

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4425

**Alternativna tastatura za zaslone  
osjetljive na dodir temeljena na  
Fittsovom zakonu**

Juraj Šušnjara

Zagreb, lipanj 2016.

*Umjesto ove stranice umetnite izvornik Vašeg rada.  
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*



# CONTENTS

<b>1. Uvod</b>	<b>1</b>
<b>2. Alternativne tipkovnice</b>	<b>3</b>
2.1. Dvorak . . . . .	3
2.2. ATOMIK . . . . .	4
2.3. FITALY . . . . .	4
2.4. Colemak . . . . .	4
<b>3. Oblikovanje alternativne tipkovnice</b>	<b>6</b>
3.1. Fittsov zakon . . . . .	7
3.1.1. Vrednovanje razmještaja tipki . . . . .	8
3.2. Genetski algoritam . . . . .	9
3.2.1. Operator križanja . . . . .	10
3.2.2. Operator mutacije . . . . .	11
3.2.3. Operator selekcije . . . . .	11
3.2.4. Operator reprodukcije . . . . .	12
3.2.5. Primjena genetskog algoritma . . . . .	13
3.3. Rješenja genetskog algoritma . . . . .	15
3.3.1. Implementacija tipkovnica za Android operacijski sustav . . . . .	16
<b>4. Provedba eksperimenta i obrada rezultata</b>	<b>18</b>
4.1. Oblikovanje eksperimenta . . . . .	19
4.2. Eksperimentalno vrednovanje tipkovnica . . . . .	20
4.2.1. Prikaz i interpretacija rezultata . . . . .	21
<b>5. Zaključak</b>	<b>26</b>

# 1. Uvod

Tipkovnice i pisaće tehnologije postoje i razvijaju se već dosta vremena. Prvi pisaći strojevi osmišljeni su i patentirani još 1700-tih, dok su u proizvodnju krenuli 1870-tih godina. Na prvim se takvim strojevima nije čak ni mogao vidjeti onaj tekst koji se unosio jer se papir nalazio unutar njega sve do završetka stranice. Od tada su se dogodile mnoge promjene u dizajnu, načinu unošenja teksta, rasporedu tipki i samoj tehnologiji. Pisaći uređaji su s vremenom postajali sve jednostavniji i lakši za korištenje.

Godine 1867. Cristopher Latham Sholes, uređivač novina iz Wisconsina, patentirao je svoj prvi pisaći stroj koji je razvio s prijateljima Carlos Glidden-om i Saumel W. Soule-om. Taj je stroj, kao i svi njegovi prethodnici, bio mehanički i zbog toga se već napisani znakovi nisu mogli samo tako izbrisati kao što se to može danas na računalima i mobitelima. Ukoliko je unio nešto krivo, korisnik je morao izvaditi papir i započeti iznova. Pisaći stroj je također često znao zaglaviti ako bi se brzo, jedan za drugim, pritisnula dva znaka koja se nalaze jedan pored drugog. Sholes je zbog toga želio osmisliti raspored znakova na tipkovnici koji bi korisniku omogućio da radi manje pogrešaka i brže unosi tekst. Prijedlog je bio da razdvoji najčešće korištene parove slova (*npr. "th" u engleskom jeziku*), tako da ne budu jedno pokraj drugoga, a i da korisnik naizmjenice tipka s lijevom i desnom rukom. Da bi se to ostvarilo bilo je potrebno proučiti bigrame<sup>1</sup> za određeni jezik. Sholes se mučio nekoliko godina kako bi usavršio raspored dok konačno nije došao do onog kakav se i danas koristi, a to je QWERTY.

QWERTY tipkovnica dobila je svoje ime zbog prvih 6 tipki u prvom retku. Izmjena ruku tijekom unošenja teksta se pokazalo kao željeno ponašanje pri oblikovanju tipkovnice. Dok jedna ruka pritisne slovo druga se može pripremiti za pritisnuti iduće, čineći tako cijeli proces bržim. Međutim, kada se niz slova unosi jednom rukom, šanse za pogrešku se povećavaju i ritam se može poremetiti pa se na taj način smanjuje brzina i povećaje broj pogrešno unešenih znakova. Iako je QWERTY oblikovana da se izmjenjuju ruke, mnogo riječi na engleskom jeziku se unosi samo jednom rukom.

---

<sup>1</sup>Frekvencija pojavljivanja parova slova u pojedinom jeziku

Tisuće engleskih riječi se mogu unijeti samo lijevom, a nekoliko stotina samo desnom. Općenito, pri unošenju bilo koje riječi, češće se koristi lijeva ruka. Zbog toga lijevaci imaju prednost pri korištenju ovakve tipkovnice. Postoji više različitih varijanti QWERTY-a s nekim manjim izmjenama, primjer je QWERTZ tipkovnica. QWERTZ je prvi put uvedena u Njemačkoj zbog toga što je slovo Z u njemačkom jeziku puno češće korišteno nego slovo Y. Takva tipkovnica je preuzeta još u nekoliko Europskih zemalja, uključujući Hrvatsku.

Nakon pisaćih strojeva pojavila su se računala, a nakon njih mobiteli i pametni uređaji. Na njima nije više postojao problem zaglavljivanja i nemogućnost ispravljanja pogrešaka kao kod pisaćih strojeva. Nadalje, na pametnim uređajima upotrebljavaju se *meke tipkovnice* (engl. *soft keyboards* ili *onscreen keyboards*). Takve tipkovnice zamjenjuju hardware-ske i koriste se na zaslonima osjetljivim na dodir. One se koriste na drugačiji način od klasičnih tipkovnica na računalima, tekst se najčešće unosi stilusom (engl. *stylus*) te koristeći jedan ili dva prsta. Dok se na računalima i pisaćim strojevima koriste svih 5 prstiju obe ruke. To za sobom povlači sljedeće pitanje. Zašto još i dalje koristimo QWERTY tipkovnicu iako se način tipkanja mijenja ovisno o tehnologiji, a i ta je tipkovnica pokazala mnogo mana s vremenom. Nadalje, QWERTY je oblikovana za engleski jezik, zašto onda govornici drugi jezika koriste tu tipkovnicu. Odgovor na ta pitanja je jednostavan, ljudi su navikli na nju i teško je preći na neku drugu. Problem koji se iz ovoga javlja jest oblikovanje alternativne tipkovnice koja bi se pokazala učinkovitijom za pojedini jezik.

## 2. Alternativne tipkovnice

Kao što je u prethodnom poglavlju navedeno, QWERTY tipkovnica pokazuje mnogo nedostataka. Zbog toga je oblikovano još mnogo alternativnih tipkovnica koje su se eksperimentalno pokazale učinkovitije. Takve tipkovnice su napravljene s namjerom da zamjene QWERTY i da postanu standardne ali taj pokušaj je ostao bezuspješan. Neke od takvih tipkovnica su navedene u nastavku.

