

# Metoda unosa teksta za uređaje sa zaslonom osjetljivim na dodir zasnovana na mišićnoj memoriji

---

Graf, Karlo

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:190:112260>

Rights / Prava: [Attribution 4.0 International](#)/[Imenovanje 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2024-12-25**



Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Engineering](#)



SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**Metoda unosa teksta za uređaje sa  
zaslonom osjetljivim na dodir zasnovana na  
mišićnoj memoriji**

Rijeka, rujan 2024.

Karlo Graf  
0069085452

SVEUČILIŠTE U RIJECI  
**TEHNIČKI FAKULTET**  
Sveučilišni diplomski studij računarstva

Diplomski rad

**Metoda unosa teksta za uređaje sa  
zaslonom osjetljivim na dodir zasnovana na  
mišićnoj memoriji**

Mentor: izv. prof. dr. sc. Sandi Ljubić

Rijeka, rujan 2024.

Karlo Graf  
0069085452

Umjesto ove stranice umetnuti zadatak  
za završni ili diplomski rad

## Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2024.

-----  
Karlo Graf

# Zahvala

Zahvaljujem mentoru izv. prof. dr. sc. Sandiju Ljubiću na podršci tijekom pisanja ovoga rada i korisnim raspravama i savjetima, te na prijedlogu odlične teme.

Također, zahvaljujem se svim profesorima i osoblju fakulteta na prenesenom znanju i podršci tijekom studija. Želio bih zahvaliti kolegama i kolegicama na zajedničkom radu, razmjeni ideja i međusobnoj podršci.

Na kraju, posebno hvala svojoj obitelji i prijateljima koji su mi pružali neprestanu podršku i motivaciju, te mi pomogli prebroditi sve izazove.

# Sadržaj

<b>Popis slika</b>	<b>viii</b>
<b>1 Uvod</b>	<b>1</b>
<b>2 Srodni radovi</b>	<b>5</b>
2.1 Nevidljiva korisnička sučelja . . . . .	5
2.2 Tipkovnice sa smanjenom veličinom . . . . .	6
2.3 Dekodiranje tipkovnice . . . . .	7
2.4 Nevidljiva tipkovnica . . . . .	8
<b>3 Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji</b>	<b>9</b>
3.1 Implementacija tipkovničkog servisa . . . . .	9
3.2 Algoritmi za predviđanje unosa teksta . . . . .	12
3.2.1 Referentni algoritam . . . . .	12
3.2.2 Physical distance algoritam . . . . .	14
3.2.3 String distance algoritam . . . . .	16
<b>4 Empirijsko istraživanje učinkovitosti (HCI eksperiment)</b>	<b>18</b>
4.1 Ispitna aplikacija . . . . .	18
4.2 Procedura eksperimenta i sudionici . . . . .	23
<b>5 Rezultati i diskusija</b>	<b>26</b>
<b>6 Zaključak</b>	<b>37</b>

*Sadržaj*

<b>Bibliografija</b>	<b>39</b>
<b>Pojmovnik</b>	<b>42</b>
<b>Sažetak</b>	<b>43</b>
<b>A Izvorni kod</b>	<b>45</b>



## Popis slika

3.1	<i>Invisible keys</i> dizajn . . . . .	11
3.2	<i>Blank keys</i> dizajn . . . . .	11
4.1	Okruženje za testiranje . . . . .	20
4.2	Alatna traka . . . . .	21
4.3	<i>Session settings</i> aktivnost . . . . .	22
4.4	Prosječan dnevni broj zadataka pisanja sudionika, na mobilnom uređaju . . . . .	24
5.1	TER metrika za sve kombinacije varijabli <i>raspored</i> ( <i>Invis - Invisible keys</i> , <i>Blank - Blank keys</i> ) i <i>algoritam</i> ( <i>R - Referentni</i> , <i>PD - Physical distance</i> , <i>SD - String distance</i> ) . . . . .	27
5.2	Promjena TER metrike za referentni algoritam, u odnosu na broj upisanih fraza . . . . .	28
5.3	Rezultati Mauchly-evog testa sferičnosti unutar SPSS okruženja . . . . .	28
5.4	Rezultati Two-way Anova Repeated Measures testa za TER metriku, unutar SPSS okruženja . . . . .	29
5.5	Rezultati međusobne usporedbe s Bonferroni korekcijom varijable <i>algoritam</i> (1 - Referentni, 2 - <i>Physical distance</i> , 3 - <i>String Distance</i> ) po TER metrici . . . . .	30
5.6	WPM metrika za oba rasporeda tipkovnice . . . . .	32
5.7	Promjena WPM metrike u odnosu na broj upisanih fraza . . . . .	32
5.8	Rezultat Paired Samples t-testa za varijablu <i>raspored</i> . . . . .	33
5.9	Rezultati vrednovanja atributa upotrebljivosti . . . . .	34
5.10	Rezultati Wilcoxon signed-rank testa za attribute upotrebljivosti . . . . .	34
5.11	Rezultati NASA TLX upitnika . . . . .	35

*Popis slika*

5.12	Rezultati Wilcoxon signed-rank testa za NASA TLX faktore . . . .	36
------	--	----

# Poglavlje 1

## Uvod

U suvremenom svijetu mobilni uređaji postali su nezamjenjiv dio svakodnevice, te je u 2023. godini zabilježeno skoro 7 milijardi korisnika [1]. Ova široka upotreba mobilnih uređaja rezultat je njihove sveobuhvatne primjene u svakodnevnom životu, od komunikacije i zabave do produktivnosti i edukacije. U radu M. Daniyal et. al. izmjereno je da studenti u prosjeku koriste mobilni uređaj  $6,16 \pm 2,60$  sati na dan [2], što dodatno naglašava značaj mobilnih uređaja u modernom društvu, posebno među mlađom populacijom.

Jedan od glavnih načina interakcije sa mobilnim uređajem, to jest sa aplikacijama koje se nalaze na njemu, jest kroz uporabu virtualne tipkovnice. Virtualna tipkovnica je postala nezamjenjiv alat za unos teksta, ne samo za slanje poruka i pretraživanje, već i za unos podataka u različitim aplikacijama i uslugama. Prilikom pisanja bilo kakvih poruka ili pretraživanja primarno se koristi virtualna tipkovnica koja prati standardni QWERTY ili QWERTZ raspored te dolazi već instalirana na mobilnom uređaju. Ovaj standardizirani raspored omogućuje korisnicima da bez poteškoća prelaze s fizičkih na virtualne tipkovnice, oslanjajući se na mišićno pamćenje razvijeno kroz dugotrajno korištenje.

Postoje mnogi radovi koji istražuju bolje varijacije ili rasporede standardne tipkovnice, bilo to za jednoručno pisanje [3], dvoručno, ili čak korištenjem posebnih senzora unutar samog mobilnog uređaja [4]. Ove inovacije imaju za cilj poboljšati korisničko iskustvo i prilagoditi tipkovnicu specifičnim potrebama korisnika, čime se povećava učinkovitost i smanjuje napor potreban za unos teksta. Ovaj rad, za razliku od većine drugih, se primarno ne fokusira na poboljšanje postojećeg rasporeda tip-

## *Poglavlje 1. Uvod*

kovnice nego na problem koji uključuje samu veličinu tipkovnice na zaslonu. Dok su prethodna istraživanja uglavnom usmjerena na optimizaciju rasporeda tipki za brži i točniji unos, ovdje se pažnja posvećuje praktičnim aspektima korištenja virtualne tipkovnice, osobito u kontekstu ograničenog prostora na zaslonu mobilnih uređaja. Tipkovnica predstavlja glavni fokus korisniku prilikom unosa teksta te prekriva dragocjeni prostor na relativno malenome ekranu. U većini slučajeva tipkovnica prekriva barem 40% zaslona što je značajan dio koji bi se mogao koristiti za prikaz drugih informacija. Ovaj problem postaje još izraženiji kada aplikacije zahtijevaju istovremeni prikaz dodatnog sadržaja, kao što su obrasci, slike ili interaktivni elementi, što često dovodi do frustracije korisnika koji su prisiljeni spustiti tipkovnicu ili pomicati dio prikazanog ekrana kako bi nastavili s radom.

U radu Li et. al. [5] provedena je anketa nad preko 50 iPad korisnika te je većina sudionika izrazila želju smanjenja prostora kojeg tipkovnica prekriva. Tipkovnica također može smetati kada je potreban stalni prikaz nekog dijela aplikacije. Na primjer, u aplikacijama za komunikaciju ili uređivanje dokumenata, korisnici često trebaju pregledavati sadržaj dok upisuju tekst. Zbog toga je pri izradi određenih aplikacija potrebno paziti na prostor koji će prekriti sama tipkovnica kako bi aplikacija i dalje mogla normalno funkcionirati. Osim prekrivanja velikog dijela ekrana, virtualne tipkovnice također predstavljaju problem prilikom unosa povjerljivih informacija, jer je lako vidljivo koje tipke korisnik pritišće. To otvara pitanja vezana uz sigurnost i privatnost, posebno u javnim ili radnim okruženjima, gdje postoji rizik da netko neovlašteno vidi osjetljive podatke poput lozinki ili financijskih informacija.

U tu svrhu, u ovome radu predstavljena je nevidljiva tipkovnica unutar koje su skoro sve tipke skrivene te se time omogućava skriveni unos podataka i oslobađa se prostor za sadržaj samih aplikacija u kojima bi se takva tipkovnica koristila. Jedine tipke koje nisu skrivene nego su prikazane samo kao obris s nazivom su tipka za razmak i tipka za potvrdu unosa. Ovaj pristup omogućuje korisnicima da unose tekst bez vizualnih smetnji, dok istovremeno pruža osnovne informacije o funkcionalnosti tipki koje su često potrebne za jednostavnu navigaciju i strukturiranje unosa. Zanimljivo je što je već u 1999. godini predstavljen nevidljivi način interakcije [6] čiji je cilj približiti se stvarnosti gdje ne postoji posrednik za komunikaciju. Uz nevidljivu tipkovnicu implementirana je i druga varijacija unutar koje tipke nisu nevidljive već ne sadrže oznake za slova te se prikazuje samo njihov obris. Takva varijacija tip-

## *Poglavlje 1. Uvod*

kovnice i dalje ispunjava zahtjev skrivanja unosa, ali i zaklanja veći dio ekrana. Uz sam izgled tipkovnice u pozadini su izrađena tri različita algoritma koja na drugačije načine pokušavaju predvidjeti što je korisnik htio upisati. Prvo pitanje koje se postavlja pri izradi ovakve tipkovnice jest koliko je korisnik zapravo precizan pri korištenju tipkovnice kada su mu potrebne tipke skrivene. Općeniti odgovor jest da korisnici svakodnevno upotrebljavaju virtualne tipkovnice te im je pisanje teksta postalo refleksivno, to jest razvili su mišićno pamćenje za zadatak upisa teksta.

Mišićno pamćenje oblik je proceduralnog pamćenja koje uključuje konsolidaciju specifičnog motoričkog zadatka u memoriju kroz velik broj ponavljanja. Kada se određeni pokret ponavlja tijekom dužeg razdoblja stvara se dugoročna mišićna memorija za taj zadatak što omogućuje da se taj pokret ili zadatak obavlja s minimalnom količinom svjesnog napora. Ovaj proces smanjuje potrebu za pažnjom i stvara maksimalnu učinkovitost unutar motoričkog i memorijskog sustava. Točan mehanizam mišićnog pamćenja nije u potpunosti istražen i objašnjen, ali većina teorija sugerira da tijekom konsolidacije memorije dolazi do opće redistribucije informacija. Ova redistribucija uključuje premještanje sjećanja iz privremenih memorijskih sustava u dugoročne, gdje postaju stabilnija i otpornija na zaboravljanje. Jedan od ključnih koncepata u razumijevanju ovog procesa je Hebbovo pravilo [7], koje navodi da se sinaptička povezanost mijenja kao funkcija ponavljajuće paljbe ili aktivacije neurona. U kontekstu mišićnog pamćenja, to znači da visoka količina slične simulacije koja proizlazi iz izvođenja istog zadatka dovodi do ponavljanja aktivacije u određenim motoričkim mrežama. Ovo ponavljanje uzrokuje povećanje učinkovitosti tih mreža, što omogućuje brže i preciznije izvođenje zadatka s manje svjesnog napora. Sinapse koje se često aktiviraju zajedno postaju sve jače povezane, što dovodi do trajnih promjena u mozgu koje podržavaju automatsko izvođenje naučenih vještina. Primjeri uključuju vožnju bicikla ili motornog vozila, gdje se vozač s iskustvom ne mora svjesno fokusirati na svaki pojedini korak vožnje, već se može usmjeriti na prometne uvjete ili druge važne aspekte. Slično, pisanje na tipkovnici postaje sve brže i preciznije kako se razvija mišićno pamćenje za raspored tipki. Kroz ove procese, mišićno pamćenje omogućuje izvođenje zadataka s visokom razinom preciznosti i minimalnim svjesnim naporom što postavlja temelje za ideju pisanja na nevidljivoj tipkovnici.

