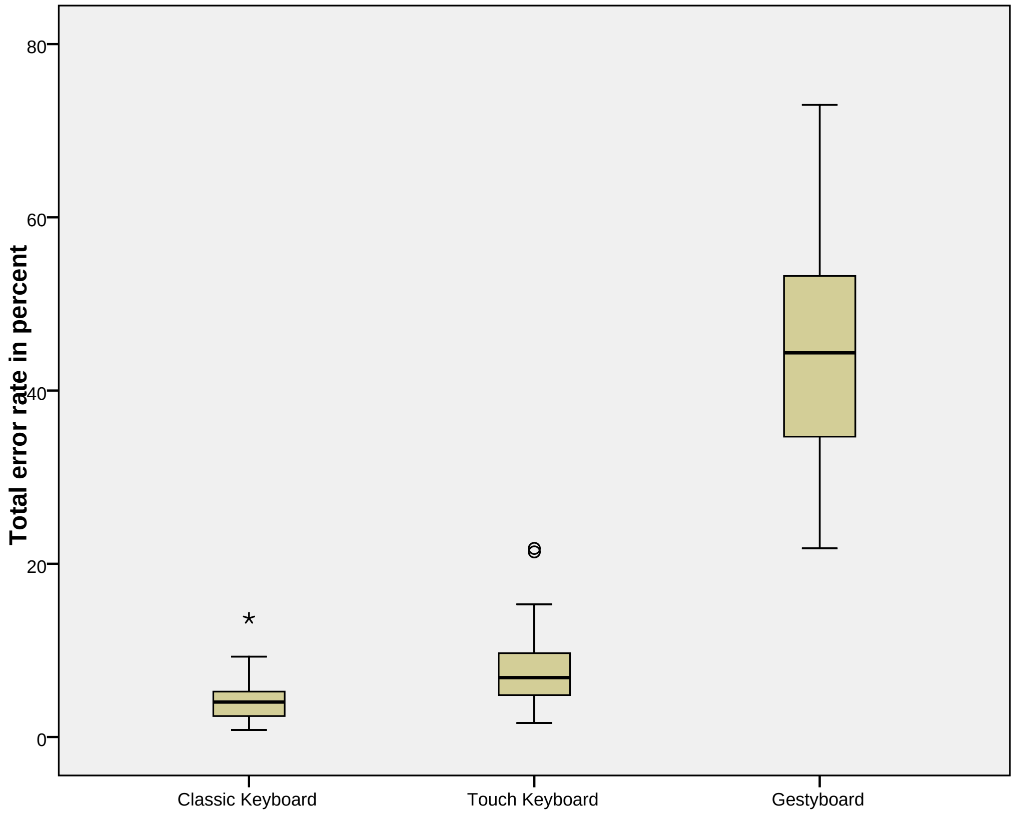


Figure 13.13.: Total error for 3 different keyboards. Whiskers represent 95% confidence interval.

We obtained the following mean error rate for the three keyboards: For the classical hardware keyboard 4.32% with $STD = 2.60$, for the virtual touchscreen keyboard 7.90% with $STD = 4.88$ and for the Gestyboard 45.23% with $STD = 13.75$, see Figure 13.13. The classical hardware keyboard and the virtual touchscreen keyboard were significantly different $t(36) = -4.74$, $p < 0.01$, the same is also the case for the virtual touchscreen keyboard and Gestyboard $t(36) = -15.56$, $p < 0.01$. There was however no significant difference, in error rate nor in speed between the three different age groups, who had unequal experience with typing.

13.2.7.3. Error rate per character

In addition to the total error rate, the error rate per character was also analyzed. Figure 13.14 shows the mean values of the error rate in percentage over all participants. Each character is ordered as in the Gestyboard layout in three rows and two groups, for the left and the right hand. The height and the color encode the error rate of each character. The color coding goes from light green for a low error rate to a dark red for a high error rate.

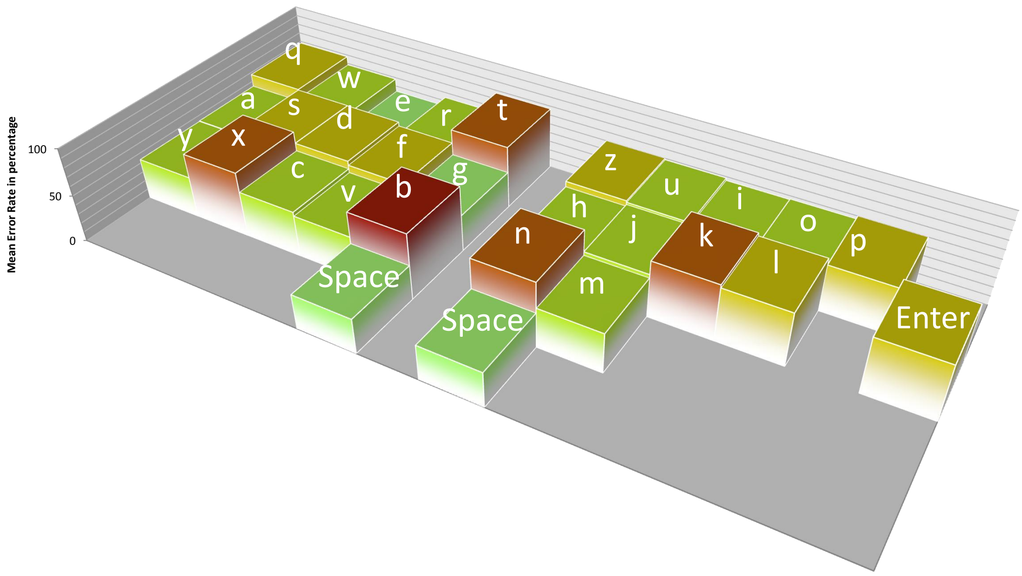


Figure 13.14.: Mean error per character for Gestyboard input.

The mean error rate for the upper row is 49.0%, for the middle row is 46.0% and for the lower row is 55.4%. What we can say about H2 is that not all of the horizontal central keys from the initial middle position ('a' 's' 'd' 'f' and 'j' 'k' 'l') are easier than the rest of the characters, especially 'k' (60.25%), one of the direct tapping keys, is nearly as error-prone as 'enter' (59.61%) which requires a right sliding gesture of the little finger. This result is also represented in our statistics as we have no significant difference between the initial direct tapping characters and the neighboring sliding letters.

Concerning H3: The index finger letters, which are not directly under the finger ('z' 'h' 'n' 't' 'g' and 'b') have a mean error rate of 56.65%, whereas the rest of the characters have a mean error rate of 46.27%. A paired sample t-test showed values of $t(5) = 2.5$ with $p < 0.05$. We have a significant difference between the two groups of letters. Thus, when taking the mean values into consideration, H3 is true. However, the letters 'g' and 'h' were less error-prone than the remaining letters of this group. Consequently, H3 is actually true for the letters positioned diagonally to the index fingers. Vertical sliding gestures are not more error-prone than even the home-row keys (see Figure 13.14).

The five toughest characters were 'b' (70.76%), 'n' (69.23%), 'x' (69.23%), 't' (66.74%), 'k' (60.25%). Least problems appeared for 'e' (33.72%) and 'space' (37.21%). The most problematic characters 'b' and 'n' are typed both by a diagonal down sliding movement. 't' (66.74%) and 'z' (50.0%) which are written by a diagonal up movement of the index finger have lower mean error rates, however the values are not significantly different from 'b' and 'n' as we suggested in H4.

13.2.7.4. Correlation analysis

A correlation analysis and a following bivariate linear regression analysis showed that there is a significant linear dependence between the error rate and the characters per minute with the Gestyboard ($p < 0.01$), with a regression line of $y = 40.16 - 0.203x$, see Figure 13.15. The correlation is a weak negative relationship. This means, that if the error rate increases, the typing speed falls. We suggest, that the users who had a lower error rate adapted quicker to the Gestyboard and could type faster than those who had problems using the Gestyboard.

Pearson's correlation coefficient is $r = 0.402$ and the coefficient of determination that will give some information about the goodness of fit of the model, shows a distribution of $r^2 = 0.162$ or 16%. This means also that 84% of the variance in the diagram is influenced by other variables not part of the model. Hence, we ran a multi regression analysis in order to find other influencing variables. We tested gender, class level, error rate of the touch keyboard, error rate of the classical hardware keyboard as well as the multitouch experience. However, none of them showed a significant difference. Neither had the usage of the mobile phone, keyboard or computer or the experience with the computer any significant influence.

13.2.7.5. Qualitative Feedback

We asked the participants to express their suggestions for improvement, further usage, difficulties, preference, physical and mental demand as well as whether their knowledge of the 10-finger-system has some benefit or not. This interview was conducted after the evaluation with each participant individually.

Improvements and difficulties Several different categories were mentioned for improving the Gestyboard.

Movement The sliding gesture for the upper and the lower row as well as performing two different gestures, sliding and tapping, were mentioned frequently as being problematic. The movements were uncommon and therefore not that much liked. Especially sliding had the disadvantage that it is not always possible to move all fingers individually without moving other fingers too. This resulted in mistyping. Some of the pupils would therefore prefer tap movements instead of the combination of both movements. Some subjects stated that they had less problems with the movement after practicing it for a while. The pupils reported that their motivation increased

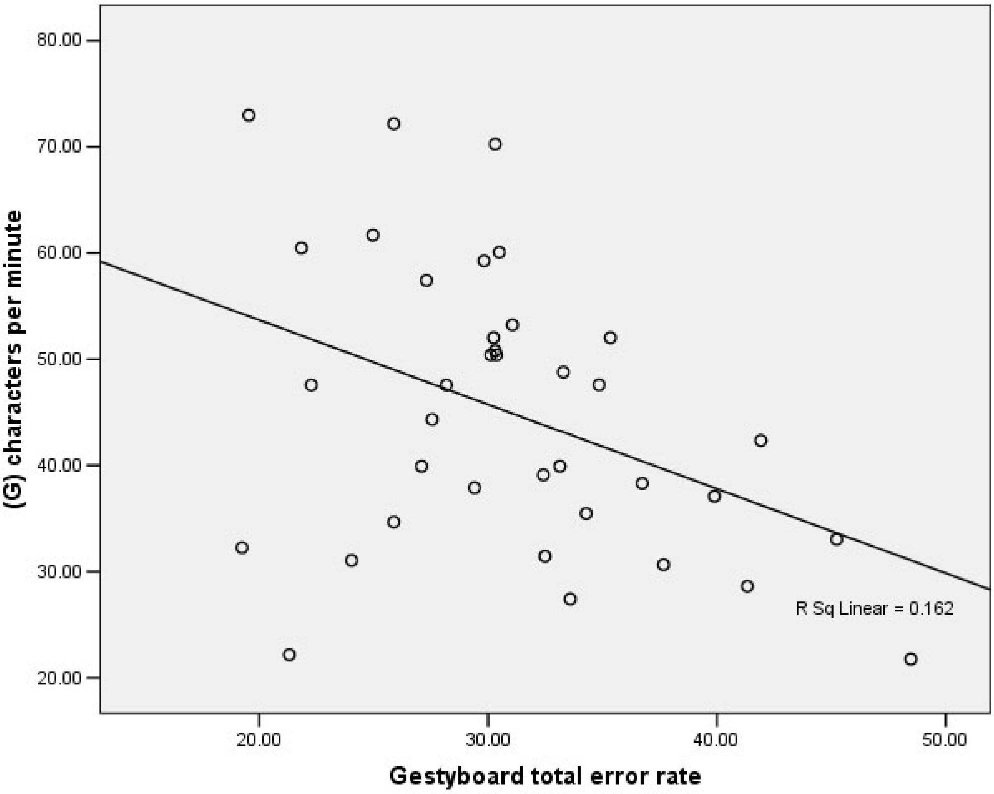


Figure 13.15.: Scatter plot showing the correlation between error rate and characters per minute with the Gestyboard.

during the test to get even better after some more training. Interestingly, we got also some sporadic answers saying that the movement was not a problem at all. During our test we had also the chance to speak to a teacher, who teaches the pupils typing. We got very valuable feedback concerning the two different movements. She explained that besides the standard process of typing with a classic keyboard, the Gestyboard requires, that the typist has to differentiate for each letter whether a sliding or tapping movement has to be performed. This adds additional mental load for typing with the Gestyboard.

Automatic reset to initial position Furthermore, the pupils mentioned that it is cumbersome to push the sliders of the Gestyboard upwards to type any key in the upper row or downwards to type any key in the lower row and then drag it back again to the initial position in the middle row. An automatic reset is preferred, which sets the slider automatically back to its initial position when it is released somewhere else than there.

Difficult keys It was also mentioned that some keys were more difficult than others. Especially the movement of the little finger caused problems. This finger is also used for the Enter key. Furthermore, also the space key was mentioned as being problematic. Both keys appear very frequently in a text and we are investigating other specific gestures or placements in order to type them easily.

Typical statements from the users were:

"Typing is strange, because with the classic keyboard one normally pulls the finger up."

"...But the switch between tapping and sliding is not that good."

"There is a ring under my finger, if this is away I need to click once more."

"I think it is better, if it is just tapping like on a normal smartphone."

Physical demand We asked the users whether typing was exhausting for their hands and whether they felt any pain in their hands after the session. The answers did not show a clear direction. Some subjects reported their hands' fatigue, while others did not. After further analysis we observed that, more female users mentioned fatigue in their hands.

Mental demand The pupils also described, that the mental demand was higher with the Gestyboard than on a classical hardware keyboard and subjects had to think more. Primarily the searching for the letters was a reason for that. Although the pupils knew the 10-finger-system, it was difficult to get used to the Gestyboard with its division into two parts for the left and right hand. It was often reported, that it was especially hard to use initially, but became easier with practice. In this category there were three times as many negative statements as positive ones.

Benefit of the 10 Finger System Knowledge We got 3 times more positive than negative answers when we asked whether they had any profit from knowing the 10-finger-system.

Further usage and preference The majority of the participants would like to use the Gestyboard in the future. A lot of them would like to train to get better in typing with it. There were only very few people who did not want to use the Gestyboard any further and preferred a touch keyboard, because the Gestyboard was too slow or unfamiliar. The preference for the Gestyboard is visible in the answers as we got almost 150% more positive answers about the familiarization than negative.