### 2.1. Dvorak

Dvorak tipkovnica (prikazana na slici 2.1) patentirana je 1936. godine, a osmislili su je August Dvorak i William Dealey. Njena svrha je bila da zamjeni QWERTY te da značajno ubrza brzinu unošenja teksta. Dvorak je smatrao da QWERTY ima mnogo mana:

- mnogo često upotrebljivanih kombinacija slova zahtjeva čudne i neprikladne položaje prstiju
- mnogo često upotrebljivanih kombinacija slova zahtjeva da se prstom preskače srednji red (engl. *home row*)
- mnogo često upotrebljivanih kombinacija slova se unose jednom rukom
- većina unošenja teksta se obavlja lijevom rukom iako su većina ljudi dešnjaci
- oko 16% tipkanja se radi u donjem retku, 52% u gornjem retku i samo 32% u glavnom odnosno srednjem redu

Dvorak je proučavao učestalost pojavljivanja slova i fiziologiju ljudske šake kako bi rješio sve probleme QWERTY tipkovnice. Težio je prema tome da unošenje teksta bude ritmično, da se izmjenjuju ruke što više moguće te da se smanji broj pogrešaka. Svi samoglasnici se nalaze u srednjem retku, najčešće korišteni simboli su na lijevoj strani dok su najčešće korišteni suglasnici na desnoj strani tipkovnice. Takav razmještaj rezultira sljedećom statistikom: 70% tipkanja obavlja se u srednjem retku, 22% u gornjem i 8% u donjem. Dvorak tipkovnica namijenjena je engleskom jeziku. Drugi

jezici imaju različite frekvencije pojavljivanja slova pa čak i različita slova, zbog toga se dovodi u pitanje prednost njena prednost kada se ne koristi za engleski jezik. Bitno je naglasiti da je Dvorak definirao principe koji se mogu primjeniti za oblikovanje tipkovnica i za ostale jezike.

## 2.2. ATOMIK

Ova raspodijela napravljena je za korištenje stilus-a (engl. stylus) na zaslonima osjetljivim na dodir. Razvio ga je IBM koristeći Metropolis algoritam kako bi matematički minimizirali pokrete potrebne za napisati riječi na engleskom jeziku. Tipkovnica je prikazana na slici 2.2.

## 2.3. FITALY

FITALY tipkovnica namijenjena je za unošenje teksta jednim prstom ili stilusom. Pretpostavlja se da je početan položaj stilusa ili prsta na sredini tipkovnice. Unošenje teksta na ovaj način rezultira malim pomacima do drugih tipki. Razmještaj tipki je napravljen uzevši u obzir frekvenciju slova u engleskom jeziku. Na slici 2.3 prikazana je FITALY tipkovnica te je uz svako slovo navedena učestalost korištenja u 10000 slova. Tipka *space* je najveća jer se najviše koristi i jer je tada lakše s određenog slova doći do te tipke (najveća udaljenost od bilo kojeg slova do *space-a* je 2). Najčešće korišteni parovi slova nalaze se u sredini i jedan blizu drugoga. Koristeći ovakav razmještaj znači da je više od pola tipki udaljeno za 1 od centralnih tipki *N* i *E*, a 84% svih tipki je udaljeno za 2. Najveća udaljenost dvije tipke je 5.

## 2.4. Colemak

Colemak tipkovnica (slika 2.4) nudi male promjene u odnosu na QWERTY uz povećanu brzinu unošenja teksta. Poprilično je slična QWERTY da bi je korisnici lakše prihvatili. Razvio je Shai Coleman, a cilj mu je bio češće korištena slova postaviti ispod najjačih prstiju kako bi korisnicima olakšao korištenje. Na taj način sprječava pojavu *RSI sindroma*<sup>1</sup>. Colemak tipkovnica ima QWERTY bazu, a od nje se razlikuje u 17

---

<sup>1</sup>engl. Repetitive strain injury. Skupni izraz za grupu bolesti koje pogađaju mišiće, ligamente i živce u oštećenom području tijela, a nastaju uslijed dugotrajnog ponavljanja određenih automatiziranih radnji u neodgovarajućem položaju.



znakova. Kao i kod Dvorak tipkovnice, težnja je bila da korisnik što manje pomiče prste te da se srednji redak što više koristi. Prste je potrebno pomicati čak 2.2 puta više koristeći QWERTY, a kod Colemak tipkovnice 35 puta više riječi je moguće napisati u srednjem retku.

~	!	@	#	\$	%	^	&	*	(	)	{	}	← Backspace
Tab	1	2	3	4	5	6	7	8	9	0	[	]	→
Caps Lock	A	O	E	U	I	D	H	T	N	S	-	=	Enter
Shift	Q	J	K	X	B	M	W	V	Z	Shift	↵		
Ctrl	Win Key	Alt								Alt Gr	Win Key	Menu	Ctrl

Figure 2.1: Dvorak

{	}	B	K	D	G	.	,	?	!	7	8	9	
~	→	C	A	N	I	M	Q	←	4	5	6	\$	
(	)	F	L	E	S	Y	X	←	1	2	3		
\	/	J	H	T	O	P	V	!	+	-	0	*	/ =
caps	↵	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷	↶	↷
g	←	→	↑	↓	←	→	↑	↓	←	→	↑	↓	↶

Figure 2.2: ATOMIK

Z 20	V 77	C 230	H 415	W 138	K 49
F 176	I 551	T 701	A 615	L 319	Y 133
(space) 1741	N 550	E 976	(space) 1741		
G 147	D 305	O 590	R 497	S 497	B 110
Q 20	J 20	U 210	M 187	P 150	X 20

Figure 2.3: FITALY

~	!	@	#	\$	%	^	&	*	(	)	-	=	← Backspace
Tab	Q	W	F	P	G	J	L	U	Y	:	{	}	
← Backspace	A	R	S	T	D	H	N	E	I	O	"	'	Enter
Shift	Z	X	C	V	B	K	M	<	>	?	/	↵	Shift
Ctrl	Win Key	Alt									Alt Gr	Win Key	Menu Ctrl