Osim oslanjanja na mišićno pamćenje, pisanju na nevidljivoj tipkovnici pomaže

## *Poglavlje 1. Uvod*

činjenica da većina tipkovnica koristi isti raspored, a praksa pri korištenju fizičkih tipkovnica na računalu omogućuje pisanje bez gledanja u tipke koje se pritišću. Ovaj način pisanja, poznat i kao metoda "touch typing", superioran je u odnosu na "hunt-and-peck" pristup, gdje korisnici gledaju u tipke tijekom pisanja. "Touch typing" oslobađa pogled korisnika na sam tekst koji se unosi, omogućujući pravovremeno ispravljanje bilo kakvih grešaka koje se dogode tijekom upisa. Na temu nevidljive tipkovnice već postoji rad [8] koji će biti detaljnije analiziran u drugom poglavlju. Ovaj rad pokazuje da je korištenje nevidljive tipkovnice izvedivo i da, uz određene prilagodbe, brzina pisanja može biti usporediva sa standardnom tipkovnicom. Stoga, fokus ovog istraživanja nije na usporedbi nevidljive tipkovnice s klasičnom, već na vrednovanju različitih algoritama za predviđanje unosa i varijacija rasporeda tipkovnice.

# Poglavlje 2

## Srodni radovi

Korisnička sučelja su ključni element u interakciji između korisnika i tehnologije, a njihova evolucija neprestano donosi nove izazove i mogućnosti. U kontekstu razvoja naprednih tehnologija, sve više se istražuju koncepti koji teže smanjenju vidljivosti ili potpuno nevidljivim sučeljima, koja korisnicima omogućuju prirodniju i intuitivniju interakciju. Ova ideja, premda još uvijek u razvoju, privlači veliku pažnju unutar znanstvene zajednice, te je postala važan smjer istraživanja u okviru *Human Computer Interaction* (HCI) discipline.

### 2.1 Nevidljiva korisnička sučelja

Kao što je već spomenuto u uvodu, unutar HCI discipline nevidljiva korisnička sučelja predstavljaju izuzetno zanimljivo područje istraživanja te postoje mnogi radovi koji se bave ovom tematikom. Fishkin, Moran, i Harrison [6] predvidjeli su da će korisnička sučelja evoluirati kroz nekoliko stadija, pri čemu je finalni stadij nevidljivo sučelje. Iako današnja korisnička sučelja imaju mnogo prostora za poboljšanje kako bi se uopće približila tom idealu, istraživanje u ovome području je i dalje aktivno i relevantno. Primjer takvog istraživanja je rad Sean, Daniel i Patricka [9] u kojem su predstavljena imaginarna sučelja koja ne zahtijevaju direktnu interakciju. Ova sučelja nude inovativan način interakcije koristeći korisnikove prazne ruke bez ikakve vizualne povratne informacije. U okviru ovoga sučelja, korisnici definiraju prostor za interakciju jednom rukom, dok drugom rukom crtaju ili gestikuliraju kako bi izveli određene akcije. Eksperimentom se nastojalo analizirati granice u kojima korisnici

## *Poglavlje 2. Srodni radovi*

moгу koristiti sučelje koje postoji samo u njihovoj mašti, i otkriveno je da korisnikova mašta može djelomično zamijeniti vizualnu povratnu informaciju koju inače pruža uređaj na ekranu. Danas postoji mnogo uređaja koji koriste sličan princip nevidljivih sučelja, često se oslanjajući na geste ili zvuk za interakciju. Primjeri uključuju moderne kontrole unutar automobila koje omogućavaju vozaču da prihvaća ili odbija pozive, mijenja ton glazbe ili mijenja postavke navigacije koristeći samo geste, čime se vozaču omogućava daljnji pogled na cestu bez ometanja. U pametnim kućama, mnogi uređaji također koriste sličan tip interakcije, kao što je paljenje svjetla na zvuk ili pokret, pomicanje zastora gestama, ili, poput Amazonove Alexe, koji odgovaraju na pitanja korisnika bez potrebe za fizičkom interakcijom. Iako nevidljiva korisnička sučelja predstavljaju veliki potencijal, njihova realizacija zahtjeva veliku količinu istraživanja i razvoja. U mnogim slučajevima, trenutna rješenja koja su već dostupna i funkcionalna ostaju neusporedivo učinkovitija, jer implementacija nevidljivih sučelja može dovesti do složenijih interakcija, manjka povratnih informacija za korisnike te sporijeg usvajanja novih sustava. Također, trenutni sustavi su često pristupačniji i jednostavniji za implementaciju, što je ključno za široku prihvaćenost. Stoga, iako nevidljiva sučelja imaju svoje mjesto u budućem razvoju unutar HCI područja, njihova praktična primjena i dalje zahtijeva pažljivo razmatranje u odnosu na postojeće tehnologije.

## **2.2 Tipkovnice sa smanjenom veličinom**

Među svim korisničkim sučeljima, virtualne tipkovnice predstavljaju vrlo popularno pod-područje istraživanja unutar HCI polja, s fokusom na različite aspekte kao što su ergonomski raspored tipki, optimizacija brzine pisanja, prilagodba određenim vrstama pisanja [3], te dinamičko prilagođavanje rasporeda tipkovnice temeljeno na korisnikovim navikama. Također, istraživanja se bave i rasporedima koji nastoje smanjiti količinu zaslona koju tipkovnica prekriva [10] ili korištenje potpuno drukčijih načina upisa poput primjene senzora svjetla [4]. Tipkovnice smanjenih dimenzija ciljaju riješiti sličan problem kao i nevidljiva tipkovnica opisana u ovom radu — zauzimanje velikog dijela zaslona. Junjue, Zhao i Zhang u svome radu [11] predstavljaju inovativan pristup virtualnim tipkovnicama, omogućujući korisnicima unos teksta na sličan način kao i na fizičkoj tipkovnici, ali koristeći samo tvrdu površinu poput stola. U njihovom rješenju korisnik upisuje slova koristeći papir sa isprintanom



slikom tipkovnice, dok mobilni uređaj koristi ugrađeni mikrofoni kako bi odredio lokaciju dodira i po tome utvrdio koja je tipka pritisnuta. Ovaj pristup eliminira potrebu za prikazom tipkovnice na ekranu, što može ubrzati proces unosa, ali također postavlja ograničenja na fizičku interakciju s uređajem. S obzirom na ove izazove, mnoga istraživanja poput onog od Green et. al [10] zadržavaju korištenje same virtualne tipkovnice, ali nastoje limitirati njezinu veličinu primjenom pametnijih rasporeda. U ovom slučaju, raspored tipkovnice je optimiziran tako da su svi redovi tipkovnice spojeni u samo jedan, koristeći funkcionalnost višestrukog dodira (engl. multi-tap) za odabir pojedinih slova. Ovaj pristup koristi ideju starih mobilnih tipkovnica gdje je različiti broj pritisaka iste tipke rezultirao različitim slovima. Također, u sustav je integrirano predviđanje upisanog teksta koristeći rječnik koji sadrži približno 30 tisuća riječi, što veličinski odgovara rječniku korištenom i u ovome radu.

### **2.3 Dekodiranje tipkovnice**

Osim same prilagodbe rasporeda tipkovnice, brojni radovi su eksperimentirali s različitim metodama dekodiranja korisničkog unosa kako bi poboljšali točnost i učinkovitost. Zbog prirodne nepreciznosti ljudskoga prsta, unos putem virtualnih tipkovnica podložan je šumu i pogreškama, što otežava precizno dekodiranje. Kako bi se ovaj problem prevladao, razvijeni su različiti algoritmi, među kojima su mnogi bazirani na Bayesovom pravilu, koje koristi statistički pristup umjesto determinističkog. Deterministički pristupi, u kojima se svakom dodiru dodjeljuje određena tipka, često nisu dovoljno fleksibilni zbog visoke nepreciznosti unosa. Stoga je sve popularniji statistički pristup, u kojem se iz jednog dodira izračunavaju vjerojatnosti određenog unosa te se dobivene vjerojatnosti kombiniraju kroz više dodira kako bi se preciznije predvidio unos. Ova metoda omogućava preciznije dekodiranje jer uzima u obzir nesigurnost i varijabilnost u unesenim podacima. Daljnje poboljšanje se postiže upotrebom jezičnih modela za izračun vjerojatnosti pojave određene riječi iz grupe mogućih riječi, čime se dodatno povećava točnost unosa. Bi i Zhai [12] predstavili su Bayes Touch statistički kriterij selekcije, koji kombinira Bayesovo pravilo vjerojatnosti s generaliziranom hipotezom dvostruke Gaussove distribucije dodira. Dvostruka Gaussova distribucija dodira koristi dvije različite Gaussove distribucije prilikom dodira prsta na zaslonu: jedna predstavlja kompromis između brzine i preciznosti unosa dok druga modelira nepreciznost ljudskog prsta. Tu distribuciju su

također derivirali Bi i Zhai [13] te su ujedno iz tog rada dobiveni prvobitni parametri Gaussove distribucije koja se koristi unutar algoritama navedenih u ovome radu. Nadovezujući se na taj model, Ma. et. al. [14] predstavljaju napredni model "rotirajuće" dvostruke Gaussove distribucije, uzimajući u obzir i smjer kretanja prsta tijekom unosa. Ovaj model produžuje širinu i visinu mete u prikladnom smjeru, čime se povećava preciznost dekodiranja i poboljšava akvizicija tipke. Uzimanje u obzir smjera kretanja omogućava još veću točnost u prepoznavanju unosa, čime se dodatno unapređuje korisničko iskustvo.

## **2.4 Nevidljiva tipkovnica**

U kontekstu virtualnih tipkovnica na mobilnim uređajima, Zhu et. al. [8] proveli su značajan eksperiment za evaluaciju korisnosti nevidljive tipkovnice, čiji je dizajn sličan prvome rasporedu koji se razmatra u ovom radu. Kako bi uopće ustanovili mogućnost preciznog pisanja na nevidljivoj tipkovnici izvršen je preliminarni eksperiment usmjeren na testiranje varijacija prilikom pisanja te detekcije bilo kakvih promjena unutar samog stila pisanja. Rezultati preliminarnog eksperimenta su ukazali da korisnici puno više griješe po y osi u usporedbi sa standardnom vidljivom tipkovnicom. Na temelju tih podataka izrađen je prostorni model koji je unaprijedio dekodiranje pritisaka u pripadajuće tipke te je istovremeno implementiran jezični model. Jezični model dodatno poboljšava preciznost unosa uzimajući u obzir povijest riječi koje je korisnik prethodno koristio, čime se značajno povećava točnost. Izveden je eksperiment u trajanju od tri dana te su rezultati nakon trećega dana pokazali da ne postoji signifikantna razlika između nevidljive i vidljive tipkovnice u količini grešaka i brzini pisanja. S obzirom na ove rezultate, ovaj rad se ne fokusira na usporedbu nevidljive tipkovnice i vidljive već na analizu algoritama za predviđanje unosa i različitih varijacija tj. rasporeda nevidljive tipkovnice. Iako su prostorni i jezični modeli predstavljeni u radu pokazali visoku učinkovitost, važno je nastaviti istraživanje i eksperimentirati s različitim modelima kako bi se utvrdilo postoji li bolji pristup za unapređenje preciznosti i učinkovitosti unosa na nevidljivim tipkovnicama. Ova istraživanja mogu pridonijeti razvoju naprednijih rješenja koja će poboljšati korisničko iskustvo i učinkovitost u primjeni nevidljivih tipkovnica.

## Poglavlje 3

### Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

Kako bi se omogućilo uspješno korištenje nevidljive virtualne tipkovnice zasnovane na mišićnoj memoriji, ključno je razviti i implementirati temeljne komponente koje čine ovu tehnologiju. U ovom poglavlju razmatraju se ključni koraci u razvoju tipkovničkog servisa unutar Android sustava, kao i algoritmi za predviđanje unosa teksta. Prvo će biti opisani postupci implementacije samog tipkovničkog servisa, koji osigurava osnovne funkcionalnosti potrebne za rad tipkovnice. Nakon toga, pažnja će se usmjeriti na algoritme za predviđanje unosa teksta, uključujući detaljnu razradu različitih pristupa koji su razvijeni kako bi se povećala točnost i učinkovitost tipkanja na nevidljivoj tipkovnici. Ovi elementi zajedno čine osnovu za pripremu i izvođenje potrebnog eksperimenta tj. empirijsku evaluaciju.

#### 3.1 Implementacija tipkovničkog servisa

Prvi korak u implementaciji bilo kakve virtualne tipkovnice jest izrada samoga servisa unutar Android sustava. Svaka virtualna tipkovnica u Android sustavu pokreće se u obliku `InputMethodService` servisa koji upravlja korisničkim unosom kroz niz specifičnih metoda i događaja. Da bi se realiziralo bilo kakvo prilagođeno sučelje, potrebno je napraviti prilagođeni razred koji nasljeđuje `InputMethodService` razred, čime se omogućuje pristup osnovnim funkcionalnostima metode unosa. Nadalje, ključni aspekt kod izrade tipkovnice je implementacija sučelja `KeyboardView.OnKeyboardActionListener` koje pruža osnovu za rukovanje različitim ko-

### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

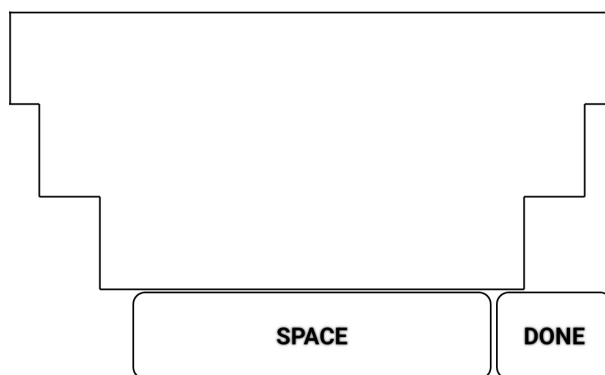
risničkim interakcijama s tipkovnicom. Implementacijom ovog sučelja omogućava se nadjačavanje (engl. *override*) potrebnih metoda poput `onKey()` metode koja se poziva prilikom pritiska na tipku ili metode `swipeRight()` koja se poziva prilikom akcije prelaska prstom udesno preko tipkovnice. Svaka virtualna tipkovnica posjeduje svoj specifični raspored tipki, što zahtijeva izradu posebne datoteke koja definira kako će se tipkovnica prikazivati na zaslonu. Osim rasporeda tipkovnice, potrebna je i datoteka koja sadrži cjelokupni prikaz (engl. *view*) tipkovnice, uključujući informacije o veličini i poziciji elemenata na ekranu. Ovu datoteku je potrebno pročitati prilikom stvaranja prikaza te instancirati razred `Keyboard` i postaviti instancu kao zadanu tipkovnicu.