Several typical statements from the pupils that show their opinion:

"If I used it more, I could write very quickly."

"It is funny to use and interesting."

"There is a difference to the classic keyboard, but if I typed longer I would get used to it."

"Gestyboard makes more fun. I would prefer it, if I had to choose."

Already during this first evaluation we got the impression that despite several weaknesses the Gestyboard showed high potential for the future with further improvements. Further, we could reduce current disadvantages by adapting it better to users needs.

13.2.8. Conclusion and Discussion of the First Gestyboard Iteration

In this section the first iteration of our development procedure of a novel text input technique, the Gestyboard, has been described. A first prototype was implemented for this evaluation and compared to both, the classical hardware and the virtual touchscreen keyboard, in an extensive evaluation. The results of our qualitative interviews with the participants show that most of the participants liked the Gestyboard concept and would like to continue using it. Although the quantitative results of the Gestyboard were worse than the results for the standard keyboards, it made sense to continue that approach for the following reasons: The participants had no chance to get used to the new gestures. Although the standard layout of the Gestyboard was the QWERTY layout, the participants still needed time to decide which finger has to be moved for which keystroke. Besides the training to improve the performance of the Gestyboard there were additional aspects of the first prototype that needed optimization.

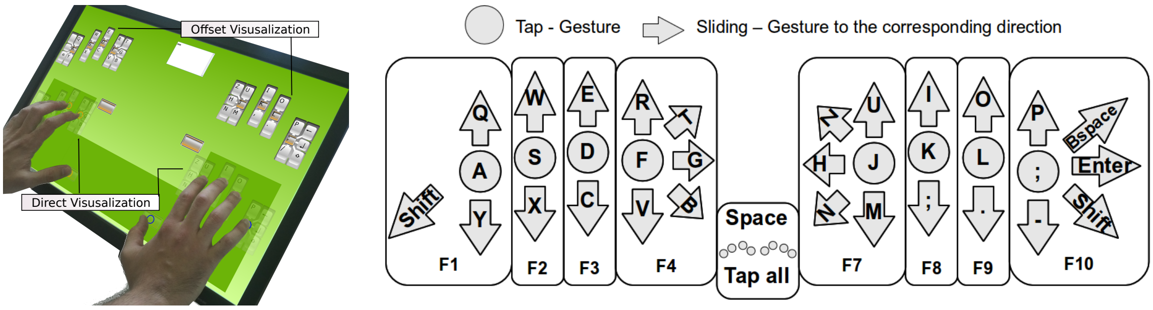
One improvement for the next iteration is to enhance the finger detection mechanism. For this first prototype, the users needed to hit the rings that indicated the last known position of the finger associated to that ring. When the user touched the ring again, the system knew that this should be the finger which had been hovering. By eliminating the need for retouching the virtual rings when fingers were released, the speed of the text entry can also be improved. Additionally, in the next iteration the space key activation gesture was changed to a tap gesture with both hands, instead of tapping with the thumbs. The reason for this is that the thumbs can then be used for other keys which are not included in the concept yet, like the modifier keys of the classical keyboard. As a positive side effect, the users will automatically reset the finger position after each typed word. Finally, it is important to give users the chance to learn the gestures in a longitudinal evaluation.

13.3. Second Iteration: Gestyboard 2.0 - Multi-Session Evaluation

This section is structured as follows: First, the changes made to the concept and its implementation are described in Section 13.3.1. Then, details of the two conducted evaluations of the second iteration are given in Section 13.3.2. Its results are presented in Section 13.3.3. The results of this study has been published by Coskun et al. in APCHI 2012 [25].

13.3.1. Concept 2.0

To conform to the rules of the 10-finger-system, the space key was activated in the previous iteration by tapping with either the left or the right thumb. However, we discovered in our first evaluation that space keystrokes were often executed accidentally. The reason for this was that, due to their larger size, thumbs sometimes were recognized not as one, but two touch points. This was interpreted as a rapid alternation between two corresponding touch points by the software. To overcome this source of errors a new space key activation metaphor was implemented. To input a space, a simultaneous tap with all ten fingers has to be performed. As a positive side effect, each time a space is typed all keys are re-centered according to the positions of the user's fingers. An overview of all gestures of the second version of the Gestyboard is given in Figure 13.16 (b).



(a) (b)

Figure 13.16.: a) The graphical representation of the Gestyboard b) The gesture overview. Fx = Finger x

Another result of the first evaluation was that users were accidentally hitting additional keys as they tried to perform the more challenging diagonal sliding gestures. The reason for this can be seen in Figure 13.17. On the left side the less error-prone approach is shown. A keystroke is only performed when the user re-centers the finger after the visual representation of the letter was reached. That way it is guaranteed that the finger of the user always returns to its appropriate home row letter. However, to support more advanced users the second option was implemented, which is illustrated on the right side of Figure 13.17. The user is still able to re-center the finger, but for some words there might be a gain of speed by following the second alternative. However, this was one of the main sources for errors in our first version of the Gestyboard. But because of the risk to slow down the users by introducing the mentioned restriction, we kept the algorithm like it was (right side of the figure). Instead, the positioning of the visual representations has been optimized by increasing the gap between diagonally positioned letters and those which are adjacent to them. The blind-typing requirement is still fulfilled, because it is difficult to move the finger exactly at the gap-area.

13.3.2. Evaluation

One goal of the evaluation is to determine whether our solutions for the mentioned challenges improved the performance of the Gestyboard. Another goal is to find out how fast people adapt to this input method over the course of multiple sessions. To be able to compare the performance of a new text input concept with the performance of the classic

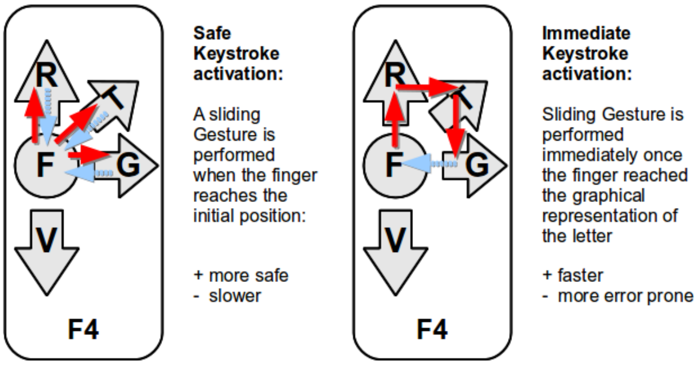


Figure 13.17.: Two alternatives to type the letters “RTG”. Left side: The finger always has to be re-centered to activate a keystroke. Right side: A keystroke is activated when the key is reached by the finger.

keyboard, the users need a lot of training. The first (Proof of Concept) study of the second iteration is described in the subsequent section (13.3.2.1) while the second (long-term) evaluation is described in the following section (13.3.2.2). The purpose of the first study is to show that a considerable speed can be reached with the Gestyboard. The purpose of the second study is to get a first impression of the learning behavior of the Gestyboard.

13.3.2.1. First Evaluation: Proof of Concept

This section describes the first study conducted in the context of the second iteration of the Gestyboard development.

Participants: Two developers having limited experience with the Gestyboard were chosen as the participants for this evaluation. Due to their experience in this field and their prior exposure to the concept they could provide valuable insights on how the technique could be improved and to show that a considerable speed can be reached with the Gestyboard.

Procedure In order to detect some learning effects, each participant had to complete eleven sessions. 1371 characters were typed in session eight and eleven while in each of the remaining sessions, 248 characters were typed. Overall, 4726 characters were typed by the two developers. There was a time interval of one to two days between the sessions. For each session we chose a different subset from TEPS [61].

13.3.2.2. Second Evaluation: Learning Effect

We conducted yet another study involving unbiased participants. The principles of the third formal evaluation are described in the following:

Participants: We evaluated the system with 12 students (11 male, 1 female). All were between the age of 21 and 32 years. None had prior experience with the Gestyboard. Moreover, none of the participants were accustomed to the 10-Finger-System. This decision was based on the fact, that knowing the 10-Finger-System beforehand does not guarantee that users immediately adapt to the Gestyboard concept. They might have a higher learning curve, but because most people did not know the 10-Finger-System by heart anyway, thus we decided that it makes more sense to evaluate the Gestyboard 2.0 with people not knowing the 10-Finger-System.

Procedure: The evaluation was a within-subject design and consisted of three individual sessions, conducted on separate days with a time period of two to three days in between. For each session a different subset of TEPS [61] was used. The participants were asked to enter a text with 1000 characters per session. In the first session our participants typed the phrase set using the Gestyboard, the classical touchscreen keyboard and the classical hardware keyboard (3 stations). There was a ten minute break between each station. During this break the participants filled out the System Usability Scale (SUS) questionnaire [14] for each station. The second and third session had exactly the same setup except that the classical hardware keyboard was excluded, as it was not necessary to collect learning effects on the already well known classical hardware keyboard. After the third session, the participants were interviewed to get qualitative feedback. On average it took the participants 70, 52, and 47 minutes to complete the first, second, and third sessions, respectively. Overall, the participants typed 3,000 characters on each station (9,000 in sum). Although this is not enough to become an advanced user, it gave us a first insight into the learning behavior.

13.3.2.3. Apparatus for both Evaluations

The Gestyboard was shown on a 3M multitouch 22" monitor. We decided to use Tipp10[2] for our evaluation purposes again. We measured the following data with Tipp10: duration to type the sentences, overall number of errors and the number of errors per character in each session.

13.3.3. Results

This section presents the results of both evaluations of the second iteration in our development procedure of the Gestyboard.

13.3.3.1. First Formal Evaluation: Proof of Concept Results

The two participants of the proof of concept study were familiar with the Gestyboard. As expected, the speed increased while the error rate decreased throughout the sessions. Figure 13.18 (a) shows the results of the two participants individually. The measured typing speed was 108 cpm (21.6 wpm) and the lowest error rate was 4% on average.

[2] `http://www.tipp10.com/`

13.3.3.2. Second Formal Evaluation: Learning Effect Results

The results of the second formal evaluation of the second iteration are given in the following:

Learning Effect: Figure 13.18 (b) shows the cpm, the number of errors and the time needed to type 1000 characters for each session. Average speed increased from session one to session three from 42 to 63 which is an increase of 44%. Simultaneously, the error rate decreased from session one to session three from 25 to 14 errors (-48%).

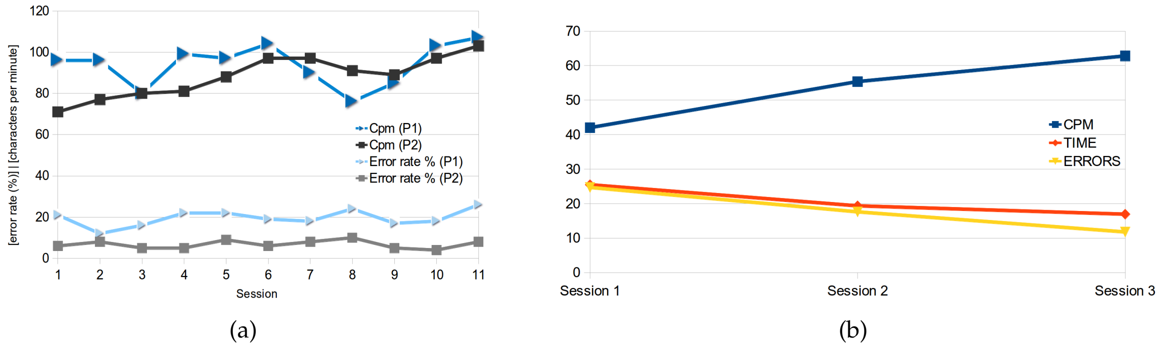


Figure 13.18.: a) Proof of Concept: Performance of two developers during eleven sessions with 276 characters per session. b)Learning effect: Mean values for cpm, errors and the time needed to type 1000 characters for each session for the Gestyboard.

Comparison to the Classic Touchscreen and Hardware Keyboard: Figure 13.19 shows the comparative results of the characters per minute and the absolute error value for each participant. Overall, the Gestyboard still does not perform as good as the classic virtual touchscreen keyboard and the classic hardware keyboard (both in speed and error rate).

Comparison to the first Version of the Gestyboard Figure 13.20 compares the first version of the Gestyboard with the current version. While the average cpm was 31 for the previous version of the Gestyboard, it increased to 45 for the current version, which is an improvement of 45%. The number of errors decreased from 42 to 24, which is a decrease of 42%.

Figure 13.21 shows the mean error rate in percentage for each character of Gestyboard 1.0 and Figure 13.22 shows the same for the current version of the Gestyboard. By comparing these two figures it is noticeable that the number of errors made for each character in version 2.0 of the Gestyboard diminished significantly compared to its predecessor. This is especially the case for the letters ’x’, ’b’, ’t’,’n’, and ’k’.

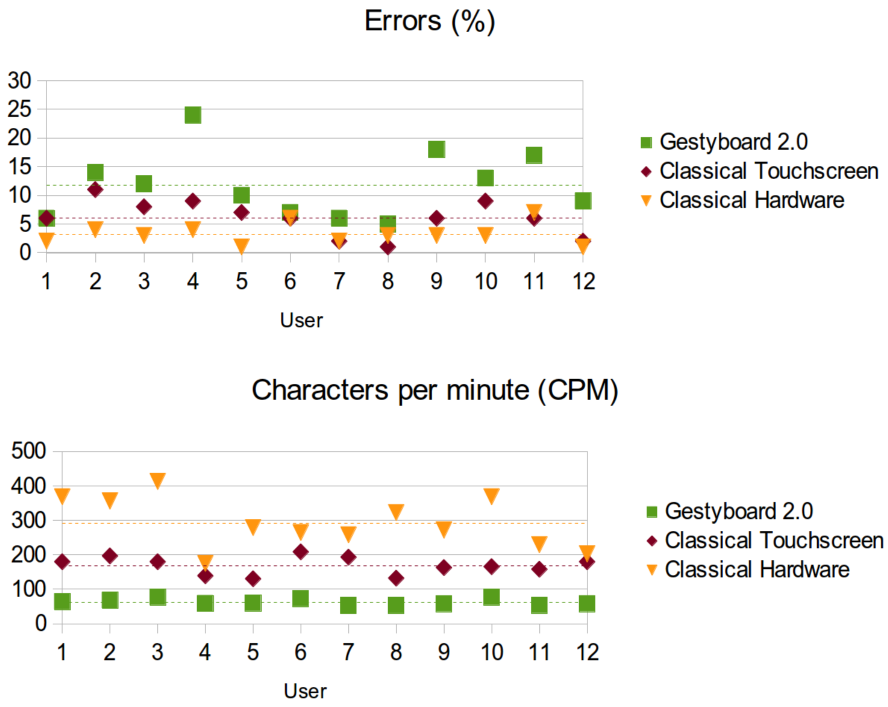


Figure 13.19.: Comparison between the Gestyboard, the classical touchscreen keyboard and the classical hardware keyboard for each user (Last session). The lines represent the mean values.