Figure 2.4: Colemak

### 3. Oblikovanje alternativne tipkovnice

Ovo poglavlje obuhvaća opis samog problema te pristup njegovog rješavanja. Cilj ovog rada jest osmisliti alternativnu tipkovnicu za zaslone osjetljive na dodir koristeći Fittsov zakon. Jednostavnije rečeno, cilj je dizajnirati tipkovnicu s određenim rasporedom tipki koja bi korisniku omogućila brže unošenje teksta u odnosu na QWERTY tipkovnicu. Algoritam koji se koristi temelji se na zasadama Fittsovog zakona (detaljnije opisan u poglavlju 3.1). Taj zakon predviđa vrijeme potrebno da korisnik reagira i pomakne *nešto* s jedne lokacije na drugu. To *nešto* može biti npr. kursor miša, ruka, noga ili konkretno što se tiče ovog rada: prst. To vrijeme je funkcija udaljenosti do određene lokacije i širine te lokacije:  $f(D, W)$ . U okviru pronalaženja optimalnog rasporeda znakova na tipkovnici, udaljenost predstavlja međusobnu udaljenost dvije tipke na tipkovnici koje će se uzastopno stisnuti, a širina jest širina same tipke. Kao što se već da naslutiti, cilj je minimizirati funkciju vremena što istovremeno znači i minimizirati vrijeme potrebno da se unese određeni tekst. Uzmimo kao primjer riječ PAS i da korisnik tekst unosi prstom jedne ruke. Kako bi se izračunalo vrijeme koje je potrebno da korisnik to unese treba izračunati udaljenost od P do A  $d(P, A)$  i širinu od A  $w(A)$ , pa udaljenost od A do S  $d(A, S)$  i širinu od S  $w(S)$ . Pretpostavlja se da je početni položaj korisnikova prsta iznad slova P, kada njega stisne potrebno je preći udaljenost  $d(P, A)$  da bi se stisnula tipka A širine  $w(A)$  pa udaljenost  $d(A, S)$  da bi se stisnula tipka S širine  $w(S)$ . Iz ovoga se vidi kako će se pomoću Fittsovog zakona dobiti tipkovnica koja će imati slova koja se često uzastopno koriste, jedan blizu drugoga. Zbog toga je potrebno analizirati statistiku bigrama za ciljani jezik tipkovnice.

Kako bi se pronašao raspored tipki koji će prema Fittsovom zakonu imati najkraće vrijeme unosa teksta bilo bi najjednostavnije isprobati svaku moguću i pronaći onu s najmanjim vremenom. Ali to nije izvedivo, ako se uzme u obzir hrvatska abeceda koja ima 30 slova, potrebno je ispitati  $30! = 2.65 * 10^{32}$  mogućih različitih razmještaja (starost svemira je  $10^{10}$  sekundi). Ovaj problem predstavlja tipičan problem optimizacije, pitanje je kako pronaći minimum funkcije zadane Fittsovim zakonom uzevši u obzir sve moguće razmještaje slova. Postoje razne metode za riješiti taj prob-

lem. U ovom radu se koristio genetski algoritam. Genetski algoritam i način na koji je iskorišten detaljno je opisan u poglavlju 3.2.

### 3.1. Fittsov zakon

Fittsov zakon je opisni model ljudskog pokreta koji se prvenstveno koristi u *HCI-u*<sup>1</sup> i *ergonomiji*<sup>2</sup>. Kao što je već navedeno, Fittsov zakon predviđa vrijeme brzog pokreta do određene lokacije, to je funkcija udaljenosti do odredišta i širine tog odredišta. Ovaj zakon modelira čin pokazivanja (engl. *act of pointing*), to može biti fizičko dodirivanje nekog objekta, primjerice rukom i prstom, a može biti i virtualno kao što je strelicom miša na računalu. Fittsov zakon se pokazao učinkovit u mnogo različitih slučajeva, koristeći razne dijelove tijela (ruke, noge, usne, kretanje oka...), u raznim okolinama (čak i pod vodom) i u raznim društvenim skupinama (mladi, stariji, ljudi s posebnim potrebama...).

Ideja Fittsovog zakona prikazana je na slici 3.1 gdje se kao udaljenost do odredišta gleda udaljenost od dvije istaknute lokacije, a širina odredišta je širina "Take the Tour" gumba. Na slici 3.2 vidi se razlika između klizne trake (engl. *scroll-bar*) kod OSX-a i Windows-a. Kod Windows-a su strelice za pomicanje klizne trake krajnje lijevo i krajnje desno, što više prati mentalni model osobe kojoj je to intuitivnije. S druge strane kod OSX-a su obe strelice jednu uz drugu kako bi se smanjilo vrijeme pomicanja strelice miša s jedne na drugu, što je konkretno ono čega se Fittsov zakon dotiče.

Paul Fitts je 1954. godine predložio metriku pomoću koje se vrednuje težina zadatka pokazivanja (brzog pomicanja na određenu lokaciju). Ta metrika se temelji na informacijskoj analogiji, gdje je  $D$  udaljenost do odredišta i gleda se na njega kao na signal, a  $W$  je širina ili tolerancija koja se ponaša kao šum.  $ID$  predstavlja indeks težine (engl. *index of difficulty*) i mjeri se u bitovima. Formula je dana izrazom 3.1.

$$ID = \log_2\left(\frac{2D}{W}\right) \quad (3.1)$$

Fitts je također predložio indeks izvedbe (engl. *index of performance*) kao mjerilo ljudske izvedbe. Ova metrika kombinira  $ID$  (*index of difficulty*) s vremenom pomicanja  $MT$  (*movement time*) na određenu lokaciju. Prosječna stopa informacije koja je dobivena serijom pokreta jest prosječna informacija po pomicanju podijeljena s vre-

---

<sup>1</sup>Interakcija čovjeka i računala (engl. *Human Computer Interaction*)

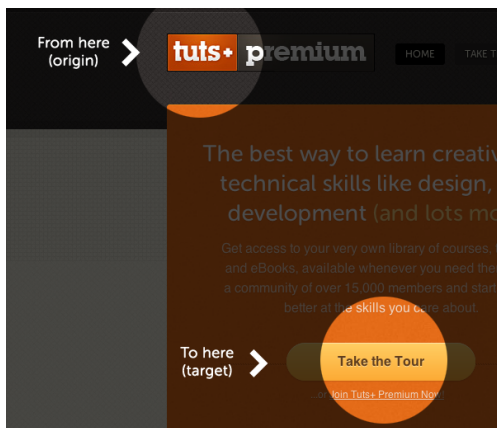
<sup>2</sup>Znanstvena disciplina koja istražuje ljudski organizam i ponašanje, te pruža podatke o prilagođenosti predmeta s kojima čovjek dolazi u kontakt.

menom pomicanja. Formula je dana izrazom 3.2.

$$IP = ID/MT \quad (3.2)$$

$MT$  je definiran izrazom 3.3. Parametri  $a$  i  $b$  koji najviše odgovaraju tom izrazu pronalaze se linearnom regresijom.

$$MT = a + b * ID = a + b * \log_2(2D/W) \quad (3.3)$$



**Figure 3.1:** Uporaba Fittsovog zakona



**Figure 3.2:** Uporaba Fittsovog zakona

### 3.1.1. Vrednovanje razmještaja tipki

Za potrebe ovog rada korištena je modifikacija Fittsovog zakona kako bi se evaluirala pojedina tipkovnica. Formula je dana izrazom 3.4

$$t = \sum_{i=1}^N \sum_{j=1}^N \frac{P_{i,j}}{IP} \log_2 \left( \frac{D_{i,j}}{W_{i,j}} + 1 \right) \quad (3.4)$$

gdje je:

- $t$ : prosječna vremenska cijena (engl. *time cost*)
- $P$ : matrica vjerojatnosti prelaska s jednog slova na drugu (bigrami)
- $D$ : matrica međusobne udaljenosti dva slova na tipkovnici
- $W$ : širina tipke na kojoj se nalazi određeno slovo
- $IP$ : Fittsov indeks izvedbe mjeren u bitovima po sekundi [*bit/s*]
- $N$ : broj slova u ciljanoj abecedi

Ovaj model se može pojednostavniti ako postavimo  $W = IP = 1$  što neće utjecati na konačne rezultate.