Iako osnovna implementacija virtualne tipkovnice ne zahtijeva dodatne prilagodbe, u ovom radu je izrađen posebni razred, *CustomKeyboard* koji nasljeđuje razred `Keyboard`. Cilj ove prilagodbe je omogućiti nadjačavanje metode `changeKeyHeight()`, što je ključno za pravilno rukovanje tipkama nestandardne veličine, jer bi u protivnom moglo doći do nepredviđenih problema tijekom korištenja. Raspored tipki se postavlja unutar instance *CustomKeyboard* razreda, s mogućnošću kasnije zamjene tog rasporeda s novim ako se stvori druga instanca razreda i postavi kao zadana. Ova funkcionalnost zamjene omogućava testiranje više varijacija iste tipkovnice tj. rasporeda, bez potrebe za stvaranjem dvaju posebnih servisa.

Kao što je već spomenuto, nevidljiva tipkovnica implementirana u ovome radu može imati dva različita rasporeda: *Invisible keys* i *Blank keys*. Raspored *Invisible keys* sadrži samo osnovne funkcijske tipke: razmak i potvrdu unosa te su sve ostale tipke nepostojeće. Ostale funkcijske tipke su potpuno onemogućene za svrhe testiranja, budući da fraze koje se koriste u eksperimentu ne sadrže velika slova, točke ili zareze. Tipka za brisanje je također onemogućena jer je funkcija brisanja deaktivirana kako bi se izbjegle dodatne komplikacije pri testiranju što će biti detaljnije objašnjeno u potpoglavlju 3.2. Tipke za razmak i potvrdu unosa prikazane su zbog njihove važnosti unutar algoritama za predikciju i samoga testiranja. Naime, predikcija se vrši riječ-po-riječ pa su pogreške pri korištenju tipke za razmak daleko značajnije nego kod standardnih tipkovnica. Tipka za potvrdu unosa označava završetak unosa i ima ključnu ulogu u preciznom mjerenju rezultata, pa je važno da se koristi isključivo kada je to potrebno. Prostor za prikaz tipkovnice nije potpuno proziran - prikazani su obrisi rubova kako bi se korisnicima omogućila osnovna

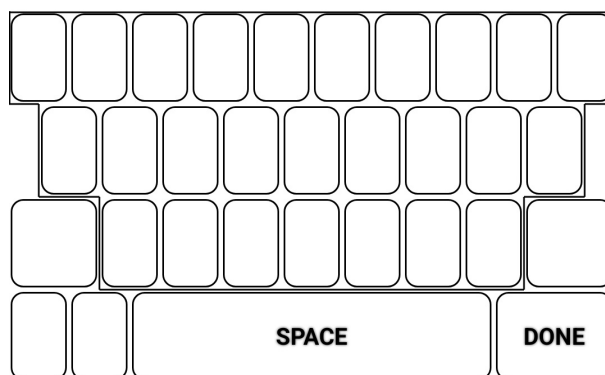
### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

vizualna orijentacija (Slika 3.1). Implementacija ovog obrisa ima dvostruku svrhu: prvo, omogućava korisnicima lakšu i bržu prilagodbu tipkovnici, jer obris pruža barem minimalni vizualni orijentir što olakšava početnu prilagodbu. Drugo, potpuno prozirna tipkovnica mogla bi uzrokovati slučajno isključenje iste, prilikom pritiska izvan njezinih granica, što bi nepotrebno kompliciralo i usporilo eksperiment. Drugi



Slika 3.1 *Invisible keys* dizajn

raspored, *Blank keys*, uključuje sve tipke kao i standardna tipkovnica, ali se za sve tipke osim onih već spomenutih, prikazuje samo obris kako bi se testiralo hoće li se korisnici bolje snalaziti s rasporedom koji je sličniji standardnoj tipkovnici (Slika 3.2). Osim navedenih rasporeda postoji i standardni raspored tipkovnice koji se ni-



Slika 3.2 *Blank keys* dizajn

kada ne prikazuje korisnicima, već se koristi isključivo kao referenca za izračune u

algoritmima za predikciju, koji će biti detaljnije objašnjeni u potpoglavlju 3.2. S obzirom da nevidljiva tipkovnica ne koristi samo pritiske tipki već i same koordinate svakog dodira, bilo je potrebno implementirati `View.onTouchListener` sučelje kako bi se nadjačala metoda `onTouch()` koja se poziva prilikom svakog dodira unutar zadanog prikaza. Prilikom svakog pozivanja ove metode prvo se detektira vrsta dodira, a zatim se na temelju te informacije šalje odgovarajući podatak putem `Broadcast`-a, što omogućuje precizno praćenje interakcije korisnika s tipkovnicom. Spomenuti `Broadcast` se prihvaća u glavnom razredu, a dobivene koordinate dodira pohranjuju se za daljnju uporabu unutar algoritama. Unutar servisa dodatno su nadjačane metode `swipeLeft()` i `swipeRight()` kako bi se omogućila izmjena zadanog rasporeda tipki prelaskom prsta preko tipkovnice udesno ili ulijevo.

## 3.2 Algoritmi za predviđanje unosa teksta

Virtualne tipkovnice podložne su velikoj količini šuma i pogrešaka prilikom unosa teksta zbog njihove relativno malene veličine i nepreciznosti ljudskoga prsta u takvim zadacima. Ova pojava posebno dolazi do izražaja unutar nevidljivih tipkovnica jer je, kao što je dokazano u spomenutom radu *Typing on an Invisible Keyboard* [8], šum kod unosa na takvoj tipkovnici znatno veći nego kod unosa na standardnoj. Zbog toga je potrebno implementirati posebne algoritme kako bi se povećala preciznost unosa. U ovome radu opisana su 2 algoritma za predviđanje unosa: *Physical distance* i *String distance*, zajedno sa referentnim algoritmom koji ne mijenja korisnikov unos, već se ponaša kao da korisnik upotrebljava običnu tipkovnicu, kako bi se mogla analizirati njihova učinkovitost.

### 3.2.1 Referentni algoritam

Referentni algoritam funkcionira tako da se za svaki korisnikov dodir dodijeli odgovarajuće slovo u ulaznom nizu znakova. Za postizanje ove funkcionalnosti koristi se *BroadcastReceiver* za čitanje koordinata dodira koje se šalju iz servisa tipkovnice prilikom svakog poziva metode `onTouch()`. Koristeći dobivene koordinate, prvo se utvrđuje ako poslani dodir pripada pritisku jednoj od vidljivih tipki koja se i dalje nalazi na tipkovnici ili jednoj od onemogućenih tipki. Ova provjera je potrebna jer `onTouch()` događaj ne "iskoristi" dodir, već ga prosljeđuje dalje kako bi se aktivi-

### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

rale potrebne funkcije poput `onKey()`. Da takva provjera ne postoji, svaki bi pritisak jedne od vidljivih tipki rezultirao registracijom dodatnog pritiska nevidljive tipke, što bi stvorilo nepotrebne greške. Nakon ove inicijalne provjere, ako dodir zaista pripada jednoj od nevidljivih tipki, izračunava se udaljenost između dobivenih koordinata i svake nevidljive tipke. Lista nevidljivih tipki inicijalizira se iz prethodno spomenutog standardnog rasporeda koji se učitava prilikom svakog prikazivanja tipkovnice. Za jednostavniju manipulaciju i zapis ovih tipki, definiran je posebni razred `CustomKey`, koji olakšava rukovanje tipkama i sadrži metodu za precizno izračunavanje njihove udaljenosti od zadanih koordinata. Udaljenost se računa putem Euklidske udaljenosti, što je prikazano u formuli 3.1

$$d(A, B) = \sqrt{(x_A - x_B)^2} + \sqrt{(y_A - y_B)^2} \quad (3.1)$$

gdje su A i B dvije različite točke. Koordinate točke A po x i y osi su označene kao  $x_A$ ,  $y_A$ , dok su koordinate točke B označene kao  $x_B$ ,  $y_B$ . Ovaj prilagođeni razred omogućava fleksibilnije upravljanje nevidljivim tipkama i jednostavniji izračun udaljenosti, što je od ključne važnosti za pravilno funkcioniranje algoritama za predviđanje unosa. Navedena izračunata udaljenost se zatim prosljeđuje kao ulaz u Gaussovu distribuciju kako bi se izračunala vjerojatnost stvarnog pritiska na određenu tipku. Korištenje Gaussove distribucije umjesto obične udaljenosti ili kvadrirane udaljenosti omogućuje bolje modeliranje prirodne greške, jer veće udaljenosti imaju vjerojatnost blizu nule. Kao što je već spomenuto, mnogi radovi predlažu različite varijacije Gaussove distribucije za točnije modeliranje dodira ili grešaka na virtualnim tipkovnicama [13, 14]. U ovome radu standardna devijacija za korištenu Gaussovu distribuciju iznosi 8mm, te je dobivena povećanjem devijacije izračunate u radu *FFitts law: modeling finger touch with fitts' law* [13]. Povećanje je izvršeno kako bi se uzela u obzir veća nepreciznost prilikom unosa unutar nevidljive tipkovnice. Iako bi se za određivanje najbliže tipke zadanom dodiru mogla koristiti obična udaljenost, ova se izračunata vrijednost također koristi unutar *Physical distance* algoritma, pa je zbog konzistentnosti iskorištena i za određivanje najbližeg slova unutar referentnog algoritma.

Također je razmatrana i implementacija četvrtog algoritma, koji bi se oslanjao na postojeće modele predikcije koji se koriste unutar popularnih virtualnih tipkovnica. Te modele moguće je koristiti putem `SpellCheckerService` servisa, koji se povezuje s trenutno odabranim prediktivnim modelom unutar samoga uređaja. Za uporabu takvih modela, potrebno je imati instaliranu tipkovnicu koja takav model

### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

sadrži, npr. tipkovnica *Gboard*. Unutar tog algoritma korisnikov unos bi se zapisivao na isti način kao i unutar referentnog algoritma, a zatim bi se dobiveni niz proslijedio u metodu `getSentenceSuggestions()` kako bi se dobio konačni rezultat. Međutim, implementacija ovog algoritma je naposljetku odbačena iz dva razloga. Prvo, testiranje bi se trebalo izvršiti na istom uređaju zbog razlika u prediktivnim modelima između različitih uređaja. Drugo, ispravci koje su ti modeli predlagali su često bili pogrešni ili uopće nisu bili generirani, što je dodatno smanjilo učinkovitost predloženog algoritma.

#### 3.2.2 Physical distance algoritam

*Physical distance* algoritam u početnoj fazi funkcionira isto kao i referentni algoritam, no umjesto jednostavnog odabira slova s najvećom vjerojatnosti, pohranjuje sva slova i njihove pripadajuće vjerojatnosti u mapu, gdje je svako slovo ključ, a izračunata vjerojatnost njegova vrijednost. Ta se mapa zatim dodaje u niz, a postupak se ponavlja za svaki novi korisnikov dodir na tipkovnici. U trenutku kada korisnik pritisne tipku za razmak ili tipku potvrde unosa, nakon osnovnih provjera, pokreće se predikcija upisane riječi koristeći već spomenuti niz mapa. Predikcija se provodi nad listom od 30,000 najkorištenijih engleskih riječi, koja je izvedena iz Peter Norvigove kompilacije 1/3 milijuna najčešćih engleskih riječi. Nekoliko neispravnih elemenata ručno je uklonjeno iz liste zbog njihove učestale pojave tijekom testiranja, što je ometalo točnost algoritma. Ti elementi su većinom bili neispravne varijacije čistih riječi poput *the*, *and*, *is* i slično. Tijekom predikcije, algoritam prolazi kroz cijelu listu riječi i traži one sa duljinom jednakom broju dodira unutar niza. Za svaku pronađenu riječ, zbrajaju se vrijednosti svakoga slova unutar riječi, koristeći se spomenutim nizom mapa. Dobiveni rezultat se zatim pohranjuje u *TreeMap* radi automatskog sortiranja. Nakon što algoritam obradi sve riječi, odabire se riječ koja se nalazi na prvom mjestu unutar mape, odnosno riječ s najvećom ukupnom vrijednošću. Unutar spomenute liste riječi za predikciju, također se nalazi dodatna vrijednost za svaku riječ koja predstavlja broj ponavljanja te riječi unutar izvornog skupa podataka iz kojeg su preuzete: *Trillion Word Corpus Dataset* [15]. Ova vrijednost (*freq*), izražena kao logaritam broja ponavljanja, normalizira se i koristi tijekom izračuna za povećanje vrijednosti riječi za maksimalno 20% (*w*). Ovaj izračun prikazan je u



### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

formuli 3.2

$$total = total \cdot \left(1 + \frac{\log(freq) - \log(minFreq)}{\log(maxFreq) - \log(minFreq)} \cdot w\right) \quad (3.2)$$

gdje je:

- $freq$  - broj ponavljanja riječi
- $maxFreq$  - najveći broj ponavljanja riječi
- $minFreq$  - najmanji broj ponavljanja riječi
- $total$  - vrijednost riječi unutar algoritma

Koristi se logaritam broja ponavljanja riječi umjesto samog broja kako bi se smanjila razlika između učestalosti riječi. Na primjer, prva riječ na listi (*the*) se pojavljuje gotovo dvostruko češće od druge riječi na listi (*of*). Riječi na samome početku liste imaju toliko visoke vrijednosti da bi direktno korištenje broja ponavljanja moglo značajno iskriviti rezultate predikcije. Logaritamska transformacija pomaže uravnotežiti te vrijednosti, približavajući ih i omogućujući precizniju predikciju. Iako prolazak kroz cijelu listu nije najučinkovitiji pristup u smislu složenosti algoritma, brzina izvršavanja je dovoljno visoka da ne predstavlja problem pri unosu teksta. Također, omogućen je prikaz prvih  $n$  najvjerojatnijih riječi, no ta funkcionalnost se ne koristi unutar tipkovnice jer se fokus stavlja na automatsku predikciju bez dodatne interakcije korisnika.