13.3.4. Conclusion and Discussion of the Second Gestyboard Iteration

In the second formal evaluation of the second iteration of the Gestyboard the participants reached a typing speed of 108 cpm (21.6 wpm). According to Sax et al. [86] this is already a competitive result when compared to the classical virtual touch screen keyboards. Sax et al. argued through the work of Lopez et al. [57] that the mean typing speed of classical touchscreen keyboards is 92.5 cpm (18.5 wpm). However, our evaluation of the classical touchscreen based keyboard resulted in a mean typing speed (in our third session) of 168 cpm (33.6 wpm). This difference to the results given in [86] can be attributed mainly to two factors: First, our participants also improved their typing speed for the classical touchscreen keyboard during our sessions. Second, and even more important, the evaluation in [57] was performed on an Apple iPhone which is a small screen device and therefore not as ergonomic as a keyboard used on a large screen. Consequently, the results of our second formal evaluation prove that the Gestyboard 2.0 can compete with touchscreens on small screen devices but not yet on standard touch-screen keyboards. The participants could not yet reach the performance of the classic touchscreen keyboard (62.8 cpm and 11.75% error rate), but it nevertheless revealed that there is a fast learning behavior. Our participants improved their typing speed by 44% with an average of 63 cpm in just three sessions, and reducing their errors by 48%. The performance can increase dramatically once a user is familiarized with the gestures. A potential benefit from the 10-Finger-System is also evi-

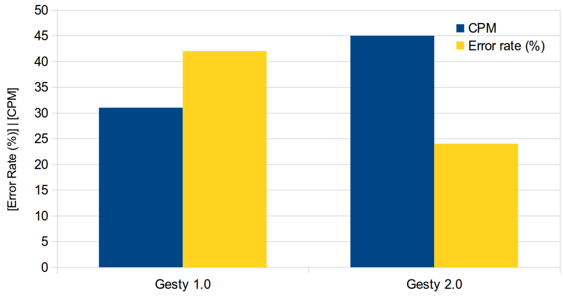


Figure 13.20.: Comparison between Gestyboard 1.0 and Gestyboard 2.0

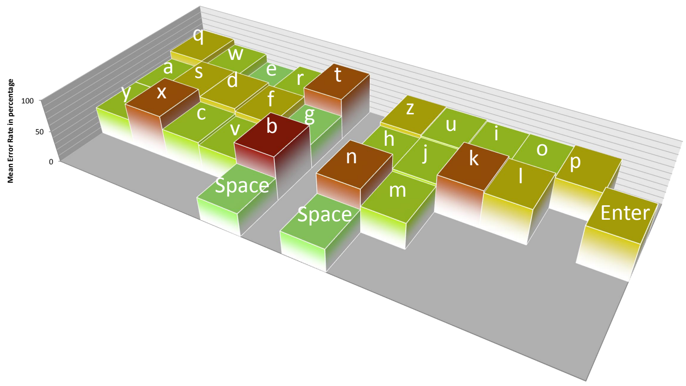


Figure 13.21.: Errors per character for the first version of the Gestyboard (1.0)

dent. Additionally, some of our participants reported that they also became habituated to the finger movements required for a whole sequence of gestures. This was more apparent for words which they typed frequently. This not only improves the typing speed, it also enables the user to blind type without any haptic feedback. Additionally, the participants reported errors made due to moving the wrong finger. A large amount of time was additionally spent for searching the correct letters. However, this was more the case at the beginning of each session, when the participants required time to re-acquaint themselves with the gestures. The space key gesture was another source of error. When users removed their hands for a short period of time and then position their hands back on the screen, the system detected the entry of a space key. In consequence, a space key was often initiated even when the users just wanted to relax their fingers or recenter them. It also occurred when they wanted to abort an action.

Our qualitative interviews with our participants revealed some flaws in the current implementation and the concept itself. To remove one such flaw, we decided to change the gesture for the space key in the third iteration from tapping with all ten fingers to tapping with the five fingers of the left hand. The same gesture performed with the right hand can then be used for the backspace key. This way, people can remove both hands without the

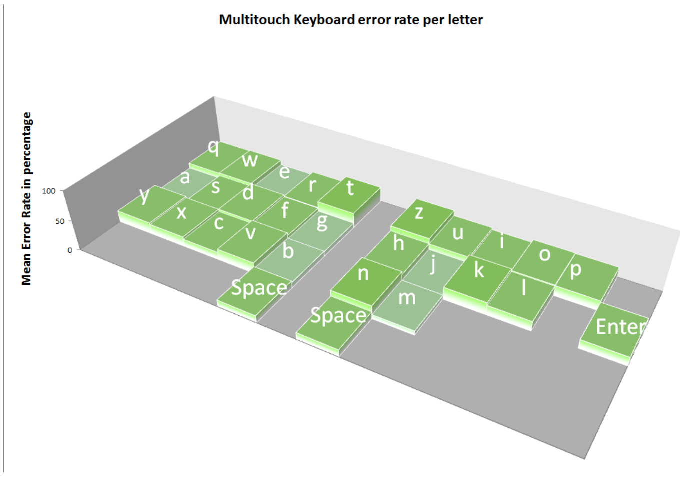


Figure 13.22.: Errors per character for the current version of the Gestyboard (2.0)

risk of performing the space gesture. By eliminating those weak points, the performance of the Gestyboard itself can be further improved for the third iteration. In addition to these improvements, an extended longitudinal evaluation with a detailled analysis of different error rates is part of iteration three. This way, the performance of the Gestyboard can be better compared with the performance of the classical keyboard.

13.4. Third Iteration: Gestyboard 3.0 - Longitudinal Evaluation

This section deals with the third iteration of the Gestyboard development. In Section 13.4.1 the concept of the third iteration and the changes made to the previous version are described. Then, details concerning the longitudinal evaluation are presented in Section 13.4.2 and the results are shown in Section 13.4.3. The results of this study are planned to be submitted to CHI 2014.

13.4.1. Gestyboard 3.0

For the third iteration, the Gestyboard was reimplemented from scratch to ensure that the system's performance would not be a factor limiting the users' input speeds. Additionally, a better finger tracking algorithm was implemented to allow the users to hover more than one finger at the same time. This was not possible in the previous versions. Another problem was the backspace key, one of the most frequently used keys on the classical hardware keyboard. Following the 10-Finger-System, the user had to perform a diagonal sliding gesture to the upper right with the right pinky finger. Yet, the diagonal sliding gesture has also been identified as the most challenging gesture and should therefore not be used for frequently needed keys (like the backspace key). This problem did not appear in our previous evaluations, because a continuous text ticker (Tipp10) had been used with the ticker only continuing when the correct letter was typed. Consequently, the backspace key did not appear in the input stream. For more natural typing task [91] we allowed this time to type errors and we left it up to the user whether to correct mistakes or not. Since we predicted, that the backspace key would be a new challenge for the participants, we decided to change the backspace gesture to a tap with the right hand instead of performing a diagonal sliding gesture. To be consistent, we then also changed the space gesture to a tap with the left hand only. As a positive side effect, users could remove both hands without worrying about accidentally typing a space when touching the screen again. The updated concept is shown in Figure 13.23. The right thumb is intended to be used for the arrow keys of the classical keyboard (also with sliding gestures in the appropriate direction). The left thumb is still free for any other interaction (more modifiers for example).

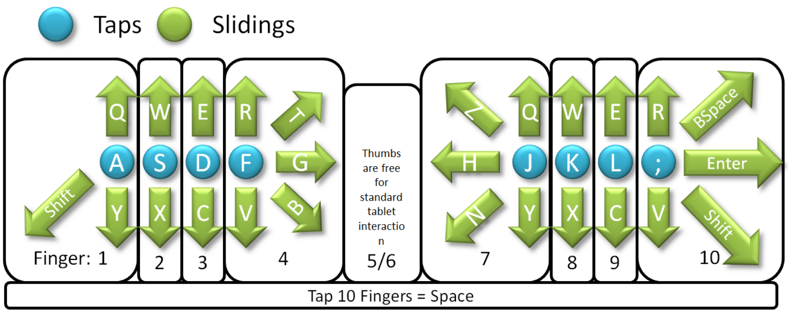


Figure 13.23.: The Gestyboard concept in version 3.0

Figure 13.24 shows the final visualization of the Gestyboard 3.0. Another difference to the previous versions is that the layout of the keys could be changed by sliding the left pinky finger to the left.

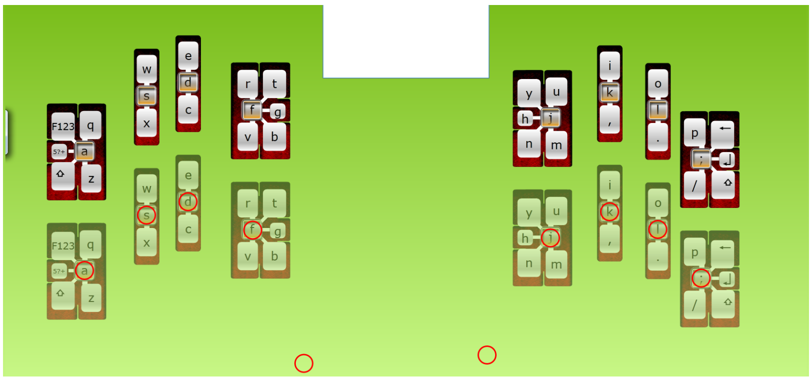


Figure 13.24.: The standard layout of the Gestyboard 3.0.

Since our primary goal was to first achieve the same performance as the classical keyboard, we did not include numerical or special characters in the longitudinal evaluation we focused on the alphanumerical characters instead.

13.4.2. Longitudinal Evaluation & Accelerated Learning

The goal of the third iteration was to find out how users perform in a longitudinal evaluation to explore the expert potential of the Gestyboard.

13.4.2.1. Participants

While nine test users started with the longitudinal evaluation, seven (3 female) of them completed all 20 sessions. None of them used the Gestyboard concept before. Three of them and one of the developers volunteered to continue with an accelerated learning evaluation in addition to those 20 sessions. The developer only took part in the accelerated learning evaluation. All of the participants were students of computer science and were between 19 and 25 years old. All participants who finished the 20th session were rewarded a 50 €compensation. To learn about their speed with the classical hardware keyboard they first typed a short text from TEPS [61]. The results ranged from 36 to 67 WPM.

13.4.2.2. Procedure

In each of the 20 sessions, the users typed 20 minutes with both, the Gestyboard 3.0 and the Windows 7 soft-keyboard. The task was to transcribe a specific text also taken from TEPS. The tool TextTest [105] was used for this purpose. After the test users had finished the tasks with both techniques in the first and the last session, they were asked to fill out additional questionnaires (SUS [14], NASA-TLX [71]). Users were not asked to focus on speed or the error rate. This decision was influenced by Soukoreff et al [91] who argued that requiring test users to focus on one of those variables influences the results, as this way of typing does not necessarily represent the natural typing behavior of the specific user. The minimum time-period for a break between two sessions for a single user was two hours while the maximum time-period was four days. Both text input techniques ran on a 23″ 10-Finger-Input multi-touchscreen from ViewSonic.

13.4.3. Results

In the longitudinal evaluation 140 sessions were conducted with transcription data of 93 hours. Each participant typed 6 hours and 40 minutes for each text input technique. In sum, 24252 sentences consisting of 68500 characters were transcribed.

System Usability Scale (SUS) Figure 13.25 gives the result of the SUS questionnaire. In the first session, the SUS score of the Gestyboard ranged from 12.5 to 77.5 (AVG 43.2) while the SUS score of the Windows keyboard ranged from 35.0 to 87.5 (AVG 66.8). In other words, the Gestyboard reached 64% of the Windows softkeyboard's SUS score. In the last session, the SUS score for the Gestyboard ranged from 12.5 to 87.5 (AVG 53.2) while the SUS score of the Windows keyboard ranged from 57.5 to 85.0 (AVG 70.4). Here, the Gestyboard's average SUS score reached 75% of the Windows SUS score. This means, that the distance between the average SUS results of both systems got smaller.

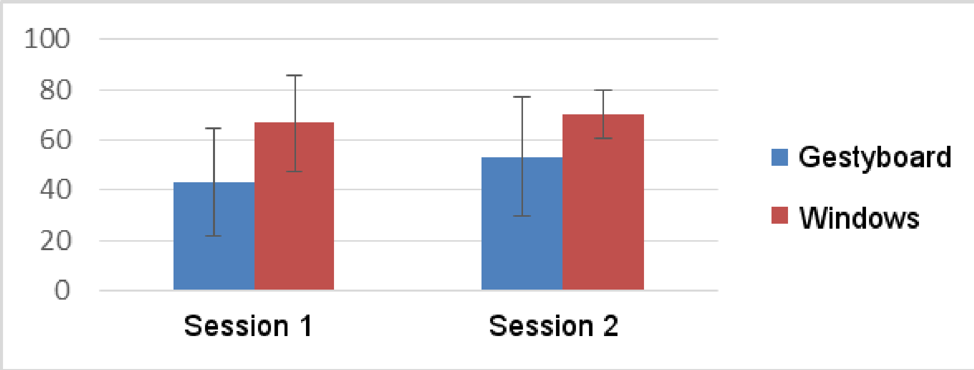


Figure 13.25.: The SUS scores for the Gestyboard and for the Windows 7 softkeyboard

Interviews We conducted short interviews after the first and the last session. After the first session, most of the users mentioned that the usage of the Gestyboard concept was difficult. Just one participant stated that the Gestyboard was easy to use after the first session. Interestingly, the users liked the concept, even at this early stage. Five of the users rated it as being "innovative" and "a good idea". It was mentioned that it was an advantage that the Gestyboard always appeared exactly below the fingers. Another feature which was mentioned as an important advantage was that the Gestyboard enabled users to blind-type with all 10 fingers: "it enables me to do something which always seemed to be impossible". After the 20th session 3 of the test users changed their opinion and wanted to use the Gestyboard more often. The opinions of the other four users did not change: Two did not want to continue using it and two wanted to continue using it. The users rated the comfort of the Gestyboard much better after the last session than they did after the first session. Being able to rest their hands on the screen and the possibility to focus on the text was rated as being very comfortable. A problem mentioned in the interviews was that long finger nails impaired the users when performing the sliding gestures.