U nastavku je opisano što izraz 3.4 zapravo znači. Kao primjer uzeta je tipkovnica prikazana na slici 3.3. Prvo što treba je matrica vjerojatnosti  $P$ . Ona je dimenzija  $N \times N$  i izgleda različito za svaki jezik (za engleski jezik najveću će vrijednost imati polje koje označava bigram  $TH$ ). Nakon toga potrebno je izračunati matricu  $D$ . Udaljenost dva slova na tipkovnici se jednostavno računaju tako da se tipkovnica gleda kao šahovska ploča gdje svako slovo ima svoje koordinate. Primjerice udaljenost slova  $z$  i  $d$  je 1, a udaljenost slova  $c$  i  $h$  je  $\sqrt{2^2 + 1^2} = 2.236$ . Kada se izračunate vrijednosti provuku kroz izraz 3.4 dobit će se vremensku cijenu tipkovnice koja približno iznosi 1.038. Jedinica ove vrijednosti nije bitna, bitno je jedino da postoji kriterij za usporedbu dvije tipkovnice. Što je vremenska cijena tipkovnice manja to je tipkovnica "bolja". Ovaj model preuzet je iz rada *"Optimizing stylus keyboard layouts with a genetic algorithm: customization and internationalization"* - Chad R. Brewbaker.

Ovakva tipkovnica namijenjena je za unošenje teksta koristeći stilus ili jednim prstom. Kako bi se generirala tipkovnicu koja je idealnija za korištenje dva prsta (primjerice palčevima kao što to većina ljudi radi na pametnim uređajima) napravljene su neke izmjene u odnosu na prethodno opisani model. Svaka tipkovnica podijeljena je na dva dijela (lijevi i desni) od kojih je svaki dio namijenjen za lijevu, odnosno desnu ruku. Svaki dio ima svoje središte za koje se pretpostavlja da je lokacija iznad koje korisnik drži prst kada unosi tekst. Vremenska cijena ovakve tipkovnice računa se na malo drugačiji način. Uzmimo za primjer tipkovnicu na slici 3.4. Lijevo i desno središte označeni su crvenim kružićem. Znakovi  $B$  i  $H$  su na istoj strani (lijevoj) pa se pretpostavlja da će ih korisnik oba unijeti koristeći palac lijeve ruke. Njihova udaljenost se tada računa kao i prije te iznosi 2. Znakovi  $R$  i  $O$  nalaze se na suprotnim stranama. Za njihovu udaljenost uzima se zbroj udaljenosti od lijevog središta do znaka  $R$  i udaljenosti od desnog središta do znaka  $O$ . Ovakav način računanja odabran je zbog toga što većina ljudi tako unosi tekst na pametnim uređajima. Kako bi se provjerila valjanost ovog modela napravljen je eksperiment koji je opisan u poglavlju 3.

## 3.2. Genetski algoritam

Genetski ili genetički algoritam (GA) je heuristička metoda optimiranja koja imitira prirodni evolucijski proces. Evolucija je robustan proces pretraživanja prostora rješenja. Živa bića se tijekom evolucije prilagođavaju uvjetima u prirodi, tj. životnoj okolini. Analogija evolucije kao prirodnog procesa i genetskog algoritma kao metode

	x		ć		z		d		č		ž		q	
	w		c		j		e		n		u		đ	
	h		m		ì		a		t		s		š	
	y		b		l		r		o		k		f	
					v		p		g					

Figure 3.3: Primjer tipkovnice

	b		c		h		đ				q		š		y		ž	
	z		ì		m		ć				f		k		t		u	
	r		n		e		č				p		o		a		s	
			l		j		w		x		g		v		d			

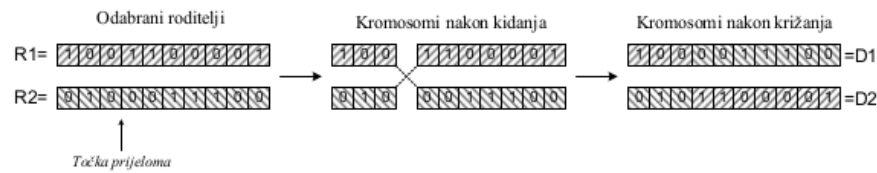
Figure 3.4: Primjer tipkovnice

optimiranja, očituje se u procesu selekcije i genetskim operatorima. Mehanizam odabira nad nekom vrstom živih bića u evolucijskom procesu čine okolina i uvjeti u prirodi. U genetskim algoritmima ključ selekcije je funkcija cilja, koja na odgovarajući način predstavlja problem koji se rješava. Slično kao što su okolina i uvjeti u prirodi ključ selekcije nad nekom vrstom živih bića, tako je i funkcija cilja ključ selekcije nad populacijom rješenja u genetskom algoritmu. Naime, u prirodi jedinka koja je najbolje prilagođena uvjetima i okolini u kojoj živi ima najveću vjerojatnost preživljavanja i parenja, a time i prenošenja svojega genetskog materijala na svoje potomke. Za genetski algoritam jedno rješenje je jedna jedinka. Selekcijom se odabiru dobre jedinke koje se prenose u slijedeću populaciju, a manipulacijom genetskog materijala stvaraju se nove jedinke. Takav ciklus selekcije, reprodukcije i manipulacije genetskim materijalom jedinki ponavlja se sve dok nije zadovoljen uvjet zaustavljanja evolucijskog procesa. Konačan rezultat je populacija jedinki (potencijalnih rješenja). Najbolja jedinka u zadnjoj iteraciji predstavlja rješenje optimiranja. Genetski algoritam izvršava se uporabom operatora križanja, mutacije, selekcije i reprodukcije. Svaki od njih objašnjen je u nastavku.