Zbog toga što se predikcija izvršava prilikom završetka upisa riječi, implementacija tipke za brisanje predstavlja nekoliko problema. Prvo, korisnik ne može unaprijed znati koju će riječ algoritam predvidjeti, pa nema smisla pokušavati brisati tijekom unosa same riječi. Drugo, brisanje nakon što je riječ već generirana je komplicirano jer bi zahtijevalo ponovno generiranje cijele riječi. U slučaju posljednje unesene riječi, to bi značilo da korisnik mora čekati rezultat algoritma kako bi odlučio treba li izvršiti ispravke, što može biti frustrirajuće. Također, korisnik ne može biti siguran u kojem je točno slovu došlo do greške jer se predikcija riječi izvršava koristeći kombinaciju svih pritisaka. Treći problem proizlazi iz toga što implementirani algoritmi vrlo često generiraju međusobno različite rezultate. To znači da bi svaki od njih zahtijevao različitu količinu brisanja, što dodatno komplicira proces. Da bi se izbjegli ovi problemi, bilo bi potrebno provesti zasebne testove za svaki algoritam, što bi značajno povećalo trajanje eksperimenta. Zbog svega navedenog mogućnost

brisanja je onemogućena.

### 3.2.3 String distance algoritam

*String distance* algoritam vrši predikciju koristeći samo unesene znakove, ignorirajući točne koordinate dodira. Stoga, ulaz u *String distance* algoritam je zapravo izlaz referentnog algoritma. Slično kao i *Physical distance* algoritam, *String distance* algoritam se pokreće svaki put kada korisnik pritisne tipku razmak ili potvrdu unosa što označava kraj unosa jedne riječi. Klasični algoritmi udaljenosti niza (engl. string distance) koriste se za izračun udaljenosti između dva različita niza znakova pomoću tri osnove operacije: brisanje, dodavanje i promjena znaka. Svaka od tih operacija ima istu vrijednost, obično jedan, i računa se koliko je takvih operacija potrebno da se jedan niz pretvori u drugi, s time da se svaka operacija poziva isključivo nad jednim znakom. Takvi algoritmi se obično koriste za usporedbu dužih nizova znakova, gdje je veći broj operacija potreban za pretvaranje jednog niza u drugi. Međutim, u kontekstu ovog eksperimenta, *String distance* algoritam koristi se za predviđanje najbližijih riječi, što predstavlja poseban izazov jer su riječi relativno kratke, pa mnoge od njih imaju jednaku udaljenost. Kako bi se riješio ovaj problem, bilo je potrebno implementirati dodatni kriterij za određivanje "pobjednika" među riječima s jednakom udaljenošću.

U početku je kao sekundarni kriterij korišten broj ponavljanja riječi u korpusu, ali je to dovelo do vrlo čestog odabira najpopularnijih riječi, što nije bilo idealno za preciznost predikcije. Stoga je donesena odluka da se uvede dio logike iz *Physical distance* algoritma kako bi se poboljšala točnost predikcije. Implementirana je matrica koja sadrži međusobne udaljenosti svih tipki na virtualnoj tipkovnici, te se navedena matrica koristila za određivanje cijene operacije zamjene slova. Za razliku od klasičnog algoritma udaljenosti niza gdje sve operacije imaju jednaku cijenu, u ovome slučaju cijena operacije zamjene je varijabilna. Cijena se kreće u rasponu od 0.5 do 2, ovisno o fizičkoj udaljenosti između slova koja se zamjenjuju. Implementacijom varijabilne funkcije cijene značajno se smanjio broj riječi s istom udaljenosti te, iako je sekundarni kriterij frekvencije riječi zadržan, njegov utjecaj je smanjen na prihvatljivu razinu, što je rezultiralo balansiranim i točnijim rezultatima u predviđanju unosa. Sama vrijednost cijene se normalizirala koristeći obrnuti rezultat

### Poglavlje 3. Razvoj metode unosa teksta zasnovanoj na mišićnoj memoriji

dobiven iz Gaussove distribucije, pri čemu je standardna devijacija bila povećana na 11.6 mm. Izračun cijene je prikazan na formuli 3.3

$$\begin{aligned} \min &= 0.5 \quad \max = 2 \\ w &= \min + (\max - \min) \cdot (1 - \text{normGauss}) \end{aligned} \quad (3.3)$$

gdje je:

- $\max$  - maksimalni iznos cijene
- $\min$  - minimalni iznos cijene
- $\text{normGauss}$  - normalizirana vrijednost iz Gaussove distribucije

Ovo povećanje standardne devijacije imalo je cilj osigurati da susjedne tipke rezultiraju vrijednošću manjom od 1, dok udaljenije tipke rezultiraju vrijednošću bližom 2, čime se potiče zamjena bliskih slova, a odvraća od zamjene udaljenijih. Sa početnom devijacijom od 8mm susjedne tipke po y osi bi rezultirale s vrijednošću većom od 1, što je poništavalo ideju algoritma. Predikcija riječi u ovom algoritmu koristi sličan pristup kao i u *Physical distance* algoritmu, s tom razlikom što se ne razmatraju samo riječi iste duljine, već i one koje su za jedno slovo duže ili kraće od unesene riječi. U klasičnim algoritmima za udaljenosti niza ovaj kriterij ne postoji, već se nizovi uspoređuju neovisno o duljini. No, u ovome je radu, zbog optimizacije, duljina ograničena na jedan znak razlike. Ova fleksibilnost omogućuje *String distance* algoritmu da ispravno predvidi riječi čak i ako korisnik unese jedno slovo više ili manje, što je prednost u odnosu na *Physical distance* algoritam, koji nema tu vrstu prilagodljivosti.

# Poglavlje 4

## Empirijsko istraživanje učinkovitosti (HCI eksperiment)

Kako bi se procijenila učinkovitost različitih rasporeda nevidljive tipkovnice i algoritama predikcije unosa teksta, potrebno je provesti detaljno empirijsko istraživanje. Ovo poglavlje opisuje ključne aspekte eksperimenta koji je dizajniran i proveden za ovu svrhu. Prvo će biti predstavljena ispitna aplikacija, koja je korištena za prikupljanje podataka i mjerenje performansi unosa teksta. Zatim će biti opisan uzorak ispitanika u eksperimentu, kao i karakteristike same skupine sudionika. Na kraju, opisana je procedura eksperimenta, koja uključuje način na koji su podaci prikupljeni i mjere opreza koje su poduzete kako bi se osigurala valjanost rezultata.

### 4.1 Ispitna aplikacija

Kako bi se omogućila empirijska evaluacija različitih rasporeda nevidljive tipkovnice i implementiranih algoritama predikcije, potrebno je uspostaviti okruženje u kojem se mogu mjeriti određene metrike performansi. Jedna od osnovnih metrika za procjenu učinkovitosti unosa teksta je *Words Per Minute* (WPM), što je standardna jedinica za mjerenje brzine tipkanja. Kao što naziv sugerira, WPM predstavlja prosječan broj riječi koje korisnik uspije upisati u minuti. Izračunava se na temelju broja svih unesenih znakova podijeljenih s pet, uz množenje rezultata s omjerom minute prema izmjenenom vremenu (Formula 4.1).

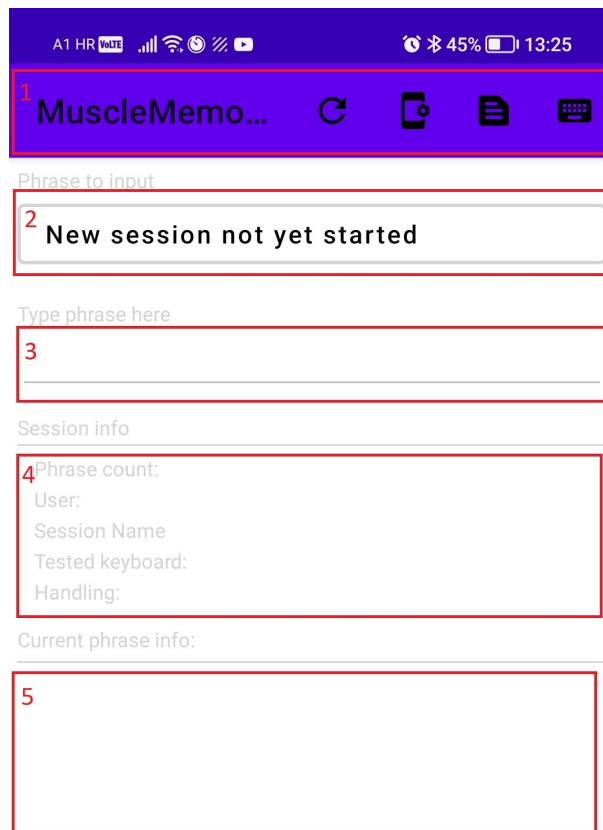
$$\text{WPM} = \frac{\text{brojZnakova}}{5} * \frac{60}{\text{vrijeme/s}} \quad (4.1)$$

WPM metrika sama po sebi nije dovoljna za preciznu evaluaciju unosa teksta jer ne uzima u obzir broj pogrešaka, već se fokusira isključivo na ukupan broj unesenih znakova. Zbog toga je potrebno koristiti dodatne metrike koje uzimaju u obzir točnost unosa. U tom kontekstu, R. William Soukoreff i I. Scott MacKenzie razvili su nekoliko specijaliziranih metrika za mjerenje broja pogrešaka: *Corrected Error Rate* (CER), *Not Corrected Error Rate* (NCER) i *Total Error Rate* (TER) [16]. CER metrika mjeri broj ispravljenih pogrešaka, što je posebno važno za tipkovnice gdje korisnici mogu ispraviti svoje greške. Međutim, budući da u ovom eksperimentu brisanje nije dozvoljeno, CER nije primjenjiv. NCER metrika mjeri broj pogrešno upisanih znakova dok se TER metrika izračunava kombinacijom NCER i CER metrike. Zbog nepostojanja mogućnosti brisanja, TER metrika postaje jednaka NCER metrici, te se stoga za procjenu točnosti unosa u ovome radu koristi samo TER metrika. U ovom slučaju TER metrika se računa kao postotak teksta koji je pogrešno unesen, te se dobiva podjelom broja pogrešno unesenih znakova tj. udaljenosti, s ukupnim brojem znakova (Formula 4.2).

$$\text{TER} = \frac{\text{udaljenost}}{\text{duljinaNiza}} * 100\% \quad (4.2)$$

Način dobivanja broja pogrešno unesenih znakova ili udaljenosti je zapravo već spomenut u ovome radu te se odnosi na klasični *String distance* algoritam bez ikakvih promjena. Za izračun navedenih metrika, često se koristi specijalizirana aplikacija koja omogućuje testiranje različitih tipkovnica. Takva aplikacija omogućava postavljanje tipkovnice koja se vrednuje i prikazuje fraze koje korisnik treba unijeti što točnije i brže moguće. Tijekom testiranja, korisnik prati zadane upute koje mogu uključivati stil interakcije (npr. jednoručno), orijentaciju ekrana (npr. portretna), i druge specifične zahtjeve. Zadane metrike se računaju posebno za svaku frazu te mjerenje vremena počinje tek kada korisnik odabere polje za unos nove fraze ili pritisne gumb za dalje, ovisno o načinu implementacije unutar aplikacije. Ovakav način testiranja osigurava precizno praćenje performansi tipkovnice u stvarnim uvjetima upisa. Zbog jednostavnosti i učinkovitije implementacije algoritama, aplikacija *Muscle Memory Keyboard* uključuje samu tipkovnicu i okruženje za testiranje. Okruženje se sastoji od (Slika 4.1):

- 1 - Alatna traka
- 2 - Prostor za prikaz fraze koju korisnik mora upisati



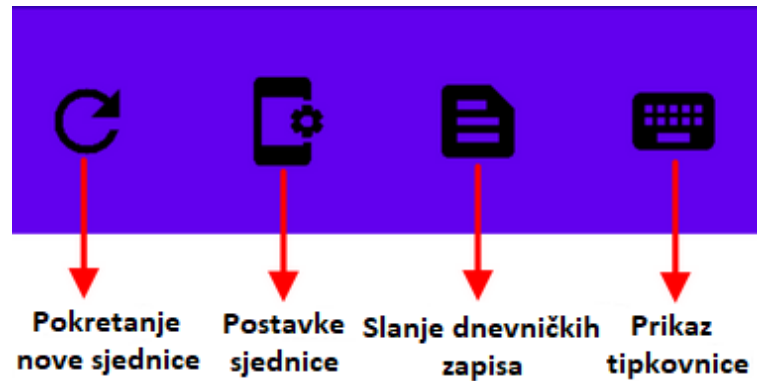
Slika 4.1 Okruženje za testiranje

- 3 - Polje za prikaz unosa
- 4 - Informacije o trenutnoj sjednici eksperimenta
- 5 - Prostor za prikaz izmjerenih metrika

Alatna traka sadrži četiri gumba (Slika 4.2):

- Početak nove sjednice
- Promjena postavki sjednice
- Slanje (engl. upload) podataka sjednice
- Prikaz tipkovnice

U ovome kontekstu sjednica predstavlja korisnikov unos unaprijed određenog broja



Slika 4.2 Alatna traka

fraza koristeći specifičnu varijaciju tipkovnice. Implementacija sjednice izvedena je putem posebnog razreda pod nazivom *Session*, koji upravlja svim ključnim parametrima potrebnim za provođenjem testa. Implementirani razred uključuje sljedeće podatke:

- Ime korisnika: Identifikator za svakog korisnika.
- Ime sjednice: Naziv za razlikovanje između različitih sjednica unutar aplikacije.
- Aktivna tipkovnica: Tipkovnica koja se koristi za unos unutar trenutne sjednice.
- Broj fraza: Ukupan broj fraza koje korisnik treba upisati tijekom sjednice.
- Preferirani raspored tipki: QWERTZ ili QWERTY.
- Način pisanja: Jednoručno, dvoručno palcem, ili jednoručno palcem.
- Orijentacija ekrana: Portretna ili pejzažna.