Longitudinal Evaluation Figure 13.26 shows the average speed in each session of both the Gestyboard and the Windows 7 soft-keyboard. As expected, the users became faster with both text input concepts from session to session. The increase of the speed attained with the Windows 7 soft-keyboard was similar to that of the Gestyboard. This was unexpected and consequently, in a fourth iteration, the concept of the Gestyboard itself has to be enhanced without losing the feature to blind-type. Possible approaches are discussed in Section 15 (Future Work).

Accelerated Learning Three of the test users and one of the developers volunteered to participate in the subsequent accelerated learning sessions. The developer performed exactly the same longitudinal evaluation as a pre-tester but with a different keyboard layout (Colemak). This layout is better suited to the Gestyboard concept because the most frequent letters are positioned on the home row. For the Gestyboard, this means more taps and less sliding gestures need to be performed. Due to this and because of prior knowledge

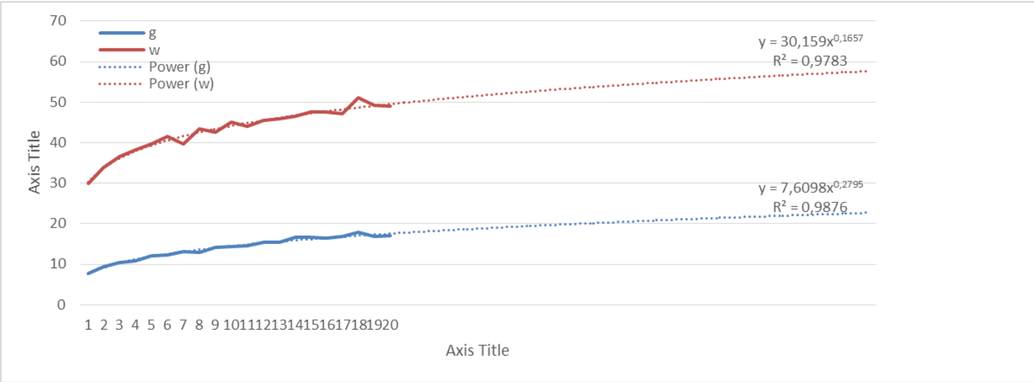


Figure 13.26.: Average and extrapolated speed

of the system, this test user was excluded from the results of the longitudinal evaluation presented in this dissertation.

For this additional study, four additional sessions were conducted in which the users repeated typing the same three sentences. In sum, each of the three sentences was typed 50 times. Figure 13.27 shows the results regarding the speed reached within the acceler-

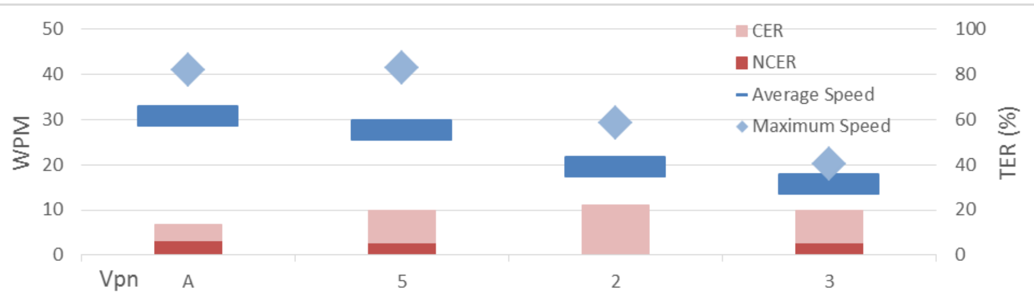


Figure 13.27.: The speed results of the accelerated learning session. ER = Error Rate; CER = Corrected ER, NCER = Not CER, TER = Total ER

ated learning session. Test user A is the user who used the optimized layout while test user 5 is the one who performed best during our longitudinal evaluation. Because just one of the users used the optimized layout, a significant difference caused by the different layout cannot be proven. Nevertheless, this user made the least errors and also reached a comparable speed to the best performer of the longitudinal evaluation. This tendency suggests the following hypothesis: "Using an optimized layout reduces the errors significantly while increasing the typing speed."

13.4.4. Discussion of the Third Gestyboard Iteration and Lessons Learned

Although the goal to reach the speed of the classical hardware keyboard has not yet been reached, our test users already attained the considerable speed of 41 WPM and they were also able to blind-type. After the 20th session, 6 out of 7 test users rated the Gestyboard as being comfortable to use and wanted to continue using it. The most important result for the future development of the Gestyboard (or similar concepts for other purposes) might be the fact that switching between taps and sliding-gestures naturally slows the user down and training this switch to muscle memory takes more time than learning just tapping because of the increased complexity. Additionally, the longer the sliding gesture is the more time it takes to perform it. Shorter sliding gestures, on the other hand, are more error prone because of unintended finger movements. Keeping those challenges in mind, we developed seven main alternative concepts for the fourth iteration of the Gestyboard to overcome those issues. These new approaches are presented in Section 15.

13.5. First Iteration: Gestyboard Backtouch - Mobile Text Input

This section describes the mobile version of the Gestyboard, the Gestyboard Backtouch, which was the initial reason of developing the general Gestyboard concept. It follows the same structure as the stationary version of the Gestyboard: First, the concept and differences to the stationary version are described in Section 13.5.1. Then, details about the evaluation design is given in Section 13.5.2. Afterwards, its results are presented in Section 13.5.3. The results of this iteration has been published in [95].

13.5.1. Gestyboard Backtouch 1.0

Compared to the original Gestyboard, the finger movements of the mobile version are apparently more challenging. This emerges from the modification of the QWERTY layout as shown in Figure 13.28. Our first goal then is to see whether people are able to accommodate to the adapted concept. To be able to answer this question, an Android application has been developed for the ASUS Transformer pad [5].

As stated earlier, the Gestyboard concept enables the user to blind-type on a touchscreen. This concept can therefore be transformed and adapted to blind-typing with the free 8 fingers on the back-side of a tablet device while holding it in both hands. The consequence for the users is having to mentally rotate the QWERTY layout by -90° of the left half side of the keyboard and by 90° of the right half side. Additionally, both sides of the keyboard have to be vertically mirrored. Figure 13.28 illustrates this requirement for the left side of the keyboard or the tablet.

Other than that transformations, the first version of the Gestyboard Backtouch works exactly the same like the Gestyboard 2.0 (stationary version) building upon the results of the Gestyboard 1.0.

13.5.2. Evaluation

We first describe the evaluation procedure and the participants in Section 13.5.2.1 and the evaluation procedure in Section 13.5.2.2.

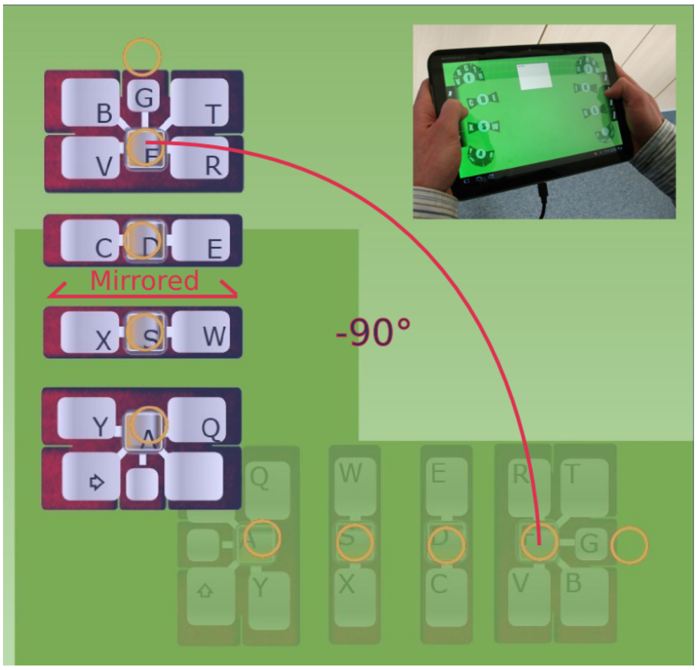


Figure 13.28.: The visual and mental adaptation of the Gestyboard concept and the resulting concept.

13.5.2.1. Participants

It was planned to perform three evaluation sessions with ten computer science students for this initial test. After performing those three sessions with ten students, four among those ten asked us if they may continue with the evaluation to further improve their performance. Consequently, we decided to add another three sessions for those four participants.

13.5.2.2. Procedure

The task of each session was to type 1.000 letters chosen from TEPS [61]. Tipp10 software tool [96] was used to present and analyze the input data as for the first and second iteration of the stationary version. Finally, we asked the participants to fill out the System Usability Scale (SUS) [14] questionnaire and conducted a short interview.

13.5.2.3. Apparatus

For evaluation purposes, the tablet was held such that the front side of the device, i.e. the display, was on the back side and hence was not facing the holder. This way, the touchscreen of the device could be used to track the fingers' movements. The gesture parameters were sent to the original Gestyboard application through a local wireless network and a layer-based software design. This allows to either react on finger gesture parameters

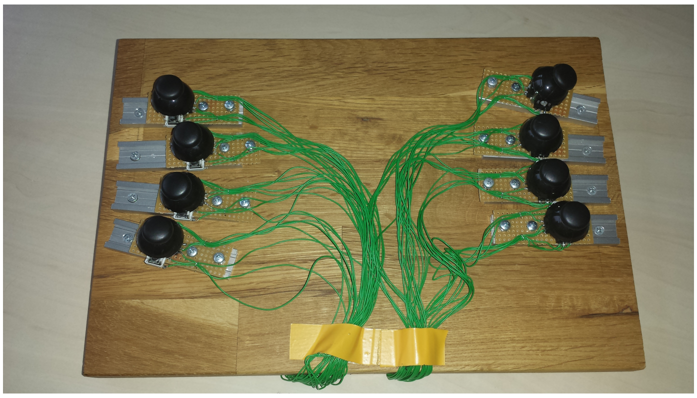


Figure 13.29.: The Gestickboard prototype: the hardware version of the Gestyboard Backtouch

coming from the touchscreen attached to the Gestyboard server or to react to the gesture parameters received through the network and sent from any kind of a client. In the case of the Gestyboard Backtouch, this client was a modern tablet device with the Android OS running on it (The first ASUS transformer published [5]).

Although not being part of this dissertation, it should be mentioned, that other clients already have been tried out and evaluated: Gestairbord ([43]) - A version of the Gestyboard which is based on a finger-tracking algorithm using the Kinect (no touchscreen). This allowed the users to type text in the air. Gestickboard (shown in Figure 13.29) - This is a hardware version of the Gestyboard Backtouch providing haptic feedback to the users. However, this is only mentioned here to show the potential of the Gestyboard concept concerning the ubiquity of the concept.

As mentioned, instead of reacting on the input of a directly attached touchscreen, the mobile version reacts on the data received from the network. This way, the Android tablet becomes a remote controller for the Gestyboard. For the final usage, the Gestyboard logic should be transferred to the tablet to have really mobile flexibility. Additionally, we added the capability to rotate and mirror the visualization of the key groups as described in Section 13.5.1. This way, the users see the visualization of their fingers' movements fitted to the rotation of their hands on an external monitor. This setup is also used during our evaluation and can be seen in Figure 13.30.

This setup is a testing prototype. The future vision is to use a tablet with a touchscreen on the front side and a touch input system (e.g. a multitouch touchpad) on the back side. Additionally, it is planned to provide an option to disable the visualization of the finger movements and the gestures completely for expert users. This way, the interface of any other application does not suffer from the visualization of a virtual keyboard as it is the case for almost all other text input concepts in the market.

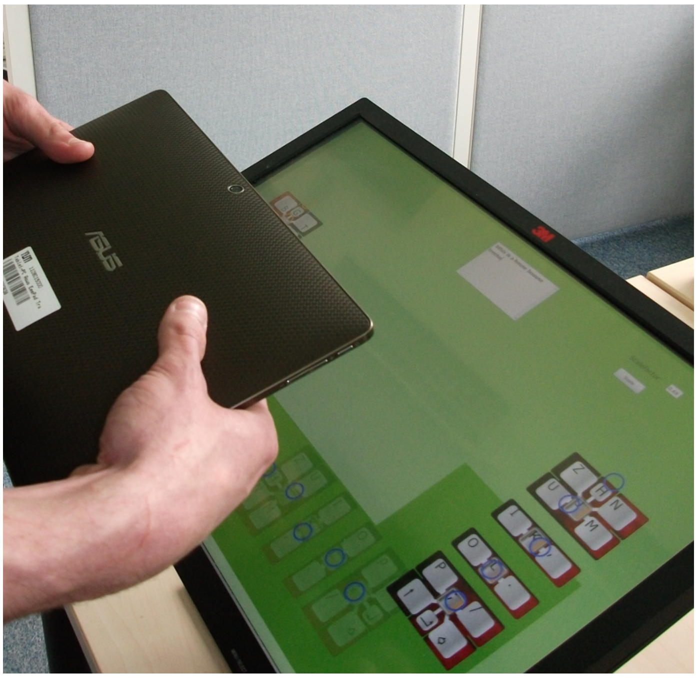


Figure 13.30.: Using an Android tablet as a remote controller for the original Gestyboard Prototype. The tablet is rotated to be able to test the concept.