### 3.2.1. Operator križanja

U procesu križanja (engl. *crossover*) sudjeluju dvije jedinke koje se nazivaju *roditelji*. Dakle, križanje je binarni operator. Križanjem nastaju jedna ili dvije nove jedinke koje se nazivaju *djeca*. Najvažnija karakteristika križanja jest da *djeca* nasljeđuju svojstva svojih roditelja. Ako su roditelji *dobri* (prošli su proces selekcije), tada će najvjerojatnije i *dijete* biti dobro, ako ne i bolje od svojih roditelja. Ako se jedinka predstavi kao niz bitova onda se križanje implementira slično kao što je prikazano na slici 3.5. Bitno je napomenuti da se križanje može izvesti na mnogo različitih načina, ovisno o

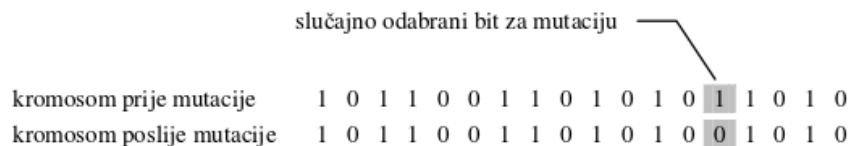
problemu i načinu prikaza jedinke.



**Figure 3.5:** Križanje s jednom točkom prijeloma

### 3.2.2. Operator mutacije

Mutacija ili slučajna promjena jednog ili više gena je unarni operator koji djeluje nad jednom jedinkom. Rezultat mutacije je izmijenjena jedinka. Parametar koji određuje vjerojatnost mutacije  $p_m$  jednog bita je ujedno i parametar algoritma. Ako vjerojatnost mutacije teži k jedinici, tada se algoritam pretvara u algoritam slučajne pretrage prostora rješenja. S druge strane, ako vjerojatnost mutacije teži k nuli, postupak će najvjerojatnije već u početku optimiranja stati u nekom lokalnom optimumu. *Jednosstavna mutacija* svaki bit kromosoma mijenja s jednakom vjerojatnošću  $p_m$ .



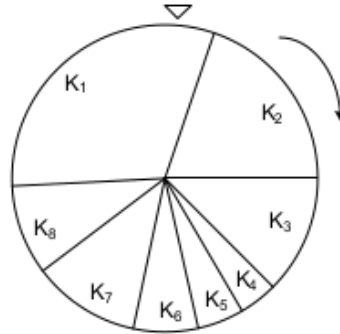
**Figure 3.6:** Jednostavna mutacija

### 3.2.3. Operator selekcije

Jedan od centralnih mehanizama, kako kod većine populacijskih algoritama pa i općenito, jest mehanizam *selekcije*. Kod populacijskih algoritama zadaća operatora selekcije jest osigurati mehanizam koji će češće boljim rješenjima dati priliku da sudjeluju u produkciji novih rješenja, čime će se proces pretraživanja prostora rješenja voditi u područja koja više obećavaju. Postoji više različitih vrsta selekcija.

*Proporcionalna selekcija* (engl. *Roulette-wheel selection*) funkcionira na način da sve jedinke postavimo na kolo, tako da bolja jedinka ima veću površinu na kolu (prikazano na slici 3.7). Potom zavrtnemo kolo i pogledamo na kojoj se jedinki kolo zaustavilo. S obzirom da jedinki pripada to veći dio oboda kola što ima veću dobrotu, ponavljamo li

eksperiment puno puta, bolje jedinke će biti češće birane od lošijih, i ta će vjerojatnost biti upravo proporcionalna dobroti same jedinke.



**Figure 3.7:** Proporcionalna selekcija

Kod *k*-turnirske selekcije se iz populacije posredstvom slučajnog mehanizma odabire *k* jedinki, i potom uzima najbolju. Ukoliko trebamo *n* roditelja,, naprosto ćemo *n* puta ponoviti turnirsku selekciju.

### 3.2.4. Operator reprodukcije

Reprodukcijom se jednostavno jedinka prenosi iz jedne generacije u drugu. Ta jedinka se odabere operatorom selekcije.

---

#### **Algorithm 1** Genetski algoritam - pseudokod

---

P = stvoriPocetnuPopulaciju(VELICINA)

evaluiraj(P)

**repeat**

    novaPopulacija P' =  $\emptyset$

**while** velicina(P') < VELICINA **do**

        odaberi R1 i R2 iz P

        {D1, D2} = krizaj(R1, R2)

        mutiraj D1, mutiraj D2konfiguracije

        dodaj D1 i D2 u P'

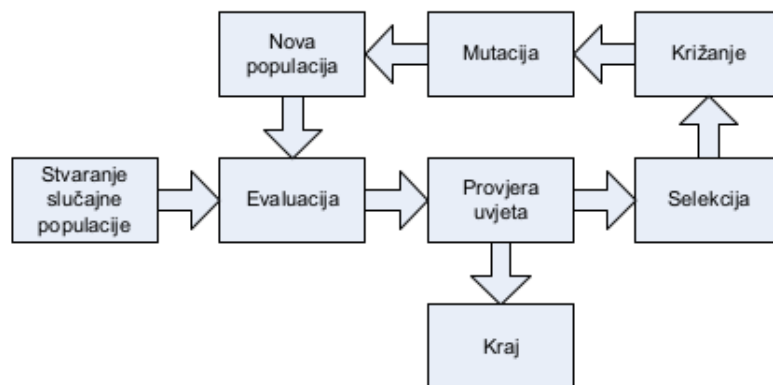
**end while**

    P = P'