Korisnik ove podatke prilagođava putem *SessionSettings* aktivnosti (Slika 4.3), koja se pokreće pritiskom pripadajućeg gumba na alatnoj traci.

Osim osnovnih podataka, razred *Session* obuhvaća i metode za izračunavanje mjerenih metrika koje se računaju za svaki algoritam prilikom završetka unosa fraze. Mjerenje vremena za unos fraze započinje tek kada korisnik odabere polje za unos, te je nakon svake unesene fraze potrebno ponovno odabrati polje za početak sljedeće fraze. Izračunate metrike pohranjuju se unutar *Session* razreda u obliku mape, a ko-

Username  
username

Session name  
session1

Keyboard  
Invisible keys ▼

Keyboard layout  
QWERTZ ▼

Number of phrases  
30

Interaction style

One hand  Two thumbs  Cradle

Orientation

Portrait  Landscape

Slika 4.3 *Session settings* aktivnost

risniku se prikazuju u specijaliziranom prostoru za prikaz unutar aplikacije. Također, izračunate metrike se prosljeđuju razredu *Logger*, koji će biti detaljnije objašnjen u drugom dijelu ovog poglavlja. Fraze koje se prikazuju korisnicima nasumično se odabiru iz liste od 500 fraza za testiranje unosa teksta, koju su izradili I. Scott MacKenzie i R. William Soukoreff [17]. Nakon što je određena fraza prikazana korisniku, privremeno se uklanja iz liste kako bi se spriječilo ponavljanje. Zbog specifičnih zahtjeva algoritama *Physical distance* i *String distance*, koji za predikciju koriste ograničene liste riječi, oko 30 fraza je uklonjeno iz testne liste zbog sadržavanja riječi koje nisu prisutne u korištenim rječnicima.

Osim tipki za upravljanje sjednicom, alatna traka također uključuje i gumb za



prikaz tipkovnice, koji je potreban kada korisnik mora promijeniti tipkovnicu ili raspored nevidljive tipkovnice prijelazom prsta udesno ili ulijevo. Da bi se omogućila empirijska evaluacija algoritama koristeći prikupljene metrike, podaci spremljeni u razredu *Session* moraju se pohraniti u posebnu datoteku. U tu svrhu, razvijen je razred *Logger*, koji strukturira i zapisuje mjerene metrike iz *Session* razreda u datoteku nazvanu prema formatu *sessionName-userName*. Svaka linija u datoteci predstavlja unos jedne fraze i sadrži osnovne podatke o sjednici, kao što su ime sjednice, ime korisnika, korištena tipkovnica, i drugi relevantni detalji. Uz te osnovne podatke, svaka linija uključuje vrijeme trajanje upisa fraze, originalnu frazu, upisanu frazu za svaki od tri algoritma, te izmjerene metrike za svaki od ta tri algoritma: WPM i TER. Iako eksperiment uključuje dvije različite sjednice, jer svaka varijacija tipkovnice zahtijeva novu sjednicu, svi podaci se pohranjuju unutar jedne datoteke. Budući da se eksperiment provodio na više uređaja, bilo je potrebno omogućiti slanje datoteka kako bi korisnici izbjegli potrebu traženja datoteka na datotečnom sustavu uređaja i ručno slanje istih. U ovu svrhu korišten je *Firebase*, Googleova platforma za razvoj mobilnih aplikacija koja nudi razne usluge, uključujući autentikaciju, mjerenje performansi i daljinsko pozivanje funkcija. U ovom eksperimentu korištena je samo funkcionalnost *Firebase Storage*, koja omogućava pohranu svih potrebnih datoteka na centraliziranom mjestu. Ovaj pristup pojednostavljuje proces prikupljanja i organizacije podataka, osiguravajući da su svi rezultati lako dostupni za daljnju obradu i evaluaciju.

## 4.2 Procedura eksperimenta i sudionici

Kao što je već spomenuto, za provedbu empirijskog vrednovanja implementiranih algoritama i rasporeda tipkovnice, potrebno je izvesti eksperiment u kojem se navedene komponente testiraju. U eksperimentu je sudjelovalo 12 sudionika te im je zadatak bio upis fraza unutar prethodno opisanog testnog okruženja. Upis fraza se vršio dvoručno palcem u portretnoj orijentaciji ekrana. Svi sudionici su bili muškarci, većinom studenti godina 19-24 s prosjekom godina od 23,67. Svi sudionici su izjavili da im je preferirani način pisanja dvoručno palcem te preferirani raspored tipkovnice standardni QWERTZ, što odgovara postavkama eksperimenta. Također, kao što je prikazano na slici 4.4, velika većina sudionika prijavljuje svakodnevno izvođenje 100 do 150 zadataka pisanja na mobilnom uređaju. Iz toga se može zaključiti da korisnici

Broj zadataka pisanja po danu



Slika 4.4 Prosječan dnevni broj zadataka pisanja sudionika, na mobilnom uređaju

imaju veliku količinu iskustva te izraženu naviku upisivanja teksta na svojim mobilnim uređajima. Ovaj podatak je vrlo bitan za izvođenje eksperimenta jer se obje varijacije tipkovnice oslanjaju na iskustvo sudionika u korištenju standardne virtualne tipkovnice. Iako je ovo iskustvo vrlo korisno prilikom korištenja *MuscleMemory* tipkovnice, ipak je bilo potrebno svakome sudioniku pružiti određeni broj trening fraza. Kako bi se izbjegle komplikacije s posebnim trening sjednicama, ukupan broj fraza za upis sa određenim rasporedom tipkovnice je bio 50, te je prvih 20 fraza funkcioniralo kao trening fraze, tj. nisu se razmatrale u završnoj analizi. Svaki sudionik je koristio oba rasporeda tipkovnice: *Blank keys* i *Invisible keys*, rezultirajući u ukupnome upisu od 100 fraza po sudioniku, od kojih je 40 odvojeno za trening. Tijekom upisa fraza računate su razne metrike, no najbitnije su one već spomenute: WPM i TER. Zbog činjenice da svaki sudionik u eksperimentu koristi oba rasporeda tipkovnice bilo je potrebno izvesti protu-balansiranje poretka, tj. sudionicima zadati drukčiji raspored tipkovnice za unos prvih 50 fraza. Zbog sličnosti rasporeda tipkovnica, upis potrebnih 50 fraza na prvom rasporedu bi donekle služio kao trening za korištenje drugog rasporeda. U slučaju kada bi svaki sudionik koristio istu varijaciju tipkovnice kao prvu, druga varijacija bi imala nepravednu prednost, te bi rezultati eksperimenta bili nevažeci. U svrhu postizanja protu-balansiranja poretka pola sudionika je imalo prvu zadanu varijaciju *Blank keys*, a druga polovica varijaciju *Invisible keys*.

Za samo izvođenje eksperimenta sudionicima su pružene detaljne upute kako instalirati i postaviti aplikaciju, te kako provesti zadani eksperiment. Jedini uvjet za sudjelovanje bio je posjedovanje pametnog telefona s Android operacijskim sustavom, te je u slučajevima kada to nije bilo moguće, sudionicima omogućeno izvođenje eksperimenta na već postavljenom uređaju. Nakon instalacije i namještanja *Muscle-Memory* tipkovnice, sudionici su izvršili spomenuti zadatak upisa 50 fraza sa obje varijacije tipkovnice, te prikupljene podatke poslali putem spomenutog *Logger* razreda. Nakon završetka zadatka s oba rasporeda tipkovnice, sudionici su bili pozvani popuniti upitnik koji se sastojao od ocjenjivanja atributa upotrebljivosti i NASA-*Task Load Index* (TLX) ocjene (Raw-TLX), što je široko korišten alat za procjenu percipiranog radnog opterećenja. Atributi upotrebljivosti ocjenjivali su se na Likert skali sa 7 razina te su se sastojali od: učinkovitosti, memorabilnosti, lakoće učenja, zadovoljstvu i količini pogrešaka. NASA-TLX faktori su se ocjenjivali na skali od 21 razine te su se sastojali od: mentalnog napora, fizičkog napora, frustracije, procijenjene učinkovitosti i ukupnog uloženog truda. Atributi upotrebljivosti i NASA-TLX faktori su se ocjenjivali za oba rasporeda tipkovnice, kako bi se kasnije rezultati mogli usporediti. Korisnici nisu ocjenjivali algoritme jer se predikcija izvodila u pozadini pa nije bilo lako moguće procijeniti njihov utjecaj.

## Poglavlje 5

### Rezultati i diskusija

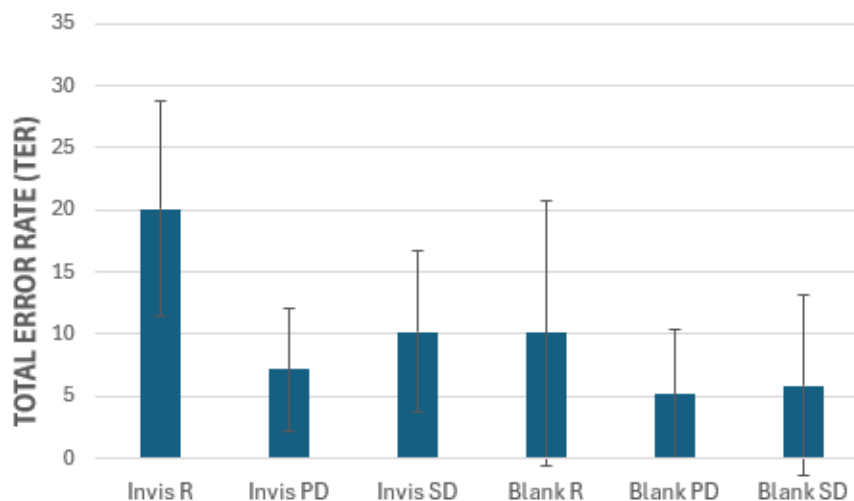
Prvi korak u analizi podataka je izračun prosjeka TER metrike za svaku kombinaciju nezavisnih varijabli, što su u ovome slučaju: *algoritam* i *raspored* tipkovnice. Uz sam prosjek TER metrike, izračunata je i pripadajuća standardna devijacija. Za svaku kombinaciju nezavisnih varijabli proučava se samo TER metrika jer je WPM metrika jednaka za sve algoritme unutar određene varijacije tipkovnice. TER metrika je izračunata iz zadatka upisa 50 fraza, s time da je računato samo zadnjih 30 zbog već spomenutog treninga. Rezultati su prikazani u Tablici 5.1, gdje TER metrika predstavlja postotak krivo unesene fraze, što znači da je niža vrijednost bolja. Također, na slici 5.1 prikazane su iste vrijednosti TER metrike u obliku stupčastog grafikona.