13.5.3. Results

This section presents the quantitative evaluation results. The results are then discussed and interpreted in Section 14.

Performance We obtained the following quality measures from Tipp10 tool: Typing speed, overall error rate, and the error rate per finger. Figure 13.31 shows the average time needed to type 1000 letters per session in minutes (dark color curve) and the percentage of the error rate (light color curve). The Words per Minute (WPM) parameter represents the average speed. The average speed in the first session among our 10 participants is 5.4 WPM while the error rate is 35.26%. However, the speed is gradually increasing and the error rate is decreasing throughout the sessions. In the last session, the 4 remaining participants reached an increased average speed of 8.6 WPM and a reduced error rate of 17.52%. Thus, despite the fact, that the tablet is occluding the fingers of the user and although there is no haptic feedback at all, our test users were able to blind-type with the 8 Fingers behind the tablet.

Error Rate per Finger Figure 13.32 shows the error rate in percentage per finger. The lowest error rates were reached with the home row keys (tap gesture) and the keys which are directly above or below them (up and down sliding gestures). The highest error rates occurred for the keys placed diagonally to the home row keys (diagonal sliding gestures).

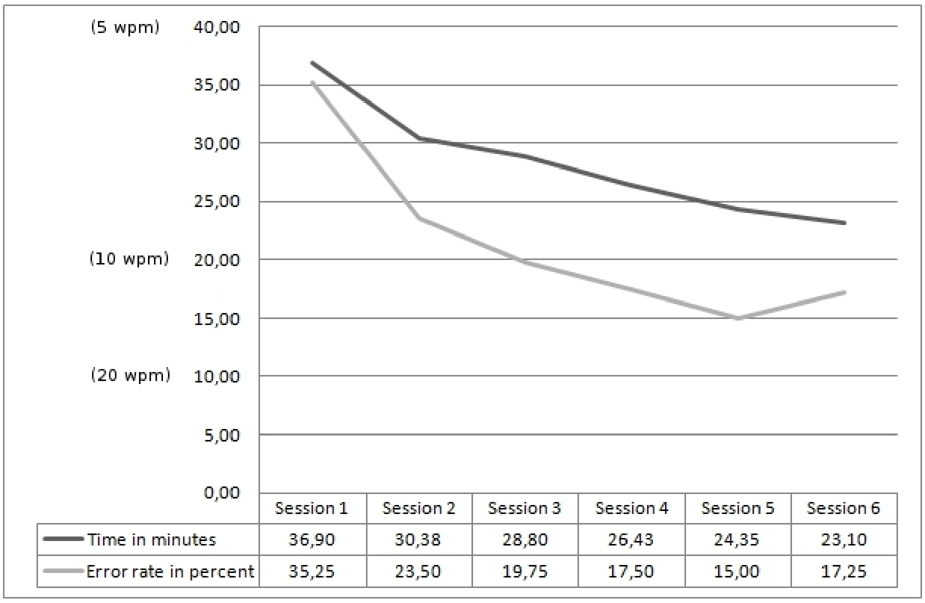


Figure 13.31.: Average time per session (minutes), Error Rate (percent), and the average speed (WPM)

System Usability Scale (SUS) In this section we introduce the SUS values gathered from the SUS questionnaires for each participant. They were filled out after each of the three sessions. Figure 13.33 shows the SUS scores for the first three sessions. We can clearly observe an overall increase in the SUS mean score throughout the sessions. Indeed, the SUS score increased from a mean score of 53.5 in the first session to a mean score of 61.5 in the second one to finally reach a score of 63.3 in the third session. We also notice in the SUS boxplot that the distribution of SUS scores among users is narrowing throughout the sessions and the maximum SUS score exceeds a value of 80 in the third session.

13.5.4. Discussion of the First Gestyboard Backtouch Iteration and Lessons Learned

From the results presented in Section 13.5.3 we observed two main conclusions. First, we noticed that despite this observed increase, SUS score was still quite average, which makes it difficult to affirm for sure the users' feedback concerning the usability. However, we concluded that the more training the users get, the higher the usability score is, which was expected. This could also be interpreted from the narrower distribution in the SUS boxplot in the last two sessions. And second, we noticed a clear increase in the learning curve of the Gestyboard Backtouch throughout the sixth sessions.

However, 6 sessions are not sufficient to compare these results with the classical hardware or touchscreen keyboards due to the large familiarity with the latter ones. MacKenzie et al.[66] used about 10 sessions for novel touch screen keyboards in order to compare them with the well-known classical QWERTY keyboard. In fact, the Gestyboard Backtouch (8.6

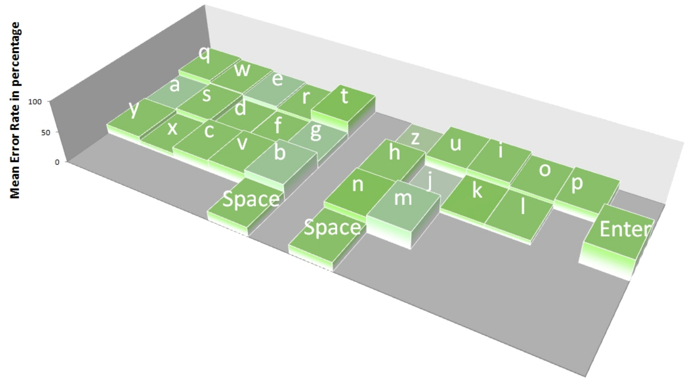


Figure 13.32.: Error Rate per finger in percent.

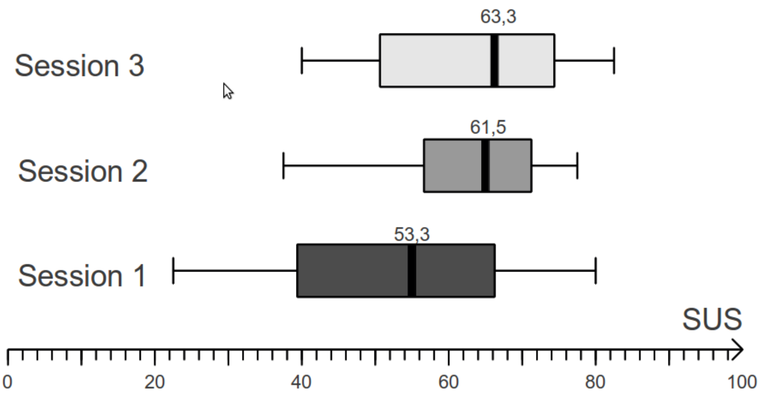


Figure 13.33.: System Usability Scale for each participant.

WPM) did not reach the performance of a text input system using the thumbs for two-handed interaction on a tablet PC (11 WPM) [50]. And this difference in typing speed can be argued by the lack of experience with both the finger based gestures and the 10 finger system. Therefore, with a better training, we expect the performance of the text input system using the thumbs to be surpassed by our solution, and the limitation to the thumbs interaction to be eliminated.

14. Conclusion & Discussion of the Gestyboard Development

In this part of the dissertation, our user-centered iterative development procedure considering two-handed text input for touchscreens has been described. Text input was rated as being an important requirement by our target group (Fire Deparment TUM and Rescue Service Stralsund). Our first approach of splitting a touchscreen keyboard such as that both half can be positioned on the left and the right bezel of the tablet worked quite well. The test users immediately understood the concept and could easily adapt from the standard layout of a keyboard to the split version (see Section 13.1). But on the same time, the users were not very fast with this approach which was due to the restriction to the thumbs.

This lead to the idea of using the eight free fingers on the back of the tablet device for high performance blind-typing - The Gestyboard concept. First, we performed three iterations on the stationary version of the concept designed to be used for large touchscreens such as tabletop devices. In the third iteration the Gestyboard concept was already optimized according to the results of the previous iterations. Thus, a longitudinal evaluation has been performed to study the learning behavior of the concept. The best speed a test user reached was 41 WPM which is an acceptable speed for touchscreen based text input. But all of the users also increased their speed on the standard touchscreen keyboard during the sessions. Consequently, the initial goal of being faster with the Gestyboard could not be reached. Because of the results and the feedback of the users, we believe that as long as the Gestyboard differentiates between tap- and sliding-gestures, this goal cannot be reached without breaking one of the requirement described in Section 11. Nevertheless, the Gestyboard concept enabled our participants to blind-type because of this differentiation between tap- and sliding gestures.

The blind-typing capability also allowed us to develop the Gestyboard Backtouch version described in Section 13.5.1. This is an approach allowing the usage of the 8 free fingers on the back side of the device, although the fingers are occluded by the tablet. Although the users were quite slow and error-prone when using the Gestyboard Backtouch the first time, they were also able to type without seeing their fingers. Proofing this was the goal of the first iteration of the Gestyboard Backtouch. However, to be used during an MCI, the Gestyboard Backtouch version has to be further optimized and a longitudinal evaluation has to be performed. To further optimize the Gestyboard concept we developed different ideas. The second iteration of the Gestyboard Backtouch version is currently on work.

The new approaches for the fourth iteration of the Gestyboard concept are given in Section 15.

15. Future Work - Gestyboard 4.0

Based on our experience, the feedback of the test users, and on the analysis of the longitudinal evaluation of the Gestyboard 3.0, the following new alternatives for the Gestyboard concept have been developed for the fourth iteration:

Gestyboard 4.0a - Finger Tracking Supported by a Depth-Camera The idea here is to enable tracking of the fingers even when no finger is touching the screen by using additional hardware for finger tracking (like the Kinect.) Of course this contradicts with **R1** not to use any additional hardware. This enables us to change the concept in a way that only one finger is touching the screen at a time. The user performs either a tap or a sliding gesture with this finger. Once the finger releases the screen the according key will be typed. The other fingers, not interacting with the screen, are hovering the screen. We believe that the error-rate can be further decreased with this technique, because the user can move his fingers in the air without activating a gesture accidentally.

Gestyboard 4.0b - Remove the Sliding Gestures by adding modifier gestures This idea intends to remove the distinction between sliding and tap gestures to further increase the typing speed of the Gestyboard. All keys are activated with taps only. To differentiate between the home row, the upper row, and the bottom row, different combinations of touching the screen with the thumbs will be used. For example: No thumb is touching the screen makes the home row key active, touching it with the left thumb activates the bottom row and touching it with the right thumb activates the top row. The advantage is that the user always performs the same gesture. The disadvantage is that the user now has to think about which thumb has to touch the screen which adds another level of complexity to the typing mechanism. Another problem with this approach is that the keys positioned diagonally need a special activation mechanism. Nevertheless, a comparison between the current version of the Gestyboard and this modified version is interesting.

Gestyboard 4.0c - Remove the Sliding Gestures by using a language model Another way to remove the sliding gestures is to use a language model. This contradicts with **R2** not to use any dictionary. But this way, there is no need for the user to differentiate between different rows. The user just taps with the finger which should type the letter according to the 10-Finger-System. The system itself then decides which word the user wanted to type based on a language model. We are currently working on this approach. The results are promising. Because this way the Gestyboard works similar to the 1Line keyboard, we called this version Gestyline.

Gestyboard 4.0d - Using Machine Learning Algorithms to improve the Gesture Recognition This alternative does not change the concept but it changes the gesture recognition algorithm. A (large) number of users will type each letter according to the Gestyboard concept. Those finger tracks will be recorded and used as the input for machine learning algorithms. This way, not only one finger will be considered but all fingers are relevant for each gesture. We hope to especially reduce the error rate with this alternative.

Gestybord 4.0e - Finger Movement Analysis Here we want to conduct studies to analyze the finger gestures needed for the Gestyboard but without any relation to text input. For example, we want to find out the difference in efficiency between: 1) just taps, 2) just sliding-gestures, 3) taps and sliding-gestures.

Gestyboard 4.0f - Add Simple Haptic Feedback In this version we want to add simple haptic feedback to the screen in a way it does not hinder the touch recognition while still being tangibly perceivable by the users. This way, they can feel the center. Again, this contradicts with **R1** but its potential impact is nevertheless interesting.

Gestyboard 4.0g - Using Different Layouts Due to the familiarity of the users with the QWERTY layout, we did not use layouts which better fit to the Gestyboard concept. But because of the potential which was shown within the accelerated learning sessions, we also want to try out this version. Especially those layouts which distributes the frequently needed keys on the home row could have a positive affect on the typing performance.

Gestyboard 4.0h - Reducing the Slide-Distances to a Minimum Without changing the current implementation, we want to conduct an evaluation with minimal sliding movements to increase the speed of the sliding gestures. To be able to differentiate between different small finger movements, only the finger with the maximum distance moved will be taken into account, while the other key-groups will be re-centered by the system. This way, the user does not need to pay attention to move a single finger only.

Part IV.

Overall Conclusion and Summary

16. Summary

The research topic of this dissertation was to develop and evaluate mobile user interfaces (UI) designed to be used in the context of MCIs. This is a challenging task for several reasons. One reason is that the target group is heterogeneous, in fact the background and the age of the rescue service employees are different. Another reason is that although some standards exist, each federal state has different rules for the rescue service and different ways of handling an MCI. Additionally, the cultures, the requirements, and the expectations of the single rescue service organizations differ from each other. Last but not least, this task is challenging because the circumstances, the surroundings, and the situation of an MCI can be dangerous, unpredictable, life-threatening, and time-critical. To be able to gather the necessary requirements to solve those challenges, multiple rescue services from different federal states were part of the consortium of the research project SpeedUp. Additionally, to ensure that the expectations of future users of the systems were met, we strictly followed Jakob Nielsens iterative and user-centered development procedure for both Mobile Map Interaction and Mobile Text Input.