**until** uvjetZadovoljen

---

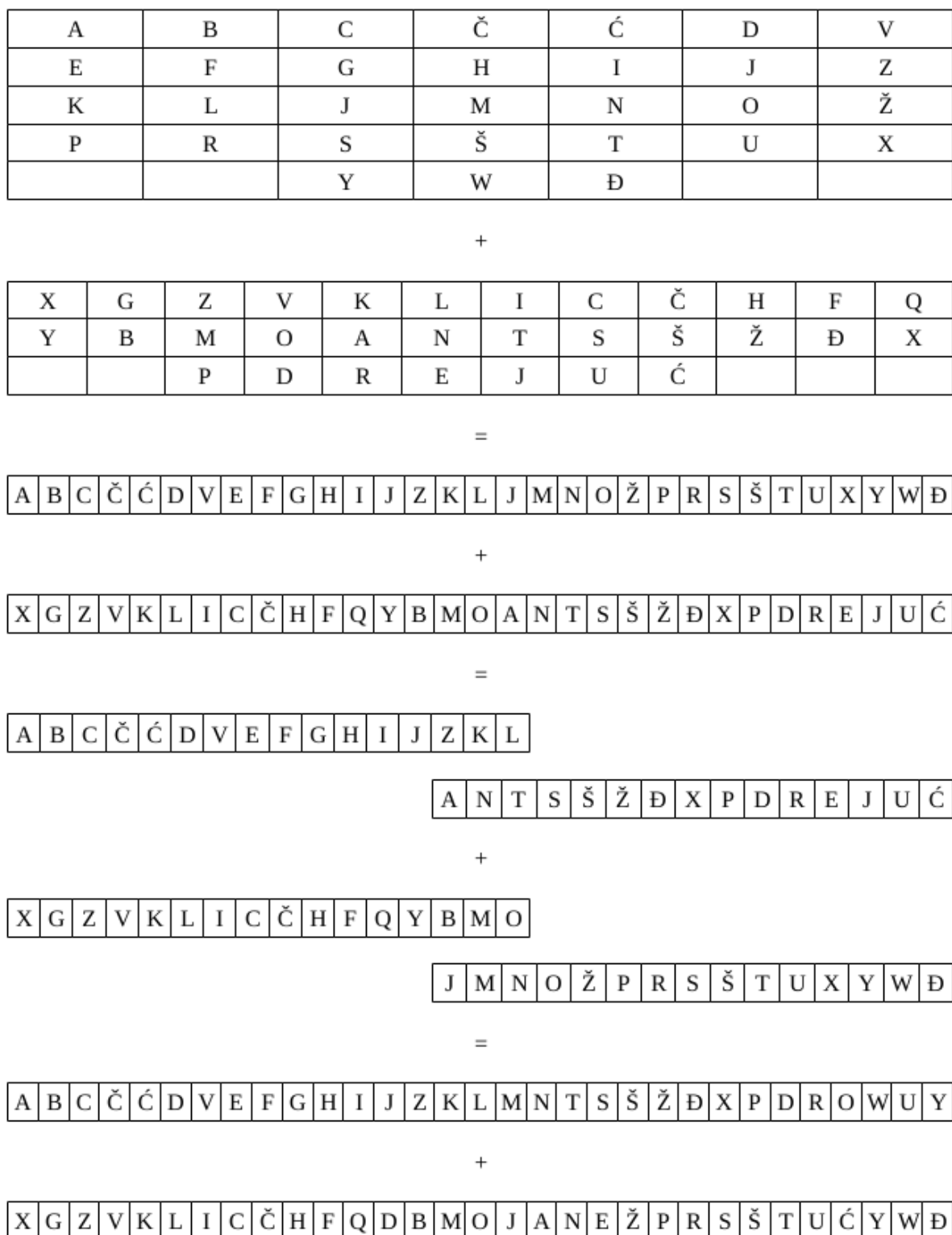




**Figure 3.8:** Shema genetskog algoritma

### 3.2.5. Primjena genetskog algoritma

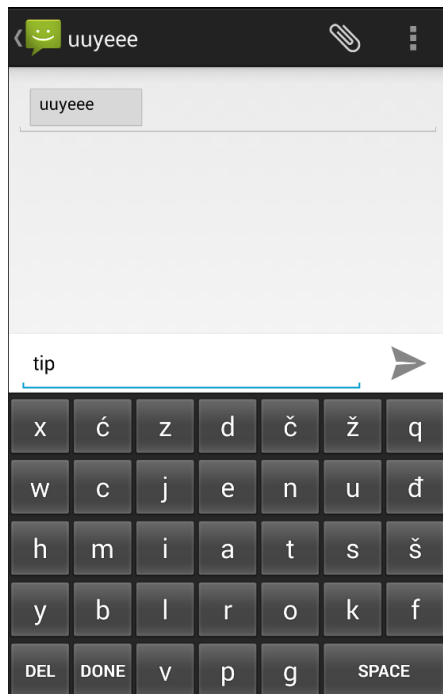
Jedinku u genetskom algoritmu predstavlja jedna tipkovnica sa svojim oblikom i rasporedom tipki. Jedna generacija sadrži 500 jedinki, a 1000 je postavljen za maksimalan broj generacija (uvjet zaustavljanja algoritma). Algoritam koristi operatore križanja, selekcije, mutacije i reprodukcije kako bi pronašao optimalno rješenje. Tip selekcije koji se koristio jest *k-turnirska selekcija* gdje je  $k$  postavljen na 24. Mutacija je implementirana na način da su slučajnom odabirom zamijenjene pozicije od  $p\%$  znakova na tipkovnici.  $p$  je postavljen na 0.3. Križanje je napravljeno tako da su svi znakovi s pojedine tipkovnice po redu nanizani u polje. Točka prekida polja je središnji element. Nakon prekida dvije polovice se zamijene i tada nastaju 2 nova polja. Svako polje može sadržavati dvostruke znakove. Taj problem je riješen tako da su u svakom pronađeni znakovi koji nedostaju i stavljeni umjesto duplikata. Proces križanja prikazan je na slici 3.9.



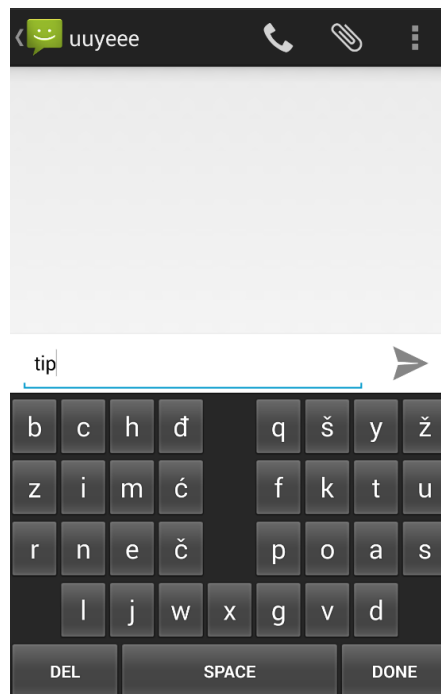
**Figure 3.9:** Primjer križanja tipkovnica

### 3.3. Rješenja genetskog algoritma

U ovom poglavlju će biti predstavljene najbolje tipkovnice koje su dobivene genetskim algoritmom te način njihove implementacije. Isprobano je nekoliko različitih izgleda (engl. *layout*) tipkovnica za korištenje pomoću jednog i pomoću dva prsta. Moguće je da postoje i neki drugi izgledi koji bi se pokazali optimalniji prema Fittsovom zakonu ali zbog ograničenosti vremena testirani su samo pojedini. Tipkovnica namijenjena korištenju s jednim prstom prikazana je na slici 3.10 i prema Fittsovom zakonu njena vremenska cijena iznosi 1.038. Može se lako uočiti da se sva često korištena slova u hrvatskom jeziku nalaze u sredini dok se ona rijetko korištena nalaze na rubovima. Takav razmještaj omogućuje brže unošenje teksta. Tipkovnica namijenjena korištenju s dva prsta prikazana je na slici 3.11 i njena vremenska cijena iznosi 0.948. Prema tome tipkovnica *FuryKey2* se pokazala učinkovitijom za korištenje što je i logično jer se koristeći dva prsta tekst može brže unositi nego s jednim. Kako bi se provjerilo imaju li navedeni rezultati smisla i je li *FuryKey1* zaista bolje koristiti s jednim prstom, a *FuryKey2* s dva prsta, potrebno je provesti eksperiment pomoću kojeg će se nad grupom ispitanika vrednovati obe tipkovnice. Provedeni eksperiment je opisan u poglavlju 4.