Tablica 5.1 Vrijednosti TER metrike za svaku kombinaciju varijabli *algoritam* i *raspored*

Algoritam	<i>Invisible keys</i>	<i>Blank keys</i>
<i>Raw</i> (Referentni)	20,097 $\pm$ 8,709%	10,11 $\pm$ 10,657%
<i>Physical distance</i>	7,141 $\pm$ 4,945%	5,165 $\pm$ 5,239%
<i>String distance</i>	10,205 $\pm$ 6,459%	5,871 $\pm$ 7,197%

Kao što se može vidjeti iz tablice, vrijednost TER metrike za referentni algoritam (*Raw*) unutar *Invisible keys* rasporeda iznosi 20,097  $\pm$ 8,709%, što je najviša vrijednost unutar rezultata. Vrijednost TER metrike za *Physical distance* algoritam unutar tog istog rasporeda iznosi 7,141  $\pm$ 4,945%, što je ujedno i najniža vrijednost unutar *Invisible keys* rasporeda, dok za *String distance* algoritam iznosi 10,205  $\pm$ 6,459%. Vrijednost TER metrike za referentni algoritam unutar *Blank keys* rasporeda iznosi

## Poglavlje 5. Rezultati i diskusija

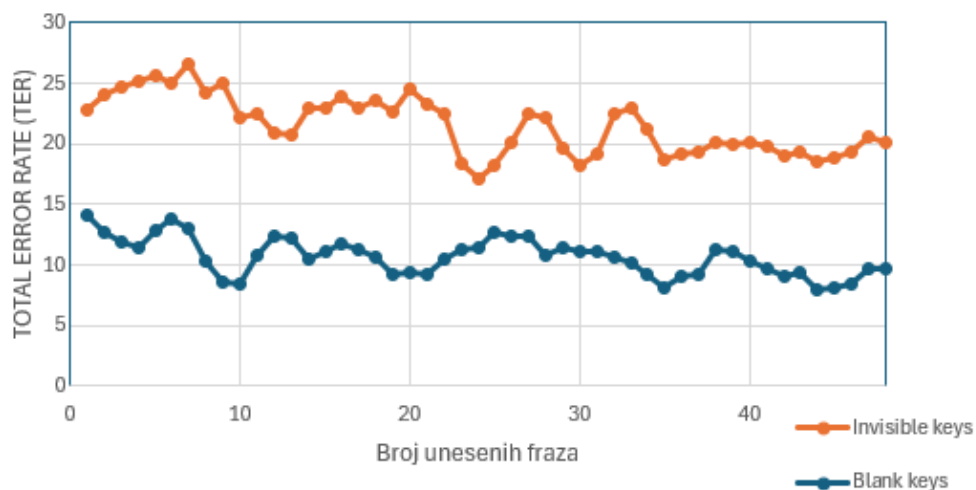


Slika 5.1 TER metrika za sve kombinacije varijabli *raspored* (Invis - *Invisible keys*, Blank - *Blank keys*) i *algoritam* (R - Referentni, PD - *Physical distance*, SD - *String distance*)

$10,11 \pm 10,657\%$  što je najveća vrijednost unutar tog rasporeda. Vrijednost TER metrike za *Physical distance* je najniža unutar rezultata:  $5,165 \pm 5,239\%$ , dok za *String distance* TER metrika iznosi  $5,871 \pm 7,197\%$  što je puno bliže vrijednosti *Physical distance* algoritma nego što je slučaj unutar *Invisible keys* rasporeda. Također, moguće je primijetiti da su vrijednosti TER metrike općenito niže unutar *Blank keys* rasporeda u usporedbi s *Invisible keys* rasporedom. Isto, vrijedi napomenuti visoku vrijednost standardne devijacije unutar svake grupe, osobito unutar *Blank keys* rasporeda gdje vrijednost devijacije nadmašuje prosjek. Naravno, samo iz ovih podataka nije moguće dokazati ako postoji signifikantna razlika unutar varijabli *raspored* i *algoritam*, pa je potrebno nastaviti s analizom. Na slici 5.2 je prikazana promjena TER metrike za referentni algoritam u odnosu na korisnikov broj upisanih fraza.

Iz pada vrijednosti može se zaključiti da korisnici postaju bolji što duže koriste tipkovnicu, što je očekivano, i ujedno razlog primjene trening fraza. Iz slične promjene u TER metrici za oba rasporeda tipkovnice može se pretpostaviti da bi veća količina treninga rezultirala sličnim poboljšanjima za oba rasporeda. Za formalnu analizu rasporeda tipkovnice i algoritama korišten je Two way Anova *Repeated Measures* (RM) test. Za provjeru sferičnosti (engl. sphericity) koristio se Mauchlyev

Poglavlje 5. Rezultati i diskusija



Slika 5.2 Promjena TER metrike za referentni algoritam, u odnosu na broj upisanih fraza

test sferičnosti, koji se zajedno sa potrebnim korekcijama, automatski izvršava unutar okruženja IBM *Statistical Product and Service Solutions* (SPSS) prilikom Two way Anova RM testa. Na slici 5.3 prikazani su rezultati Mauchly-evog testa sferičnosti za nezavisne varijable te interakciju između varijabli.

**Mauchly's Test of Sphericity<sup>a</sup>**

Measure: MEASURE\_1

Within Subjects Effect	Mauchly's W	Approx. Chi-Square	df	Sig.	Epsilon <sup>b</sup>		
					Greenhouse-Geisser	Huynh-Feldt	Lower-bound
layout	1.000	.000	0	.	1.000	1.000	1.000
algorithm	.085	24.692	2	<.001	.522	.529	.500
layout * algorithm	.176	17.356	2	<.001	.548	.563	.500

Tests the null hypothesis that the error covariance matrix of the orthonormalized transformed dependent variables is proportional to an identity matrix.

a. Design: Intercept  
Within Subjects Design: layout + algorithm + layout \* algorithm

b. May be used to adjust the degrees of freedom for the averaged tests of significance. Corrected tests are displayed in the Tests of Within-Subjects Effects table.

Slika 5.3 Rezultati Mauchly-evog testa sferičnosti unutar SPSS okruženja

Mauchly-ev test rezultira vrijednošću  $W(2) = 0,085$ ,  $p < 0,001$  za varijablu *algoritam* te  $W(2) = 0,176$ ,  $p < 0,001$  za interakciju varijabli *raspored* i *algoritam*. Ovaj

Poglavlje 5. Rezultati i diskusija

rezultat ukazuje da je uvjet sferičnosti prekršen te da je potrebno koristiti određene korekcije poput Greenhouse–Geisser. Rezultati Two-way Anova RM testa prikazani su na slici 5.4.

**Tests of Within-Subjects Effects**

Measure: MEASURE\_1

Source		Type III Sum of Squares	df	Mean Square	F	Sig.
layout	Sphericity Assumed	531.128	1	531.128	10.989	.007
	Greenhouse-Geisser	531.128	1.000	531.128	10.989	.007
	Huynh-Feldt	531.128	1.000	531.128	10.989	.007
	Lower-bound	531.128	1.000	531.128	10.989	.007
Error(layout)	Sphericity Assumed	531.643	11	48.331		
	Greenhouse-Geisser	531.643	11.000	48.331		
	Huynh-Feldt	531.643	11.000	48.331		
	Lower-bound	531.643	11.000	48.331		
algorithm	Sphericity Assumed	1068.622	2	534.311	37.125	<.001
	Greenhouse-Geisser	1068.622	1.044	1023.393	37.125	<.001
	Huynh-Feldt	1068.622	1.058	1010.318	37.125	<.001
	Lower-bound	1068.622	1.000	1068.622	37.125	<.001
Error(algorithm)	Sphericity Assumed	316.633	22	14.392		
	Greenhouse-Geisser	316.633	11.486	27.566		
	Huynh-Feldt	316.633	11.635	27.214		
	Lower-bound	316.633	11.000	28.785		
layout * algorithm	Sphericity Assumed	203.418	2	101.709	14.208	<.001
	Greenhouse-Geisser	203.418	1.097	185.487	14.208	.002
	Huynh-Feldt	203.418	1.127	180.511	14.208	.002
	Lower-bound	203.418	1.000	203.418	14.208	.003
Error(layout*algorithm)	Sphericity Assumed	157.485	22	7.158		
	Greenhouse-Geisser	157.485	12.063	13.055		
	Huynh-Feldt	157.485	12.396	12.705		
	Lower-bound	157.485	11.000	14.317		

Slika 5.4 Rezultati Two-way Anova Repeated Measures testa za TER metriku, unutar SPSS okruženja

Rezultat Two-way Anova testa za varijablu *raspored (layout)* iznosi  $F(1, 11) = 10.989$ ,  $p = 0,007 < 0,05$ . Taj rezultat ukazuje da postoji značajan efekt dizajna tipkovnice (rasporeda) na metriku TER. U ovom specifičnom slučaju, post-hoc analiza nije potrebna jer postoje svega dvije razine dotične nezavisne varijable, pa se odmah može generalizirati da je, sa aspekta pogrešaka pri unosu, raspored *Blank Keys* (MEAN = 7,049, STD = 2,197) učinkovitije rješenje od *Invisible keys* (MEAN = 12,481, STD = 1,811). Za varijablu *algoritam (algorithm)*, rezultat testa iznosi

Poglavlje 5. Rezultati i diskusija

$F(1,044, 11,486) = 37,125, p = <0,001$ . Taj rezultat ukazuje da postoji značajan efekt korištenog algoritma na metriku TER. Za razliku od varijable *raspored*, u ovome slučaju nije moguće odrediti gdje točno postoji značajna razlika, pa je potrebno izvršiti post-hoc analizu. Za post-hoc analizu varijable *algoritam*, izvele su se međusobne usporedbe (engl. pairwise comparison) s Bonferroni korekcijom. Na slici 5.5 prikazani su rezultati međusobnih usporedbi za varijablu *algoritam* po TER metrici. Bitno je napomenuti da se unutar SPPS okruženja prilikom korištenja Bonferroni metode korekcija p vrijednosti vrši automatski pa nije potrebna manualna promjena granice od 0,05.

**Pairwise Comparisons**

Measure: MEASURE\_1

(I) algorithm	(J) algorithm	Mean Difference (I-J)	Std. Error	Sig. <sup>b</sup>	95% Confidence Interval for Difference <sup>b</sup>	
					Lower Bound	Upper Bound
1	2	8.950*	1.523	<,001	4.656	13.245
	3	7.065*	.925	<,001	4.457	9.674
2	1	-8.950*	1.523	<,001	-13.245	-4.656
	3	-1.885*	.650	.043	-3.719	-.051
3	1	-7.065*	.925	<,001	-9.674	-4.457
	2	1.885*	.650	.043	.051	3.719

Based on estimated marginal means

\*. The mean difference is significant at the .05 level.

b. Adjustment for multiple comparisons: Bonferroni.

Slika 5.5 Rezultati međusobne usporedbe s Bonferroni korekcijom varijable *algoritam* (1 - Referentni, 2 - *Physical distance*, 3 - *String Distance*) po TER metrici

Rezultati međusobne usporedbe između referentnog (*Raw*) i *Physical distance* algoritma pokazuju da je za količinu grešaka *Physical distance* (MEAN = 6,153, STD = 1,4) algoritam učinkovitije rješenje od referentnog (MEAN = 15,103, STD = 2,461) -  $p < 0,001$ . Usporedba referentnog i *String distance* algoritma također ukazuje da *String distance* (MEAN = 8,038, STD = 1,823) algoritam producira značajno manju pogrešku od referentnog -  $p < 0,001$ . Za usporedbu *String distance* i *Physical distance* algoritma rezultat  $p = 0,043 < 0,05$  ukazuje da *String distance* algoritam proizvodi značajno veću pogrešku od algoritma *Physical distance*. Iz svih dobivenih rezultata, može se zaključiti da je *Physical distance* algoritam najefikasniji



## Poglavlje 5. Rezultati i diskusija

u smanjenju broja pogrešaka, dok je *String Distance* algoritam manje efikasan, ali i dalje učinkovitiji od referentnog algoritma.

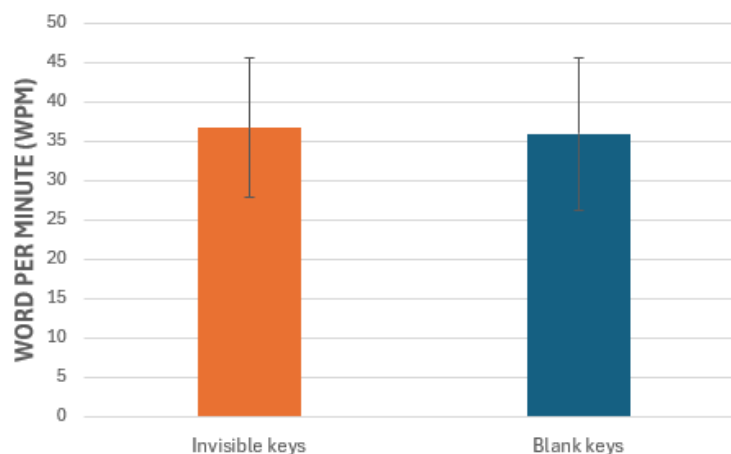
Prilikom izrade tipkovnice, očekivano je da će referentni algoritam ostvariti najgore rezultate jer ne vrši nikakvu korekciju nad korisnikovim unosom, za razliku od ostala dva algoritma, te ovi rezultati to potvrđuju. Najzanimljiviji rezultat unutar ovoga testa jest signifikantna razlika između *Physical distance* i *String distance* algoritma jer je njihova usporedba jedan od glavnih ciljeva ovoga rada. Veću učinkovitost *Physical distance* algoritma u odnosu na *String distance* algoritam moguće je objasniti ako se uzmu u obzir različiti načini na koja ta dva algoritma funkcioniraju.

*Physical distance* algoritam tijekom predikcije koristi točnu udaljenost dodira i svih tipki na tipkovnici, dok *String distance* algoritam uzima u obzir samo najbližu tipku. Kao što je već spomenuto u poglavlju 2, statistički pristup predikciji na virtualnim tipkovnicama se, kroz različite modele, pokazao učinkovitijim od determinističkog. Stoga, nije iznenađujuće da *Physical distance* algoritam pruža bolje rezultate kada njegova implementacija sadrži više statističkih metoda u usporedbi s *String distance* algoritmom. Klasični *string distance* algoritam koristi determinističke metode, te iako je njegova implementacija unutar ovoga rada drukčija od klasične, i dalje sadrži određene determinističke elemente. Najbolji primjer ove razlike je u slučaju kada korisnik unese pogrešno slovo pritiskom na sam rub tipke, moguće uzrokovano sindromom debelog prsta (engl. fat finger). U takvome slučaju *Physical distance* algoritam vrlo lako može odrediti točno slovo jer su vrijednosti pogrešnog i točnog slova, tj. tipke, veoma bliske. *String distance* algoritam, u drugu ruku, svaki dodir računa kao da je pritisnuto samo središte tipke te nije u mogućnosti lako odrediti koja je okolna tipka ispravna, jer sve imaju jednaku cijenu. Unutar korištenja nevidljive tipkovnice, ovakve greške su vrlo česte jer korisnici nisu dovoljno precizni da konzistentno pritisnu ispravnu tipku, ali su dovoljno precizni da pritisak bude blizu ispravne tipke. Iako, korisno je napomenuti da *String distance* algoritam može funkcionirati i bez točne detekcije dodira pa ga je moguće koristiti u raznim drugim tipkovnicama dok *Physical distance* algoritam ovisi o detekciji dodira pa je limitiran na virtualne tipkovnice.