The research focus of this dissertation was first developing an intuitive, efficient, and ergonomic user interface for mobile map interaction on heavy ruggedized tablet devices and second designing efficient mobile and stationary text input techniques for touchscreens. Figure 16.1 summarizes the most relevant development steps and studies conducted within the scope of this dissertation. In 2010, we first developed different concepts based on the initial requirement analysis. The outcome of this analysis was to design interaction metaphors on a digital map covering UI elements to select items, scroll and zoom. These UI elements were designed to be used on a ruggedized tablet device while holding it in both hands. The target group feedback and the discussions during the initial performed evaluations on the developed metaphors, introduced in addition to the mobile map interaction a new and important requirement. This requirement is the necessity of a fast and flexible mobile text input concept in an MCI. Therefore, we started developing the Gestyboard concept in 2011 taking into consideration that the text input system with the two thumbs will be limited and less efficient than a text input system allowing the usage of all fingers simultaneously. The ultimate goal of this concept was to allow the user to use the eight fingers on the backside of a tablet device, assuming that tablet devices equipped with touch sensitive surfaces on the backside will be ubiquitously available in the near future. Despite this goal, a stationary version of the Gestyboard was initially developed, because using the fingers on the backside of the device with no haptic feedback is challenging for the user. This way, we could first learn if people are able to adapt and accept the new Gestyboard finger gestures. Another reason is that the same text input system should be used for all instances of devices suggested in SpeedUp (mobile, semi-mobile, and stationary) in order to keep consistency. After the evaluation of the first stationary Gestyboard prototype, we started developing and testing the mobile version (Gestyboard Backtouch

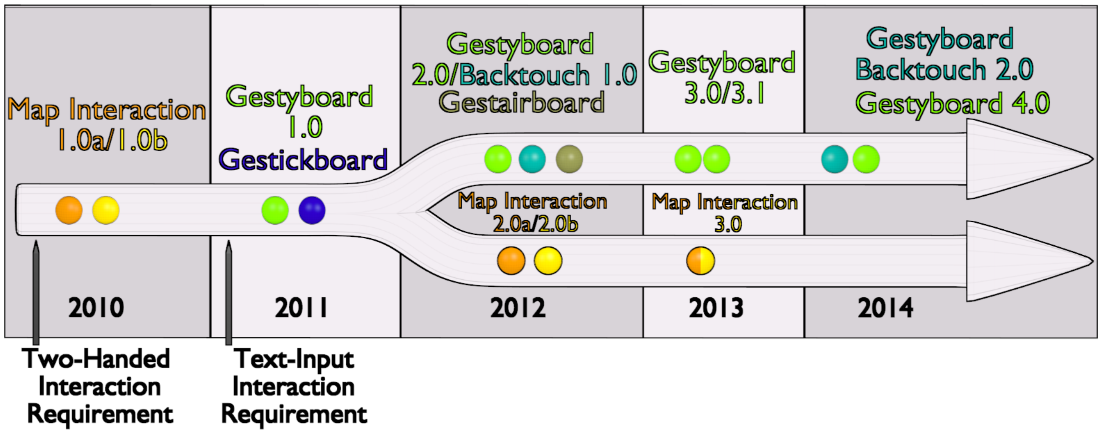


Figure 16.1.: Timeline of the most important development steps and studies conducted in the context of this dissertation.

1.0) along with the second iteration of the stationary version (Gestyboard 2.0). Although the first Gestyboard prototype was quite slow in terms of typing speed and error-prone without training, we were motivated to further develop the concept since we could see a considerable potential of the concept thanks to the acceptance and the enthusiasm of the users.

This dissertation branched into two development paths in 2012: mobile map interaction and mobile text input (see Figure 16.1). Both development paths entered the second iteration phase in 2012. Simultaneously, we started the first iteration of the Backtouch version of the Gestyboard, which allowed the users to use their 8 fingers on the backside of the device while holding it in both hands. Despite the fact that our users were quite slow with this first Backtouch prototype, they also showed a great enthusiasm when using it. In 2013 the third iteration of the Gestyboard 3.0 and the Map Interaction 3.0 emerged. In addition, we started with the second iteration of the Gestyboard Backtouch. For both, Gestyboard 3.0 and Gestyboard Backtouch 2.0, we conducted a longitudinal evaluation in order to find out whether people through training improve their speed to reach the speed of the classical hardware keyboard. While the users improved considerably their speed with the third version of the stationary Gestyboard with fewer errors, they still could not reach the speed of the classical hardware keyboard. When extrapolating the results, we suspect that as long as the Gestyboard differentiates between sliding-gestures and taps, the speed of the classical hardware keyboard will be hard to reach. However, we were extremely encouraged when the goal to enable people to touch-type has been reached. The second iteration of the mobile version (typing at the backside of the device) already started and finishes on March 2014 after the end of this dissertation. But the current results are very promising.

As far as the third iteration of the map interaction is concerned, we tried to develop one UI element capable of all investigated features: selecting, scrolling, and zooming. This composite UI element was compared with a set of the best UI elements corresponding to the three single features. The choice of those UI elements is based on the results of the user studies throughout the different iterations. The initial assumption was that the composite UI element would benefit from not only saving a lot of limited screen spaces but also reducing the distances traveled by the hand, while supporting the same features of the single UI elements. Unexpectedly, our test users from the Arbeiter-Samariter-Bund clearly preferred the individual UI elements. The reason is that the multi-purpose UI elements confused the users. Because the touchscreen of the ruggedized tablet is not reliable and frustrating, the separated elements were easier to use. Consequently, a reevaluation of the third iteration of the map interaction development on a modern tablet device equipped with a reliable touchscreen would be worth investigating. The outcome of the comparative results of this reevaluation is expected to be different with such a tablet device.

In this section we summarized the most relevant results and insights that arose throughout the different development iterations of the mobile map interaction and the mobile and stationary text input. The system development and the studies related to mobile map interaction are described in detail in Part II. The system development and the studies related to mobile and stationary text input are presented in detail in Part III. The new insights and the interesting results are respectively presented in the conclusions of the corresponding part. The overall conclusion of the map interaction part and of the text input part is given in Section 8 and Section 14 respectively. In this chapter (Chapter 16) the most important results for both parts are summarized and discussed. It also suggests alternative ways for the future development.

All publications which have been published in the context of this dissertation (Section A) and all students' contributions to the studies presented in this dissertation, are listed in the appendix (Section B).

17. Discussion

The results of the two development paths of this dissertation will be discussed in the following two paragraphs.

Map Interaction In this dissertation, we developed solutions to interact with a map application running on a heavy ruggedized tablet device while holding this device in both hands. Thus, the thumbs only are free to interact with the map application. As a consequence, we divided the screen space into an interaction area and into a visualization area in which the map was shown. Further, we developed different UI elements for the three main features of any map application: selecting, scrolling, and zooming. The goal was to find the most intuitive and efficient UI elements for these features. We also developed efficient and intuitive UI elements to fulfill these requirements. The requirements, however, got altered over time and depending on the target group priorities. When SpeedUp started, the Fire Department TUM, initial project partner, put the emphasis on the importance of having a ruggedized tablet device and were satisfied with it despite the lack of responsiveness of the touchscreen. Later on, the Arbeiter-Samariter-Bund, a new project partner who joined 18 months later, emphasized however on the importance of a highly responsive touchscreen even if the protection level has to suffer. Indeed, due to the large spreading of modern tablet devices with highly responsive touchscreens, the newer ASB users' expectations on touchscreens is quite high. Unfortunately, during SpeedUp the available ruggedized tablet devices did not fulfill at all this expectations. They also confirmed that in an MCI situation it is equally crucial to take into consideration that each used equipment has to guarantee a degree of reliability towards 100%. Recently, the reconciliation of both latter initially contradicting requirements seems to be possible. In fact, during the last four years the ruggedized tablet devices improved, for example, modern light ruggedized Android based tablet devices are available with better performance and more responsiveness in terms of touchscreens. Second, with ubiquitous and cloud computing, new strategies to meet the requirement of reaching 100% reliability can be followed. Some examples are given in Section 18.

With these recent improvements related to heaviness and responsiveness, it seems that the requirement of holding the device with both hands and hence edge interaction needs to be reevaluated. However, it is worth mentioning that the results of the studies presented in this dissertation, are not only valid for an MCI situation but also very enlightening in other use cases. An example of those use cases is the thumb interaction on mobile phones, this means holding the mobile phone with the same hand used for thumb interaction. This scenario is actually a very popular way where people use their only free hand to interact with their mobile phones. Here, the results we obtained concerning the thumb interaction precision are very useful and enriching. Another example is the usage of indirect control elements on tabletop devices. In fact, the results obtained from the studies investigating the indirect control elements (e.g. the virtual joystick, the selection quad,... etc.) can be reused here to control and interact with elements that are not in the reach area of the user.

Text Input In this dissertation we invented and developed a modern and innovative new text input concept. The concept is based on the 10-Finger system and uses finger-dependent gestures to allow blind-typing with no haptic feedback. This concept can be used for touchscreens, but also with any other technology allowing tracking the finger as used in the work of [43] where the Microsoft-Kinect was used for tracking. This shows that the Gestyboard concept can be used to realize multiple text input versions based on the same concept (Gestyboard, Gestyboard Backtouch, Gestickboard, Gestairboard,...). We also showed that the users can reach an acceptable speed with the stationary version of the Gestyboard and the current ongoing studies with the second iteration of the Gestyboard Backtouch are also very promising. We suspect that as long as the Gestyboard differentiates between sliding-gestures and taps, the speed of the classical hardware keyboard will be hard to achieve. The reason might be that the human brain and the muscular memory of the hands have to switch between those two gestures all the time during typing. Although the user adapts to this mechanism with training, this adds another dimension of complexity to the typing task when compared to the classical hardware keyboard. Some of the ideas for the next iteration of the Gestyboard development cycle described in Section 15 especially try to remove this complexity by disregarding one or more of the Gestyboard requirements. However, this also introduces disadvantages. For example, ignoring requirement 2, which is typing with no dictionary, would introduce the reduction of flexibility of typing and the concern to add in advance new words. Additionally, the usage of the dictionary should be an optional feature of the application and not an inherited property of the text input mechanism itself as it is the case for the classical keyboard hardware. This is why we think it is wise to investigate in parallel the approach of removing the complexity introduced by taps and sliding gestures as well as developing further the Gestyboard development including the latter gestures. Especially, that the Gestyboard fulfilled finally the goal of blind-typing. It is indeed an exclusive feature offered by the Gestyboard concept among the available multi finger touchscreen based text input concepts. This unique feature is even more necessary for the mobile version of the Gestyboard, particularly that the typing happens on the backside of the device. Last but not least, the Gestyboard concept benefits from its inherited ubiquity that would augment text input capabilities on any type of hardware and everywhere with no extra costs related to any additional equipment, adaptability or a complete new design of the existing device.

18. Future Work

Some ideas for the future work and research topics are given in the following two paragraphs for the map interaction and the text input separately.

Map Interaction In Section 17 we discussed the requirement of being able to interact with the tablet device while holding it in both hands. As mentioned earlier, ubiquitous and cloud computing could be a good alternative to realize the goal of reaching 100% reliability. For example, instead of using expensive ruggedized tablet devices, one might think of using affordable tablet devices. Here, if the synchronization of transferred data is continuously done at real time once users log-in the tablet, then it is less important to protect the hardware itself. The device will then be considered as an interface device only. The context of the work is then saved independently of the tablet used. Of course, this is only possible if the network is 100% reliable. This is a tough challenge because the network infrastructure might suffer from either a saturation due to the large amount of calls to reach the emergency services or due to a physical damage. But many research projects focus on improving the network during an MCI. One research topic for example is to build up an ad-hoc network once the rescue service arrives at the incident. With such an ad-hoc network, the reliability as well as the speed of the network is going to be significantly improved. Another research focus is to reduce the network load by developing applications for smartphones which can be used to get information about the incident and the involved people. It also allows the injured people to call for help and to submit their current position. Because this application can provide feedback such as text messages (e.g. “help is coming”) the injured people immediately know that they are taken care of by the official services. This will reduce the attempts the injured people make to reach help and hence the traffic load will be reduced.

Another area of future research is to investigate further the two-handed interaction requirement for the sake of comfortability making benefiting from the Gestyboard Backtouch text input concept. The interaction with the map application could then be done with the 8 fingers behind the tablet. The position of the fingers might then be visualized as shadows on the front screen. One advantage of this concept is that the whole screen space can be used to visualize the map application instead of separating the screen into an interaction and visualization area. Another advantage is that the fingers on the backside of the device can reach a larger distance than the thumbs in the front. Consequently, with a well chosen tablet size, the users are able to cover the whole screen space with the 8 fingers on the backside of the device also allowing backside direct interaction.

Text Input In Section 17 we discussed the Gestyboard approach. One conclusion of the results of our Gestyboard studies is that the performance of the classical hardware keyboard might not be achieved as long as the Gestyboard differentiates between taps and sliding gestures. However, this separation is important in order to keep the functionality of being able to blind-type and to have the flexibility and the power to type anything with no dictionary restriction and no requirement for additional hardware.

For this reason, we suggested different approaches for the fourth iteration of the Gestyboard concept in Section 15. Some of them do not fulfill all the requirements of the Gestyboard. Especially those trying to remove the complexity of taps and sliding gestures. Even though the power and the flexibility of the text input concept will be reduced, when not following all requirements, it is still worth performing further studies and comparisons with those new versions of the Gestyboard development. For example, it would be interesting to compare a dictionary based tap-only version of the Gestyboard with the current version with another keyboard layout. For instance, the Workman layout and the Colemak layout have a good chance to work better than the QWERTY layout with the Gestyboard because they fit better the Gestyboard base concept of taps and sliding gestures. The Workman layout, for example, benefits from the fact that the frequently used letters are better distributed on the keyboard in order to reduce the finger movement distances. The Colemak layout on the other side, benefits from placing the frequently used letters on the home row keys which would reduce the number of sliding with the Gestyboard and increase the number of taps. Taps are faster than sliding gestures. Thus, we expect that these layouts would improve the Gestyboard performance despite the fact that the users are well-trained with the QWERTY layout. Learning to type with another layout takes time and is challenging. A longitudinal evaluation is in fact needed to show the advantage of using those layouts with the Gestyboard concept.