**Figure 3.10:** FuryKey1 - tipkovnica namijenjena za tipkanje jednim prstom ili stilusom



**Figure 3.11:** FuryKey2 - tipkovnica namijenjena za tipkanje pomoću dva prsta

### 3.3.1. Implementacija tipkovnica za Android operacijski sustav

Obe tipkovnice implementirane su za pametne uređaje s *Android* operacijskim sustavom. Većina Android uređaja nema fizičku tipkovnicu nego se oslanjaju na tipkovnicu na dodir (engl. *soft keyboard*). To otvara velik raspon mogućnosti. Moguće je isprogramirati tipkovnice različitih izgleda i razmještaja tipki te je lako promijeniti tipkovnici i koristiti onu koja nam najviše odgovara. Koristeći *Android SDK*<sup>3</sup> nova tipkovnica se može napraviti u nekoliko linija koda. *SDK* se pobrine za mnogo stvari poput prepoznavanja pritisnute tipke i iscrtavanja tipkovnice. Za potrebe ovog rada korišten je *Android Studio* kako bi se isprogramirale željene tipkovnice. Tipkovnice na dodir se svrstavaju u *IME* (Input Method Editor). *IME* je deklariran kao servis (engl. *service*) u datoteci *AndroidManifest.xml* i koristi *BIND\_INPUT\_METHOD* dozvolu kao što je prikazano na slici 3.12. Sam izgled tipkovnice opisan je u XML datoteci. Primjer takve datoteke prikazan je na slici 3.13. Svaka tipka može sadržavati sljedeće atribute:

**keyLabel:** ovaj atribut sadrži tekst koji se prikazuje na tipki

**codes:** ovaj atribut sadrži *unicode* vrijednosti znaka koji tipka predstavlja

**keyWidth:** ovim se atributom postavlja širina tipki, najčešće je zadan u postotcima

**isRepeatable:** ako je ovaj atribut postavljen na *true* onda se znak više puta ispisiše ukoliko se tipka drži pritisnuta

```
1 <service android:name=".SimpleIME"
2   android:label="@string/simple_ime"
3   android:permission="android.permission.BIND_INPUT_METHOD"
4 >
5   <meta-data android:name="android.view.im" android:resource="@xml/method"/>
6   <intent-filter>
7     <action android:name="android.view.InputMethod" />
8   </intent-filter>
9 </service>
```

Figure 3.12: Datoteka *AndroidManifest.xml*

Potrebno je napraviti i klasu koja predstavlja *IME* servis. Takva klasa nasljeđuje *InputMethodService* i u njoj se može definirati što se događa pritiskom na tipku i ostale stvari kao što je prikazano na slici 3.14. Konačno, potrebno je instalirati tipkovnicu na pametni uređaj te je u postavka odabrati.

---

<sup>3</sup>Software Development Kit

```

01 <Keyboard xmlns:android="http://schemas.android.com/apk/res/android"
02     android:keyWidth="10%p"
03     android:horizontalGap="0px"
04     android:verticalGap="0px"
05     android:keyHeight="60dp"
06 >
07     <Row>
08         <Key android:codes="49" android:keyLabel="1" android:keyEdgeFlags="left"/>
09         <Key android:codes="50" android:keyLabel="2"/>
10         <Key android:codes="51" android:keyLabel="3"/>
11         <Key android:codes="52" android:keyLabel="4"/>
12         <Key android:codes="53" android:keyLabel="5"/>
13         <Key android:codes="54" android:keyLabel="6"/>
14         <Key android:codes="55" android:keyLabel="7"/>
15         <Key android:codes="56" android:keyLabel="8"/>
16         <Key android:codes="57" android:keyLabel="9"/>
17         <Key android:codes="48" android:keyLabel="0" android:keyEdgeFlags="right"/>
18     </Row>
19     <Row>
20         <Key android:codes="113" android:keyLabel="q" android:keyEdgeFlags="left"/>
21         <Key android:codes="119" android:keyLabel="w"/>

```

**Figure 3.13:** Datoteka u kojoj se postavlja izgled tipkovnice

```

01 public class SimpleIME extends InputMethodService
02     implements OnKeyboardActionListener{
03
04     private KeyboardView kv;
05     private Keyboard keyboard;
06
07     private boolean caps = false;
08
09     @Override
10     public void onKey(int primaryCode, int[] keyCodes) {
11
12     }
13
14     @Override
15     public void onPress(int primaryCode) {
16
17     }
18
19     @Override
20     public void onRelease(int primaryCode) {
21

```

**Figure 3.14:** Klasa koja predstavlja IME servis

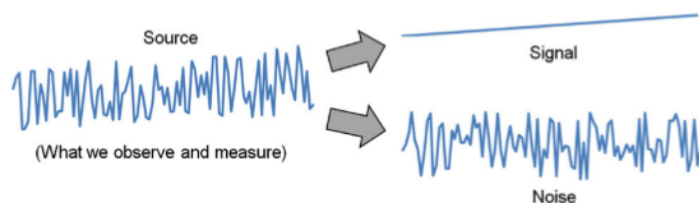
## 4. Provedba eksperimenta i obrada rezultata

Koristeći eksperimentalne metode, znanje se prikuplja kontroliranim eksperimentima provedenim u laboratorijskim uvjetima. Prikupljanje znanja preko eksperimenta može značiti otkrivanje nečeg novog ali može značiti i proučavanje već postojećeg u svrhu potvrđivanja, ispravljanja, integriranja ili proširivanja. S obzirom da se eksperimenti odvijaju pod strogo kontroliranim uvjetima, kaos i raznolikost stvarnog svijeta su znatno smanjeni ili čak potpuno uklonjeni. Kontrolirani eksperimenti moraju imati barem dvije varijable: *podešavana varijabla* (engl. *manipulated variable*) i *odzivna varijabla* (engl. *response variable*). Podešavana varijabla je najčešće svojstvo sučelja ili način interakcije koji je dan sudionicima eksperimenta u različitim konfiguracijama. Podešavanje varijable podrazumijeva sistematično podvrgavanje ispitanika različitim konfiguracijama sučelja i načinima interakcije. Barem dvije različite konfiguracije moraju postojati kako bi se proveo kontrolirani eksperiment. Prema tome, usporedba je ključna u eksperimentalnoj metodi. Podešavana varijabla se još naziva i *nezavisna varijabla* (engl. *independent variable*). Odzivna varijabla je svojstvo ljudskog ponašanja koje se može promatrati, kvantificirati i mjeriti. Najčešća odzivna varijabla je vrijeme potrebno da ispitanik izvrši određenu konfiguraciju koja se testira. Odzivna varijabla se još naziva i *zavisna varijabla* (engl. *dependent variable*).