Nakon formalnih testova za metriku pogreške, potrebno je izvršiti formalne testove za brzinu unosa nad varijablom *raspored*, koristeći mjerenu metriku WPM. Varijablu *algoritam* nije moguće usporediti po brzini unosa jer se tijekom korištenja

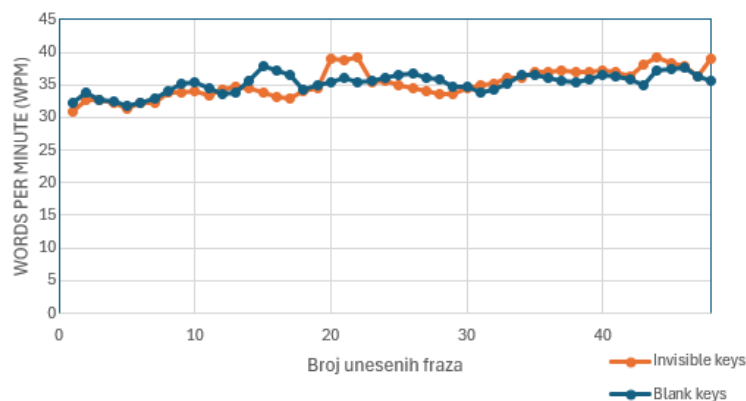
## Poglavlje 5. Rezultati i diskusija

zadanog rasporeda isti unos koristio unutar sva tri algoritma, a WPM metrika ovisi isključivo o brzini. WPM metrika je, isto kao i TER metrika, izračunata iz zadnjih 30 od 50 fraza. Za *Invisible keys* raspored prosječna vrijednost WPM metrike iznosi  $36,63 \pm 8,85$ , a za *Blank keys* raspored  $35,92 \pm 9,65$  što je prikazano na slici 5.6. Na slici 5.7 prikazana je promjena WPM metrike u odnosu na broj upisanih fraza.



Slika 5.6 WPM metrika za oba rasporeda tipkovnice

Kao što je i slučaj za TER metriku, s povećanjem broja upisanih fraza, sudionici



Slika 5.7 Promjena WPM metrike u odnosu na broj upisanih fraza

postaju sve bolji u korištenju zadanog rasporeda te se WPM metrika uskladno po-

Poglavlje 5. Rezultati i diskusija

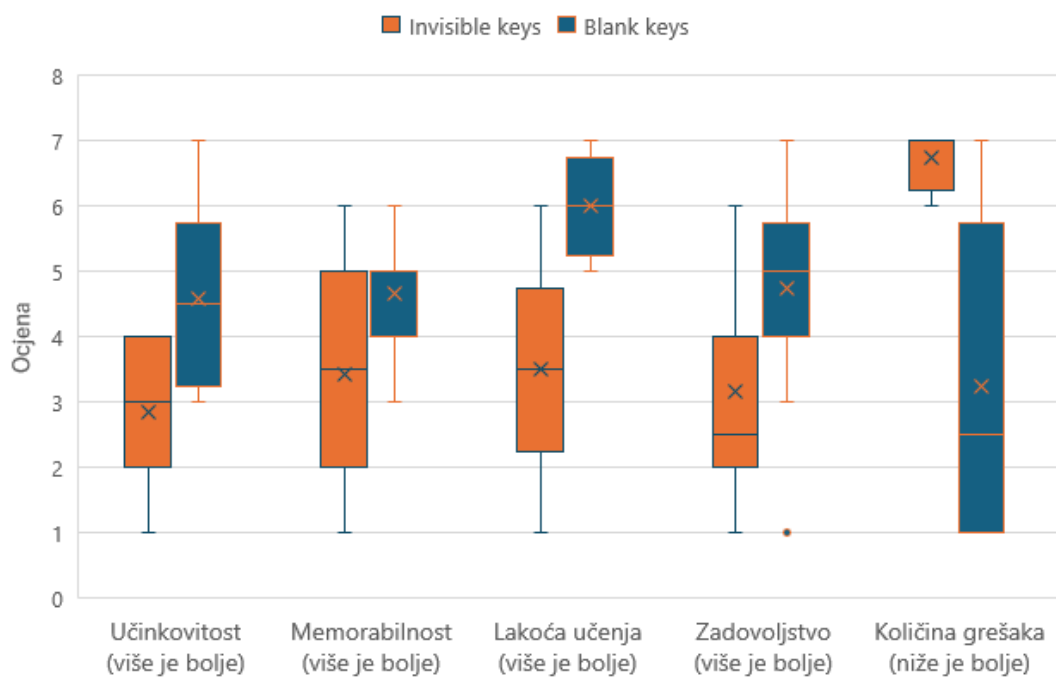
većava. Iz slične promjene u vrijednosti WPM metrike za oba rasporeda tipkovnice, može se pretpostaviti da bi veća količina treninga rezultirala sličnim poboljšanjima za oba rasporeda. Iz ovih rezultata nije moguće zaključiti ako postoji signifikantna razlika između rasporeda tipkovnica, pa je potrebno izvesti formalni test. Za formalnu usporedbu je korišten Paired Samples t-test čiji je rezultat prikazan na slici 5.8. Rezultat Paired Samples t-testa iznosi  $t(11) = 0,785$ ,  $p = 0,449 > 0,05$ . Ovaj

Paired Samples Test		Pair 1
		Invisible keys WPM - Blank keys WPM
Paired Differences	Mean	.70988888889
	Std. Deviation	3.1333187696
	Std. Error Mean	.90451121754
	95% Confidence Interval of the Difference	Lower -1.280926878 Upper 2.7007046558
t		.785
df		11
Significance	One-Sided p	.225
	Two-Sided p	.449

Slika 5.8 Rezultat Paired Samples t-testa za varijablu *raspored*

rezultat ukazuju da ne postoji značajan efekt korištenog rasporeda tipkovnice na metriku WPM. Uzimajući sve dobivene rezultate u obzir, moguće je zaključiti da je po mjerjenim metrikama performansi (TER i WPM), *Blank keys* raspored učinkovitiji od *Invisible keys* rasporeda jer se pokazao boljim pri smanjenju broja pogrešaka bez signifikantne razlike brzine unosa. Osim mjenjenih metrika performansi, svaki sudionik eksperimenta je također ispunio upitnik koji se sastojao od ocjenjivanja atributa upotrebljivosti i NASA TLX-a za oba rasporeda tipkovnice. Atributi upotrebljivosti su se ocjenjivali na Likert skali od 7 razina te su rezultati upitnika vidljivi na slici 5.9. Za usporedbu vrednovanja atributa upotrebljivosti korišten je Wilcoxon signed-rank test čiji su rezultati prikazani na slici 5.10. Rezultati testa pokazuju signifikantnu razliku u korist *Blank keys* rasporeda za attribute: učinkovitost ( $Z = -2,537$ ,  $p = 0,011 < 0,05$ ), lakoća učenja ( $Z = -2,816$ ,  $p = 0,005 < 0,05$ ), zadovoljstvo ( $Z = -2,703$ ,  $p = 0,007 < 0,05$ ) i količina pogrešaka ( $Z = -2,830$ ,  $p = 0,005 < 0,05$ ). Jedini atribut gdje ne postoji signifikantna razlika jest atribut memorabilnosti ( $Z = -1,833$ ,

Poglavlje 5. Rezultati i diskusija



Slika 5.9 Rezultati vrednovanja atributa upotrebljivosti

**Test Statistics<sup>a</sup>**

	Z	Asymp. Sig. (2-tailed)
EfficiencyBlank - EfficiencyInvis	-2.537 <sup>b</sup>	.011
MemorabilityBlank - MemorabilityInvis	-1.833 <sup>b</sup>	.067
LearnabilityBlank - LearnabilityInvis	-2.816 <sup>b</sup>	.005
SatisfactionBlank - SatisfactionInvis	-2.703 <sup>b</sup>	.007
ErrorRateBlank - ErrorRateInvis	-2.830 <sup>c</sup>	.005

a. Wilcoxon Signed Ranks Test

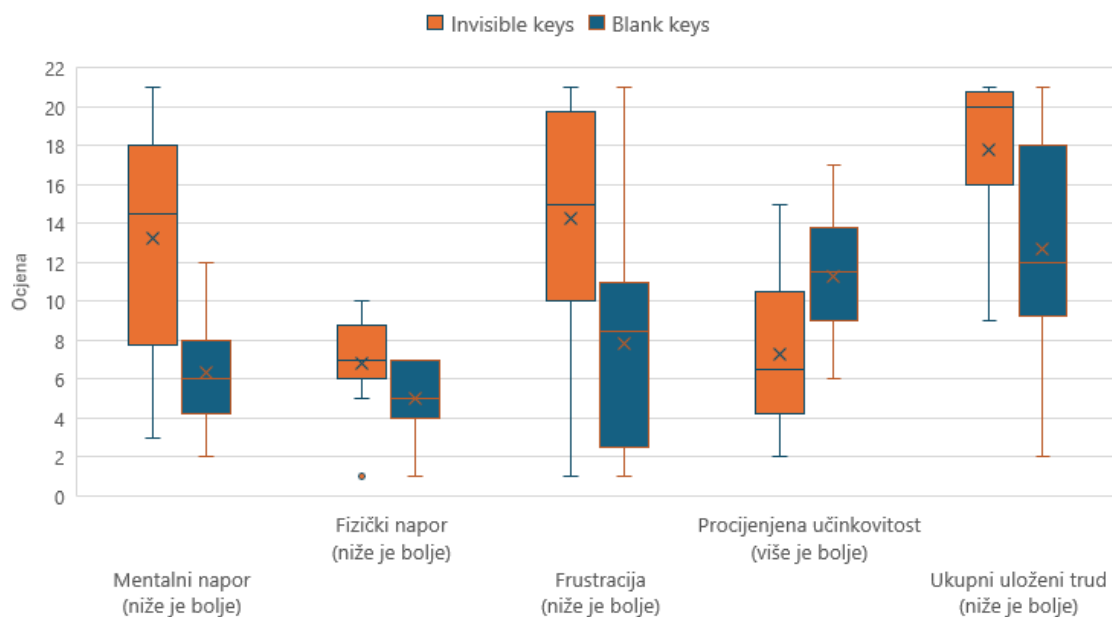
b. Based on negative ranks.

c. Based on positive ranks.

Slika 5.10 Rezultati Wilcoxon signed-rank testa za attribute upotrebljivosti

## Poglavlje 5. Rezultati i diskusija

$p = 0,067 > 0,05$ ). Signifikantna razlika u atributima efikasnosti i količini pogrešaka je usklađena sa rezultatima testova TER metrike, gdje postoji signifikantna razlika u korist *Blank keys* rasporeda. Signifikantna razlika atributa lakoće učenja i zadovoljstva vjerojatno proizlazi iz sličnosti *Blank keys* rasporeda i standardne virtualne tipkovnice. Korisnicima je „naravno, lakše naučiti raspored s kojim su više upoznati, te se ujedno osjećaju prirodnije tijekom korištenja takvog rasporeda tipkovnice. Atribut memorabilnosti je jedini atribut gdje ne postoji signifikantna razlika, što može proizlaziti iz činjenice da korisnici nisu imali potrebu prisjećati se rasporeda tipkovnice tijekom eksperimenta ili zbog općenite jednostavnosti rasporeda. Osim atributa upotrebljivosti, sudionici su također popunili već spomenuti NASA TLX čiji su rezultati prikazani na slici 5.11. NASA TLX faktori ocjenjuju se na skali od 21 razine,



Slika 5.11 Rezultati NASA TLX upitnika

te je također korišten isti Wilcoxon signed-rank test čiji su rezultati prikazani na slici 5.12. Rezultati testa pokazuju signifikantnu razliku u korist *Blank keys* rasporeda za faktore: mentalni napor ( $Z = -2,705$ ,  $p = 0,007 < 0,05$ ), fizički napor ( $Z = -2,232$ ,  $p = 0,026 < 0,05$ ), frustraciju ( $Z = -2,67$ ,  $p = 0,008 < 0,05$ ) i ukupni uloženi trud ( $Z = -2,314$ ,  $p = 0,021 < 0,05$ ). Jedini faktor gdje ne postoji signifikantna razlika

Poglavlje 5. Rezultati i diskusija

Test Statistics <sup>a</sup>					
	BlankMental - InvisibleMental	BlankPhysical - InvisiblePhysical	BlankFrustration - InvisibleFrustration	BlankPerformance - InvisiblePerformance	BlankEffort - InvisibleEffort
Z	-2.705 <sup>b</sup>	-2.232 <sup>b</sup>	-2.670 <sup>b</sup>	-1.940 <sup>c</sup>	-2.314 <sup>b</sup>
Asymp. Sig. (2-tailed)	.007	.026	.008	.052	.021

a. Wilcoxon Signed Ranks Test

b. Based on positive ranks.

c. Based on negative ranks.