Last but not least, a new innovative hardware could be built based on the Gestyboard concept. For example, a keyboard based on special 2,4,6, and 8-way joystick would provide haptic feedback and nice ways to allow fast and blind-typing by making even less errors. This is because the joystick restricts the movements to the allowed directions only. Another interesting vision is to allow the text input using the Gestyboard concept on touch-sensitive clothes, once they are common and largely available in the market. People can then type while having their hands inside their pockets or wherever they want to type on the touch sensitive surface of their clothes. This shows the potential impact of the Gestyboard concept. The only thing needed is finger tracking allowing blind-typing using 8 or 10 fingers simultaneously on any surface and augmenting text input capabilities everywhere it is needed. The ultimate goal would be to make text input a commodity or an on-demand feature that users can access everywhere and on any surface without thinking of the how it is provided, like electricity.

Appendix

A. Summary of Related Publications

2013	
	T.Coskun, C.Bruns, A.Benzina, M.Huber, P.Maier, **Gestyboard BackTouch 1.0: Two-Handed Backside Blind-Typing on mobile Touch-Sensitive Surfaces** *Ubiquitous 2013 - Tokyo, Japan, International Conference on Mobiquitous Computing - Full Paper*
	T.Coskun, C.Wiesner, E.Artinger, A.Benzina, P.Maier, M.Huber, C.Grill, P.Schmitt, G.Klinker **Gestyboard 2.0: A gesture-based text entry concept for high performance ten-finger touch-typing and blind typing on touchscreens** *Maribor, Slovenia July 01-03, 2013 SouthChi Full Paper*
	T.Velikova,T.Coskun,S. Klingenbeck,J.Roith,E.Artinger,G.Klinker **Animation-based requirements analysis of a map application user interface to support critical MCI situations** *Informatiktage 2013, Fachwissenschaftlciher Informatik-Kongress, Lecture Notes in Informatics (LNI) - Seminars, Vol S-12, pp. 197-200*
	J.Roith, T.Coskun, T.Velikova, S.Klingenbeck, G.Klinker **Gestairboard: A gesture-based touch typing keyboard using the Kinect camera** *Informatiktage 2013, Fachwissenschaftlciher Informatik-Kongress, Lecture Notes in Informatics (LNI) - Seminars, Vol S-12, pp. 127-140*
	S.Klingenbeck, T.Coskun, T.Velikova, J.Roith, G.Klinker **Fog of Triage: Usage of the Fog of War concept for the Triage in a Mass Casualty Incident (MCI)** *Informatiktage 2013, Fachwissenschaftlciher Informatik-Kongress, Lecture Notes in Informatics (LNI) - Seminars, Vol S-12, pp. 115-118*
2012	
	T.Coskun, E.Artinger, G.Klinker **Mobile and stationary map interaction in MCIs** *Summary of a selection of the Map Interaction Studies conducted at TUM in the context of SpeedUp- Invited Talk at Search and Rescue in Oradea - Romania*

	T.Coskun, E.Artinger, G.Klinker **Map Interaction on different devices in the context of MCIs** *Map Interaction for heavy ruggedized tablet PCs presented at Joint Research Center JRC in ISPRA Italy - Invited Talk*
	M. Maehler,E.Artinger, C.Stolcis, F. Wucholt, T.Coskun, Y. Yildirim-Krannig **Developing user centered maps and map symbols in mass casualty incidents - a qualitative interdisciplinary approach** *GMDS 2012/Informatik 2012 IT Unterstuetzung von Einsatzkraeften(IT support in the field of the rescue service)*
	G.Han,T. Coskun, E.Artinger, A.Benzina, G.Klinker **User-centered comparison between classical and edge interaction on a heavy rugged tablet PC used in MCIs** *GI-Jahrestagung, GMDS 2012/Informatik 2012 IT Unterstuetzung von Einsatzkraeften (IT support in the field of the rescue service), pp. 126-140.*
	T.Coskun, C.Grill, A.Benzina, E.Artinger, G.Klinker **How-to interact with a map application on a heavy rugged tablet PC when both hands are needed to hold the device** *GI-Jahrestagung, GMDS 2012/Informatik 2012 IT Unterstuetzung von Einsatzkraeften (IT support in the field of the rescue service), pp. 111-125*
	T. Coskun, E. Artinger, L.Pirritano, D.Korhammer, A. Benzina, C.Grill, A. Dippon, G. Klinker **Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback** *Matsue, Japan August 28-31, 2012 - APCHI 2012 Poster*
	E. Artinger, P. Maier, T. Coskun,S. Nestler, M. Maehler, Y. Yildirim-Krannig, F. Wucholt, F. Echtler, G. Klinker **Creating a common operation picture in realtime with user-centered interfaces for mass casualty incidents** *4th international workshop for "Situation recognition and medical data analysis in Pervasive Health environments"(PervaSense), PervasiveHealth 2012, May 2012, San Diego, CA, USA, pp. 291-296.*
	T. Coskun,A. Benzina, E. Artinger,C. Binder, G. Klinker **User-Centered Development Of UI Elements for Selecting Items on a Digital Map Designed for Heavy Rugged Tablet PCs in Mass Casualty Incidents** *International Health Informatics 2012, pp. 151-160.*
2011	
	T. Coskun, E. Artinger, L.Pirritano, D.Korhammer, A. Benzina, C.Grill, A. Dippon, G. Klinker **Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback** *Technical Report TUM* (bib)

	E. Artinger, T. Coskun, M. Schanzenbach, F. Echtler, S. Nestler, G. Klinker **Exploring Multi-touch Gestures for Map Interaction in Mass Casualty Incidents** *3. Workshop zur IT-Unterstützung von Rettungskräften im Rahmen der GI-Jahrestagung Informatik 2011* (bib)
	S. Nestler, E. Artinger, T. Coskun, Y. Yildirim-Krannig, S. Schumann, M. Maehler, F. Wucholt, S. Strohschneider, G. Klinker **Assessing Qualitative Usability in life-threatening, time-critical and unstable Situations** *GMS Medizinische Informatik, Biometrie und Epidemiologie 2011, Vol. 7(1), ISSN 1860-9171* (bib)
	S. Nestler, E. Artinger, T. Coskun, T. Endres, G. Klinker **RFID based Patient Registration in Mass Casualty Incidents** *GMS Medizinische Informatik, Biometrie und Epidemiologie 2011, Vol. 7(1), ISSN 1860-9171 8* (bib)
2010	
	E. Artinger, M. Schanzenbach, F. Echtler, S. Nestler, T. Coskun, G. Klinker **Beyond Pinch-to-Zoom: Exploring Alternative Multi-touch Gestures for Map Interaction** *Technischer Bericht: TUM-I1006* (bib)
	S. Nestler, T. Coskun, E. Artinger, P. Pichlmaier, G. Klinker **Indirect Tracking of Patients in Mass Casualty Incidents** *2. Workshop zur IT-Unterstützung von Rettungskräften im Rahmen der GI-Jahrestagung Informatik 2010, pp. 156-161.* (bib)
	T. Coskun, S. Nestler, E. Artinger, A. Benzina, G. Klinker **Is it possible to interact with a handheld device while holding it in both hands?** *2. Workshop zur IT-Unterstützung von Rettungskräften im Rahmen der GI-Jahrestagung Informatik 2010, pp. 181-186.* (bib)
	S. Nestler, E. Artinger, T. Coskun, Y. Yildirim-Krannig, S. Schumann, M. Maehler, F. Wucholt, S. Strohschneider, G. Klinker **Assessing Qualitative Usability in life-threatening, time-critical and unstable Situations** *10. Workshop Mobile Informationstechnologien / Mobiles Computing in der Medizin (MoCoMed 2010)* (bib)
	S. Nestler, E. Artinger, T. Coskun, T. Endres, G. Klinker **RFID based Patient Registration in Mass Casualty Incidents** *10. Workshop Mobile Informationstechnologien / Mobiles Computing in der Medizin (MoCoMed 2010)* (bib)

	T. Coskun, E. Artinger, S. Nestler, A. Benzina, G. Klinker **Tablet device interaction with geographical information in time-critical, life- threatening and unstable situations** *10.Workshop Mobile Informationstechnologien in der Medizin (Mocomed 2010)* (bib)

B. Summary of Related Student Work

Title	Type	Student
Repetition of MapInteraction 2.0 on an Android Device	Bachelor	Andreas Schmidt
Gestyboard Backtouch 2.0	Master	Christoph Bruns
Gestyboard 3.0: Long term evaluation	Master	Phillip Schmitt
Improving social networking through AR	Master	Md.Raihanul Islam
Mobile AR Navigation for victims in disasters	Hiwi	Sebastian Klingenbeck
Modelling and Comparison of different 3D metaphors for mobile AR Navigation	Hiwi	Teodora Velikova
Gestyboard Backtouch 1.0:	Bachelor	Christoph Bruns
Fog of Triage	Bachelor	Sebastian Klingenbeck
Requirement analysis based on animation of MCI scenarios	Bachelor	Teodora Velikova
Implementation of a SVG-GUI-Builder in the Context of the SpeedUp-Project	Bachelor	Kang-Hunn Lee
Development of a Single Touch User Interface for the Efficient and Intuitive Completion of Forms in the Area of Emergency Rescue Services	Bachelor	Daniela Korhammer
Implementation and Evaluation of an innovative multitouch-keyboard based on gestures	Hiwi	Lorenzo Piritano
Gestyboard 3.0: Reimplementation and evaluation	Master	Thomas Faltermeier
Gestairboard	Bachelor	Johannes Roith
Comparison of direct and indirect interaction on a ruggedized tablet	Master	Gel Han
Gestickboard Prototype	Hiwi	Natalia Zarawska
Evaluation of Usability	Master	Carmen Rudolph
Gestyboard 2.0	Master	Christian Wiesner
ManvBook	Hiwi	Md.Raihanul Islam
Map Interaction 3.0	Bachelor	Thomas Behrens
Motivation Sports Cave Kinect Emotion Exergotchi	Bachelor	Mathias Gorf
The impact of virtual emotions on real persons	Bachelor	Clara Lange

Increasing motivation for sports with the help of ubiquitous computing	Bachelor	Mathias Gorf
Development and Evaluation of different mouse metaphors for touchscreens	Bachelor	Uwe Trottmann
Map Interaction 2.0: Selection	Bachelor	Clemens Binder
A tiling window manager for multitouch devices	Master	Philipp Comans
.Net to WPF Transfer	Hiwi	Emal Sadran

Bibliography

[1] Thomas Acton, Willie Golden, Sorin Gudea, and Murray Scott. Usability and acceptance in small screen information systems. In *Proceedings of 9th European Collaborative Electronic Commerce Technology and Research Conference*, 2004.

[2] Pär-Anders Albinsson and Shumin Zhai. High precision touch screen interaction. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, 2003.

[3] A.S. Arif and W. Stuerzlinger. Analysis of text entry performance metrics. In *Science and Technology for Humanity (TIC-STH), 2009 IEEE Toronto International Conference*, pages 100–105. IEEE, 2009.

[4] E. Artinger, P. Maier, T. Coskun, S Nestler, M. Maehler, Y. Yildirim-Krannig, F. Wucholt, F. Echtler, and G. Klinker. Creating a common picture in realtime with user-centered interfaces for mass casualty incidents. In *PervaSense 2012*, SanDiego, USA, Mai 2012.

[5] Asus. Asus transformer pad tf300t. http://www.asus.de/Tablet/TransformerPad/. 02.08.2013.

[6] AttrakDiff2. *AttrakDiff2*. `http://attrakdiff.de/`. 02.11.2013.

[7] S. Bachl, M. Tomitsch, C.G. Wimmer, and T. Grechenig. T. challenges for designing the user experience of multi-touch interfaces. In *Proc. Workshop on Engineering Patterns for Multi-Touch Interfaces*, 2010.

[8] Patrick Bader. Entwicklung eines virtuellen Eingabegeräts für Multitouch Screens. Bachelor's thesis, Hochschule der Medien, Stuttgart, Germany, GE. Master's thesis, 08 2008.

[9] O. Bau, I. Poupyrev, A. Israr, and C. Harrison. Teslatouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 283–292. ACM, 2010.

[10] Jörg Baus, Keith Cheverst, and Christian Kray. A survey of map-based mobile guides. *Map-based mobile services-Theories, Methods, and Implementations*, pages 197–216, 2005.

[11] BBC. *Ice Age star map discovered*. http://news.bbc.co.uk/2/hi/871930.stm. 09.08.2000.

[12] M. Benson, K.L. Koenig, and C.H. Schultz. Disaster triage: Start, then save—a new method of dynamic triage for victims of a catastrophic earthquake. *Prehospital and disaster medicine*, 11(02):117–124, 1996.

[13] Mark Benson, Kristi L. Koenig, and Carl H. Schulz. Disaster triage: Start, then save – a new method. *Prehospital and disaster medicine: the offical journal of the National*, 1996.

[14] John Brooke. Sus: a 'quick and dirty' usability scale. *Usability evaluation in industry*, 1996.

[15] Steven J Castellucci and I Scott MacKenzie. Graffiti vs. unistrokes: an empirical comparison. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 305–308. ACM, 2008.

[16] T.T. Chairman-Hewett. *ACM SIGCHI curricula for human-computer interaction*. ACM, 1992.

[17] Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. An empirical study of typing rates on mini-qwerty keyboards. In *CHI'05 extended abstracts on Human factors in computing systems*, pages 1288–1291. ACM, 2005.

[18] Philip Cohen, David McGee, and Josh Clow. The efficiency of multimodal interaction for a map-based task. In *Proceedings of the sixth conference on Applied natural language processing*, pages 331–338. Association for Computational Linguistics, 2000.

[19] T. Coskun, E. Artinger, S. Nestler, A. Benzina, and G. Klinker. Tablet device interaction with geographical information in time-critical, life-threatening and unstable situations. In *GMDS 2010*, Mannheim, Germany, September 2010.

[20] T. Coskun, A. Benzina, E. Artinger, C. Binder, and G. Klinker. User-centered development of ui elements for selecting items on a digital map designed for heavy rugged tablet pcs in mass casualty incidents. In *Proceedings of ACM SIGHIT International Health Informatics Symposium (IHI 2012)*. ACM Press, 2012.

[21] T. Coskun, S. Nestler, E. Artinger, A. Benzina, and G. Klinker. Is it possible to interact with a handheld device while holding it in both hands? In *10. Workshop Mobile Informationstechnologien / Mobiles Computing in der Medizin (MoCoMed 2010)*, Mannheim, Germany, September 2010.