Slika 5.12 Rezultati Wilcoxon signed-rank testa za NASA TLX faktore

jest procijenjena učinkovitost ( $Z = -1,940$ ,  $p = 0,052 > 0,05$ ) iako je pripadajuća p vrijednost od 0,052 vrlo blizu granice 0,05. Signifikantna razlika, u svim faktorima gdje postoji, se može pripisati već spomenutoj sličnosti *Blank keys* rasporeda i standardne virtualne tipkovnice. Jedini faktor za koji je neobična signifikantna razlika jest fizički napor jer je unos s *Invisible keys* rasporedom jednako zahtjevan kao i unos s *Blank keys* rasporedom, s obzirom da je način pisanja fizički identičan. Ova razlika vjerojatno proizlazi iz razlike u mentalnom naporu koji dovodi do povećanog fizičkog napora. Također je moguće da sudionici koriste donekle drukčiji stil pisanja kada su tipke nevidljive što dovodi do neprirodnijeg i manje ergonomskog položaja rezultirajući u povećanom fizičkom naporu.

Faktor procijenjene učinkovitosti je jedini faktor gdje ne postoji signifikantna razlika što je zanimljivo jer je TER metrikom dokazano da objektivna razlika u učinkovitosti postoji. Uz same statističke testove, tijekom razgovora sa sudionicima, većina sudionika je izjavila da im je *Blank keys* raspored puno učinkovitiji te da s njime rade manje grešaka. Ovaj rezultat procijenjene učinkovitosti možda proizlazi iz subjektivne prirode NASA TLX faktora procijenjene učinkovitosti, koji odražava ukupne dojmove sudionika o njihovoj izvedbi, a ne precizne razlike u brzini ili točnosti tipkanja. Subjektivna iskustva sudionika, pod utjecajem čimbenika kao što su mentalni i fizički zahtjevi, napor i frustracija, su možda zasjenili stvarne razlike u izvedbi. Također, sudionici su vjerojatno uspoređivali učinkovitosti zadanih rasporeda s performansama standardne virtualne tipkovnice što je dovelo do manje percepcije razlike između zadanih rasporeda.

## Poglavlje 6

### Zaključak

U ovom radu predstavljena je metoda unosa teksta putem nevidljive virtualne tipkovnice za Android uređaje, nazvana *Muscle Memory*. Predstavljena tipkovnica koristi iskustvo stečeno tijekom zadataka utipkavanja i oslanja se na razvijeno mišićno pamćenje korisnika. Cilj ove tipkovnice bio je istražiti potencijalna rješenja za neke od nedostataka virtualnih tipkovnica, kao što su zauzimanje velikog dijela ekrana i nemogućnost skrivanja unosa. Predstavljena tipkovnica uključuje dva različita rasporeda, prvi od kojih je *Invisible keys* raspored, gdje su sve tipke skrivene osim razmaka i tipke za potvrdu unosa, te je prikazan obris prostora za unos. Drugi raspored, *Blank keys*, sadrži iste vidljive tipke kao i prijašnji, ali uz obris prostora za unos, sadrži i obrise svih tipki.

Virtualne tipkovnice su podložne velikoj količini šuma i pogrešaka što posebno dolazi do izražaja unutar nevidljive tipkovnice. Zbog tog razloga, bilo je potrebno implementirati algoritme predikcije kako bi se povećala preciznost unosa. U ovome radu opisana su 2 algoritma za predviđanje unosa: *Physical distance* i *String distance*, zajedno sa referentnim algoritmom koji ne mijenja korisnikov unos, već se ponaša kao da korisnik upotrebljava običnu virtualnu tipkovnicu. *Physical distance* algoritam zapisuje svaki korisnikov dodir te se uz pomoć Gaussove distribucije izračunava vjerojatnost pritiska svake tipke, dok *String distance* algoritam, koristi akcije zamjene, brisanja i dodavanja slova, sa određenim modifikacijama kako bi se povećala preciznost unutar specifičnih zahtjeva nevidljive tipkovnice. Oba algoritma provode predikciju nad listom od 30,000 najkorištenijih engleskih riječi.

Proveden je eksperiment sa 12 sudionika koji su utipkavali 50 fraza sa svakim

## Poglavlje 6. Zaključak

od dva rasporeda tipkovnice, prilikom čega se mjerila brzina unosa i broj pogrešaka (WPM i TER). Rezultati eksperimenta pokazuju da je sa aspekta smanjenja pogrešaka pri unosu, *Physical distance* algoritam najučinkovitiji, dok referentni algoritam daje najgore rezultate. Rezultati analize nad rasporedima tipkovnice ukazuju da postoji signifikantna razlika u količini pogrešaka u korist *Blank keys* rasporeda. Rasporedi tipkovnice su također uspoređeni po brzini unosa (WPM), te rezultati pokazuju da ne postoji signifikantna razlika između dva rasporeda.

Osim metrika performansi, sudionici eksperimenta su također ispunili upitnik s atributima upotrebljivosti i NASA-TLX. Rezultati upitnika za attribute upotrebljivosti pokazali su signifikantnu razliku u korist *Blank keys* rasporeda za attribute učinkovitosti, lakoće učenja, satisfakcije i količine grešaka. Rezultati NASA-TLX upitnika pokazali su signifikantnu razliku u korist *Blank keys* rasporeda, za faktore mentalnog napora, fizičkog napora, frustracije i ukupnog uloženog truda. Većina rezultata se može pripisati sličnosti *Blank keys* rasporeda i standardne virtualne tipkovnice tj. nije začuđujuće da korisnici preferiraju raspored s kojim su više upoznati. Naravno, *Invisible keys* raspored predstavlja bolje rješenje spomenutim problemima virtualnih tipkovnica, ali rezultati eksperimenta pokazuju da je takvo rješenje suboptimalno te da je preferirani raspored *Blank keys*.

Uz više treninga i dužu upotrebu oba predložena rasporeda, može se očekivati odgovarajuće poboljšanje učinkovitosti unosa teksta. U tom kontekstu, mogu se provesti dodatna istraživanja kako bi se bolje proučio učinak treninga na metrike performansi. Također, daljnja poboljšanja ili razvoj novih algoritama predikcije, poput korištenja neuronskih mreža, bi možda rezultiralo drugačijim ishodom eksperimenta. S obzirom da je u istraživanju korišten samo QWERTZ raspored znakova, također bi bilo zanimljivo istražiti kako alternativni rasporedi znakova ili različiti grupni rasporedi mogu utjecati na učinkovitost unosa teksta. Nevidljivost tipkovnice u ovome kontekstu omogućuje fleksibilnost u implementaciji različitih rasporeda, bez ograničenja koja nameće fizička veličina tipkovnice, otvarajući time mogućnost optimizacije za različite korisničke preferencije i situacije.



## Bibliografija

- [1] P. Taylor, “Number of smartphone subscriptions worldwide from 2016 to 2027.” <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>, 2024, pristupano: 3-08-2024.
- [2] M. Daniyal, S. F. Javaid, A. Hassan, and M. A. B. Khan, “The relationship between cellphone usage on the physical and mental wellbeing of university students: A cross-sectional study,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 15, p. 9352, Jul. 2022. , s Interneta, <http://dx.doi.org/10.3390/ijerph19159352>
- [3] J. Lai, D. Zhang, S. Wang, I. D. Y. Kilic, and L. Zhou, “Thumbstroke: A virtual keyboard in support of sight-free and one-handed text entry on touchscreen mobile devices,” *ACM Transactions on Management Information Systems*, vol. 10, no. 3, p. 1–19, Sep. 2019. , s Interneta, <http://dx.doi.org/10.1145/3343858>
- [4] A. Salkanovic and S. Ljubic, *LuxBoard: Ambient Light Manipulation for Contactless Text Entry on Mobile Devices*. Springer International Publishing, 2022, p. 126–139. , s Interneta, [http://dx.doi.org/10.1007/978-3-031-05409-9\\_10](http://dx.doi.org/10.1007/978-3-031-05409-9_10)
- [5] F. C. Y. Li, R. T. Guy, K. Yatani, and K. 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*, ser. UIST '11. ACM, Oct. 2011. , s Interneta, <http://dx.doi.org/10.1145/2047196.2047257>
- [6] S. Chatty and P. Dewan, *Erratum to: Engineering for Human-Computer Interaction*. Springer US, 2017, p. E1–E1. , s Interneta, [http://dx.doi.org/10.1007/978-0-387-35349-4\\_22](http://dx.doi.org/10.1007/978-0-387-35349-4_22)
- [7] D. O. Hebb, *The Organization of Behavior*. John Wiley & Sons, Inc., 1949.

## Bibliografija

- [8] S. Zhu, T. Luo, X. Bi, and S. Zhai, "Typing on an invisible keyboard," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18. ACM, Apr. 2018. , s Interneta, <http://dx.doi.org/10.1145/3173574.3174013>
- [9] S. Gustafson, D. Bierwirth, and P. Baudisch, "Imaginary interfaces: spatial interaction with empty hands and without visual feedback," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, ser. UIST '10. ACM, Oct. 2010. , s Interneta, <http://dx.doi.org/10.1145/1866029.1866033>
- [10] N. Green, J. Kruger, C. Faldu, and R. St. Amant, "A reduced qwerty keyboard for mobile text entry," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI04. ACM, Apr. 2004. , s Interneta, <http://dx.doi.org/10.1145/985921.986082>
- [11] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, ser. MobiSys'14. ACM, Jun. 2014. , s Interneta, <http://dx.doi.org/10.1145/2594368.2594384>
- [12] X. Bi and S. Zhai, "Bayesian touch: a statistical criterion of target selection with finger touch," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ser. UIST'13. ACM, Oct. 2013. , s Interneta, <http://dx.doi.org/10.1145/2501988.2502058>
- [13] X. Bi, Y. Li, and S. Zhai, "Fitts law: modeling finger touch with fitts' law," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '13. ACM, Apr. 2013. , s Interneta, <http://dx.doi.org/10.1145/2470654.2466180>
- [14] Y. Ma, S. Zhai, I. Ramakrishnan, and X. Bi, "Modeling touch point distribution with rotational dual gaussian model," in *The 34th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '21. ACM, Oct. 2021. , s Interneta, <http://dx.doi.org/10.1145/3472749.3474816>
- [15] A. Franz and T. Brants, "All our n-gram are belong to you," <https://research.google/blog/all-our-n-gram-are-belong-to-you/>, 2006, pristupano: 15-07-2024.
- [16] R. W. Soukoreff and I. S. MacKenzie, "Recent developments in text-entry error rate measurement," in *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI04. ACM, Apr. 2004. , s Interneta, <http://dx.doi.org/10.1145/985921.986081>

*Bibliografija*

- [17] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *CHI '03 extended abstracts on Human factors in computing systems - CHI '03*, ser. CHI '03. ACM Press, 2003, p. 754. , s Interneta, <http://dx.doi.org/10.1145/765891.765971>

## Pojmovnik

**CER** *Corrected Error Rate.* 19

**HCI** *Human Computer Interaction.* 5, 6

**NCER** *Not Corrected Error Rate.* 19

**RM** *Repeated Measures.* 27–29

**SPPS** *Statistical Product and Service Solutions.* 28, 30

**TER** *Total Error Rate.* 19, 23, 24, 26, 27, 29, 30, 32, 33, 35, 36, 38

**TLX** *Task Load Index.* 25, 33, 35, 36, 38

**WPM** *Words Per Minute.* 18, 19, 23, 24, 26, 31–33, 38

# Sažetak

U ovom radu implementirana je metoda unosa teksta za Android uređaje sa zaslonom osjetljivim na dodir, a koja se zasniva na mišićnoj memoriji. Korisničko sučelje navedene metode izvedeno je u dvije verzije dizajna – s potpuno skrivenim tipkama te vidljivim obrisima tipaka. Upravljanje pogreškama izvedeno je s tri različita algoritma, od kojih dva – Physical Distance i String Distance – za razliku od referentnog, koriste predikciju unosa. Organiziran je i proveden HCI eksperiment za vrednovanje učinkovitosti razvijene metode u kojem je sudjelovalo 12 ispitanika. Rezultati su pokazali da je, s aspekta metrike pogreške, dizajn tipkovnice s vidljivim obrisima tipaka značajno bolji. Također je pokazano da Physical Distance algoritam pruža najbolje, a referentni algoritam najlošije rezultate. Podaci dobiveni standardiziranim upitnicima, kojima se ispituje radno opterećenje te atributi upotrebljivosti pri unosu teksta, potvrđuju rezultate eksperimenta.

***Ključne riječi*** — Virtualna tipkovnica, unos teksta, mišićna memorija, interakcija čovjeka i računala, Android

## Abstract

In this thesis, a text entry method based on muscle memory is implemented for Android touchscreen devices. The user interface of the mentioned method is designed in two versions – with completely hidden keys and utilizing visible outlines of the keys. Error management is performed using three different algorithms, two of which – Physical Distance and String Distance – use input prediction as opposed to the reference algorithm. To empirically evaluate the developed method, an HCI experiment was organized and conducted in which 12 subjects participated. The results showed that the keyboard design with visible key outlines is significantly better regarding error metrics. It was also shown that the Physical Distance algorithm provides the best results and the reference algorithm the worst results. The data obtained through standardized questionnaires, which test text entry's workload and usability attributes, confirm the experiment's results.

***Keywords*** — Virtual keyboard, text entry, muscle memory, Human-Computer

*Pojmovnik*

**Interaction, Android**

# Dodatak A

## Izvorni kod

Izvorni kod nalazi se na sljedećem GitHub repozitoriju: <https://github.com/KarloGraf/MuscleMemoryKeyboard>