[22] Tayfur Coskun and Eva Artinger. Map interaction on heavy ruggedized tablet pcs. Joint Research Center, 2012. Invited Talk.

[23] Tayfur Coskun and Eva Artinger. Map interaction on heavy ruggedized tablet pcs. Search and Rescue - Invited Talk, 2012. Invited Talk.

[24] Tayfur Coskun, Eva Artinger, Lorenzo Pirritano, Daniela Korhammer, Amal Benzina, Claudia Grill, Andreas Dippon, and Gudrun Klinker. Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens w ith no need for tactile. techreport, 2011.

[25] Tayfur Coskun, Eva Artinger, Lorenzo Pirritano, Daniela Korhammer, Amal Benzina, Claudia Grill, and Gudrun Klinker. Gestyboard: A 10-finger-system and gesture based text input system for multi-touchscreens with no need for tactile feedback (poster). SIGCHI - APCHI 2012, 2012.

[26] Tayfur Coskun, Claudia Grill, Amal Benzina, Eva Artinger, and Gudrun Klinker. How-to interact with a map application on a heavy rugged tablet pc when both hands are needed to hold the device. In *GI Jahrestagung: GMDS 2012 - Informatik 2012 - IT Unterstuetzung von Einsatzkraeften 2012*, pages 111–125. GI, 2012.

[27] Gennaro Costagliola, Vittorio Fuccella, and Michele Di Capua. Text entry with keyscretch. In *Proceedings of the 16th international conference on Intelligent user interfaces*, pages 277–286. ACM, 2011.

[28] M Egenhofer and Werner Kuhn. Interacting with geographic information systems. *Geographical Information Systems: Principles, Techniques, Applications, and Management. D. Rhind, Wiley: New York*, pages 401–412, 1999.

[29] Latham Sholes et al. Improvement in type writing machines, 1878. US Patent App. US207559.

[30] Irving Finkel. A join to the map of the world: A notable discovery'. *British Museum Magazine*, 23:26–7, 1995.

[31] Fitaly. *Fitaly*. `www.fitaly.com/fitaly/fitaly.htm`. 02.04.2012.

[32] D. Goldberg and C. Richardson. Touch-typing with a stylus. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 80–87. ACM, 1993.

[33] David Goldberg. Unistrokes for computerized interpretation of handwriting, January 21 1997. US Patent 5,596,656.

[34] Marc Hassenzahl, Andreas Beu, and Michael Burmester. Engineering joy. *IEEE Software*, 2001.

[35] Marc Hassenzahl, Michael Burmester, and Franz Koller. Attrakdiff: Ein fragebogen zur messung wahrgenommener hedonischer und pragmatischer qualität. In *Mensch & Computer 2003: Interaktion in Bewegung*, 2003.

[36] Henry-Davis. *Babylonian clay tablet world map*. http://www.henry-davis.com/MAPS/AncientWebPages/103.html.

[37] Henry-Davis. *The earliest known map*. http://www.henry-davis.com/MAPS/Ancient

[38] E. Hoggan, S.A. Brewster, and J. Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1573–1582. ACM, 2008.

[39] Poika Isokoski. *Manual text input: experiments, models, and systems*. PhD thesis, University of Tampere, Department of Computer and Information Sciences, 2004.

[40] Poika Isokoski and Roope Raisamo. *Evaluation of a Multi-device Extension of Quikwriting*. University of Tampere, Department of Computer and Information Sciences, 2003.

[41] Poika Isokoski and Roope Raisamo. Quikwriting as a multi-device text entry method. In *Proceedings of the third Nordic conference on Human-computer interaction*, pages 105–108. ACM, 2004.

[42] Christina L James and Kelly M Reischel. Text input for mobile devices: comparing model prediction to actual performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 365–371. ACM, 2001.

[43] Johannes Roith, Tayfur Coskun, Teodora Velikova, Sebastian Klingenbeck, and Gudrun Klinker. Gestairboard: A gesture-based touch typing keyboard using the kinect camera. In Gesellschaft für Informatik, editor, *Informatiktage 2013 Fachwissenschaftlicher Informatik-Kongress*, volume S-12 of *Lecture Notes in Informatics (LNI) - Seminars*, pages 137–140, Bonn, 2013. Köllen Druch+Verlag GmbH.

[44] E.A. Johnson. Johnson touch displays, December 2 1969. US Patent 3,482,241.

[45] A. Kano, J.C. Read, and A. Dix. Children's phrase set for text input method evaluations. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, pages 449–452. ACM, 2006.

[46] Akiyo Kano, Janet C Read, Alan Dix, and I Scott MacKenzie. Expect: an expanded error categorisation method for text input. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 1*, pages 147–156. British Computer Society, 2007.

[47] Karl G. Kanz, Mike V. Kay, Wolf Mutschler, and Wolfgang Schäuble. mstart-algorithmus für sichtung, behandlung und transport bei einem massenanfall von verletzten. *Notfall & Rettungsmedizin*, 2006.

[48] M. Kölsch and M. Turk. Keyboards without keyboards: A survey of virtual keyboards. In *Workshop on Sensing and Input for Media-centric Systems, Santa Barbara, CA*. Citeseer, 2002.

[49] Lily Kong. Geography and religion: trends and prospects. *Progress in Human Geography*, 14(3):355–371, 1990.

[50] Daniela Korhammer. Development of a Single Touch User Interface for the Efficient and Intuitive Completion of Forms in the Area of Emergency Rescue Services, Munich, Germany, GE. Master's thesis, Technische Universität München, Munich, Germany, GE, 10 2010.

[51] Per Ola Kristensson. Five challenges for intelligent text entry methods. *AI Magazine*, 30(4):85, 2009.

[52] Per Ola Kristensson and Keith Vertanen. Performance comparisons of phrase sets and presentation styles for text entry evaluations. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pages 29–32. ACM, 2012.

[53] Thomas K. Landauer and Prasad V. Prabhu. *Handbook of Human-Computer Interaction*. Elsevier Science Inc., 1998.

[54] Seungyon Lee and Shumin Zhai. The performance of touch screen soft buttons. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, 2009.

[55] James R. Lewis. Literature review of touch-screen research from 1980 to 1992. Technical report, IBM, 1993.

[56] Frank Chun Yat Li, Richard T Guy, Koji Yatani, and Khai N Truong. The 1line keyboard: a qwerty layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 461–470. ACM, 2011.

[57] M.H. Lopez, S. Castelluci, and I.S. MacKenzie. Text entry with the apple iphone and the nintendo wii. 2009.

[58] J. Luk, J. Pasquero, S. Little, K. MacLean, V. Levesque, and V. Hayward. A role for haptics in mobile interaction: initial design using a handheld tactile display prototype. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 171–180. ACM, 2006.

[59] I Scott MacKenzie, Javier Chen, and Aleks Oniszczak. Unipad: single stroke text entry with language-based acceleration. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles*, pages 78–85. ACM, 2006.

[60] I Scott MacKenzie and R William Soukoreff. Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction*, 17(2-3):147–198, 2002.

[61] I. Scott MacKenzie and R. William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 extended abstracts on Human factors in computing systems*, CHI EA '03, pages 754–755, New York, NY, USA, 2003. ACM.

[62] I Scott MacKenzie and K Tanaka-Ishii. Evaluation of text entry techniques. *Text entry systems: Mobility, accessibility, universality*, pages 75–101, 2007.

[63] I Scott MacKenzie and Kumiko Tanaka-Ishii. *Text entry systems: Mobility, accessibility, universality*. Morgan Kaufmann, 2010.

[64] I Scott MacKenzie and Shawn X Zhang. The immediate usability of graffiti. In *Graphics Interface*, volume 97, pages 129–137, 1997.

[65] I Scott MacKenzie and Shawn X Zhang. An empirical investigation of the novice experience with soft keyboards. *Behaviour & Information Technology*, 20(6):411–418, 2001.

[66] I.S. MacKenzie and S.X. Zhang. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*, pages 25–31. ACM, 1999.

[67] MS Mayzner and Margaret Elizabeth Tresselt. Tables of single-letter and digram frequency counts for various word-length and letter-position combinations. *Psychonomic Monograph Supplements*, 1965.

[68] Stephanie Meece. A bird's eye view–of a leopard's spots: The çatalhöyük'map'and the development of cartographic representation in prehistory. *Anatolian studies*, pages 1–16, 2006.

[69] Harald A Mieg, Beat Brunner, Harald A Mieg, and Harald A Mieg. *Experteninterviews: Eine Einführung und Anleitung*. ETH Zürich, Professur für Mensch-Umwelt-Beziehungen, 2001.

[70] Timothi J. Mosby and Christian N. Wiswell. Virtual keyboard based activation and dismissal. US-Patent 2009/0237361, sep 2009.

[71] NASA-TLX. *NASA-TLX*. `http://humansystems.arc.nasa.gov/groups/tlx/index.html`. 02.11.2013.

[72] NASA-TLX. *NASA-TLX-questions*. `http://humansystems.arc.nasa.gov/groups/tlx/downloads/TLXScale.pdf`. 02.11.2013.

[73] S. Nestler, E. Artinger, T. Coskun, Y. Yildirim, S. Schumann, M. Maehler, F. Wucholt, S. Strohschneider, and G. Klinker. Assessing qualitative usability in life-threatening, time-critical and unstable situations. *GMS Med Inform Biom Epidemiol 2011*, 7(1), 2011.

[74] Simon Nestler and Gudrun Klinker. Using mobile hand-held computers in disasters. In *UbiComp Workshop on Interaction with Ubiquitous Wellness and Healthcare Applications (UbiWell)*, 2007.

[75] J. Nielsen. *Usability engineering*. Morgan Kaufmann, 1993.

[76] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pages 249–256. ACM, 1990.

[77] Jakob Nielsen. Using discount usability engineering to penetrate the intimidation barrier. *Guerrilla HCI*, 1994.

[78] Dan Odell and Vasudha Chandrasekaran. Enabling comfortable thumb interaction in tablet computers: a windows 8 case study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 56, pages 1907–1911. SAGE Publications, 2012.

[79] Tim Paek and Bo-June Paul Hsu. Sampling representative phrase sets for text entry experiments: a procedure and public resource. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2477–2480. ACM, 2011.

[80] Ken Perlin. Quikwriting: continuous stylus-based text entry. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 215–216. ACM, 1998.

[81] J. Preece, T. Carey, Y. Rogers, S. Holland, H. Sharp, and D. Benyon. *Human-Computer Interaction*. Ics Series. Addison-Wesley Publishing Company, 1994.

[82] Andrew Ramsay, Marilyn McGee-Lennon, Graham A Wilson, Steven J Gray, Philip Gray, and François De Turenne. Tilt and go: exploring multimodal mobile maps in the field. *Journal on Multimodal User Interfaces*, 3(3):167–177, 2010.

[83] Michael A Rappenglück. Ice age people find their ways by the stars: A rock picture in the cueva di el castillo (spain) may represent the circumpolar constellation of the northern crown (crb). 2000.

[84] Jun Rekimoto. Tilting operations for small screen interfaces. In *Proceedings of the 9th annual ACM symposium on User interface software and technology*, 1996.

[85] C. Sax, H. Lau, and E. Lawrence. Liquid keyboard., August 2011.

[86] C. Sax, H. Lau, and E. Lawrence. Liquidkeyboard: An ergonomic, adaptive qwerty keyboard for touchscreens and surfaces. In *ICDS 2011, The Fifth International Conference on Digital Society*, pages 117–122, 2011.

[87] Andrew Sears and Ben Shneiderman. High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, 1991.

[88] Textware Solutions. The fitaly one-finger keyboard. *Retrieved April*, 4:2006, 1998.

[89] R William Soukoreff and I Scott MacKenzie. Input-based language modelling in the design of high performance text input techniques. In *Graphics Interface*, volume 2003, pages 89–96, 2003.

[90] R William Soukoreff and I Scott MacKenzie. Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 113–120. ACM, 2003.

[91] Robert William Soukoreff. *Quantifying Text Entry Performance*. PhD thesis, York University Toronto, Ontario, Canada, 2010.

[92] Robert Spence. *Information Visualization - Design for Interaction*. Pearson, 2007.

[93] Swiftkey. *Swiftkey*. http://www.swiftkey.net/en/. 11.06.2013.

[94] Swype. *Swype*. `www.swypeinc.com`. 02.04.2012.

[95] Tayfur Coskun, Christoph Bruns, Amal Benzina, Patrick Maier, Manuel Huber, and Marcus TÃ¶nnis andGudrun Klinker. Gestyboard backtouch 1.0: Two-handed backside bild-typing on mobile touch-sensitive surfaces. Mobiquitous 2013 - Tokyo, Tokyo, 2013.

[96] Tipp 10. *Tipp 10*. http://www.tipp10.com/de/. 02.04.2013.

[97] T. Tullis and B. Albert. Measuring the user experience. *Collecting, Analyzing, and Presenting Usability Metrics*, 2008.

[98] Dan Venolia and Forrest Neiberg. T-cube: a fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–270. ACM, 1994.

[99] Keith Vertanen and Per Ola Kristensson. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 295–298. ACM, 2011.

[100] Julie Wagner, Stéphane Huot, and Wendy Mackay. Bitouch and bipad: designing bimanual interaction for hand-held tablets. In *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 2317–2326. ACM, 2012.

[101] D.J. Ward, A.F. Blackwell, and D.J.C. MacKay. Dasher—a data entry interface using continuous gestures and language models. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 129–137. ACM, 2000.

[102] Daniel Wigdor and Ravin Balakrishnan. Tilttext: using tilt for text input to mobile phones. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 81–90. ACM, 2003.

[103] Wikipedia. *Anaximander World Map*. http://en.wikipedia.org/wiki/Anaximander.

[104] Wikipedia. *The History of Cartography*. http://en.wikipedia.org/wiki/History_of_cartography.

[105] Jacob O Wobbrock and Brad A Myers. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 13(4):458–489, 2006.

[106] J.O. Wobbrock, B.A. Myers, and J.A. Kembel. Edgewrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 61–70. ACM, 2003.

[107] www.guinnessworldrecords.com. *Ice Age star map discovered*. http://www.guinnessworldrecords.com/records-10000/first-clay-tablet-map/.