Kontrolirani eksperiment, ako je kvalitetno oblikovan i proveden, pruža kvalitetan način izvlačenja zaključka iz podataka. Odnos nezavisne i zavisne varijable je *uzročno-posljedični*. Podešavanje sučelja i načina interakcije su uzrok promatranim svojstvima u zavisnoj varijabli, što je zapravo posljedica. Uzročno-posljedični zaključci se ne mogu dobiti koristeći korelacijskom metodom ili metodom promatranja.

## 4.1. Oblikovanje eksperimenta

Jedan način razmišljanja o oblikovanju eksperimenata je koristeći signal-šum metaforu. U toj se metafori, sva promatranja i mjerenja dijele na dvije komponente: signal i šum (slika 4.1). Na priloženoj slici možemo vidjeti kako izvor generalno raste, ali je ipak učinak šuma prevelik te je to teško uočiti. Ako se izvor razlomi na signal i šum, lako se može uočiti ponašanje signala. Kod eksperimenata koji uključuju ljudske ispitanike, signal podrazumijeva zavisne i nezavisne varijable dok šum predstavlja sve ostalo - slučajne, uglavnom neželjene utjecaje. To može biti temperatura, osvjetljenje, pozadinska buka pa čak i sami ispitanici, npr. neki od njih mogu imati loš dan, neki dobar. To sve bitno utječe na rezultate eksperimenta. Zbog svega toga proces oblikovanja eksperimenta znači pojačavanje signala uz istovremeno smanjenje šuma. To se radi tako da se pažljivo biraju postavke eksperimenta u smislu odabira varijabli i njihovi konfiguracija.



**Figure 4.1:** Signal-šum konceptualizacija eksperimentalnog oblikovanja

Oblikovanje eksperimenta je proces slaganja svih potrebnih dijelova kako bi se ispitala hipoteza. To uključuje određivanje koje varijable, zadatke i procedure koristiti, koliko ispitanika uzeti, kako ih rasporediti i tako dalje. U nastavku je opisano nekoliko pojmova povezanih s oblikovanjem eksperimenta u ovom radu.

Ako se svako mjerenje ponavlja za svakog ispitanika onda je oblik eksperimenta *unutar sudionika* (engl. *within-subjects*). Svi ispitanici tada imaju potpuno jednake uvjete pri provedbi eksperimenta. Kod *između sudionika* (engl. *between-subjects*) oblika ispitanici se dijele na više grupa i svaka grupa dobija drugačije uvjete ispitivanja dok unutar jedne grupe ispitanici imaju iste uvjete ispitivanja. Načelno, kod *između sudionika* oblikovanja potrebno je više ispitanika da bi se proveo isti broj mjerenja, a kod *unutar sudionika* potrebno je provesti više testiranja na pojedinom ispitaniku. Ponekad eksperiment mora biti *između sudionika* ako npr. promatramo učinak ljevorukih i desnorukih ispitanika na određenom testu. Ponekad mora biti *unutar sudionika* ako se npr. promatra stjecanje iskustva kod pojedinog ispitanika. Prednost *unutar su-*

*dionika* je njegova jednostavnost, nije potrebno uravnotežavati grupe jer postoji samo jedna, dok se kod *između sudionika* potrebno pobrinuti da grupe budu više-manje jednake. To se najčešće radi tako da se ispitanici nasumično odaberu za svaku grupu. Prednost *između sudionika* je to što se smanjuje utjecaj između uvjeta ispitivanja. Do toga dolazi kada ispitanik prelazi s jednog testa na drugi, u tom procesu na drugi test prenosi iskustva i znanje koje je stekao u prethodnom.

Kao što je već navedeno, kad se koristi *unutar sudionika* oblik eksperimenta, svakog ispitanika se testira na svakom testu te se zbog toga učinak ispitanika svakim testom može poboljšati ili čak pogoršati u nekim slučajevima. To je razlog radi kojeg se ponekad, u testovima nakon prvog, ne dobivaju stvarni rezultati. Da bi se to popravilo koristi se metoda suprotstavljanja i latinski kvadrati. Svaki ispitanik dobija testove u drugačijem poretku. Slika 4.2 predstavlja latinski kvadrat reda 3. Svako slovo predstavlja test koji ispitanik izvršava. U ovom slučaju svi bi ispitanici bili podijeljeni u 3 grupe i svaki bi redak odgovarao redosljedu kojim bi pojedina grupa radila zadane testove.

A	B	C
B	C	A
C	A	B

**Figure 4.2:** Latinski kvadrat reda 3

## 4.2. Eksperimentalno vrednovanje tipkovnica

Cilj eksperimenta je vrednovati prethodno opisane tipkovnice *FuryKey1* i *FuryKey2* te ih usporediti s QWERTZ tipkovnicom. Nadalje, potrebno je ispitati na koji je način najbolje koristiti svaku od navedenih tipkovnica (horizontalno ili vertikalno, tipkanje s jednim ili dva prsta).

Provedeni eksperiment oblikovan je na sljedeći način:

### Varijable eksperimenta

– Nezavisne varijable:

- tipkovnica: *FuryKey1*, *FuryKey2*, QWERTZ
- veličina pametnog uređaja: 7", 10"



- način korištenja: jedan prst, dva prsta
  - položaj pametnog uređaja: horizontalno (engl. *landscape*), vertikalno (engl. *portrait*)
- Zavisne varijable:
- brzina unošenja teksta: broj riječi koje ispitanik prosječno unese u jednoj minuti, mjerna jedinica je stoga *wpm* (engl. *words per minute*)
  - broj pogrešaka ER (engl. *error rate*): broj pogrešaka koje ispitanik napravi tijekom unosa teksta
- Zbunjujuće varijable
- iskustvo: uzevši u obzir da će ispitanik jednu tipkovnicu koristiti više puta za redom, doći će do pojave učenja što nije dobro za eksperiment, zato se koristio latinski kvadrat da bi se smanjio učinak učenja

Eksperiment je oblikovan kao mješavina *unutar sudionika* i *između sudionika*. Osam ispitanika podijeljeno je u dvije grupe po četvero. Jedna je grupa koristila samo *FuryKey1*, a druga samo *FuryKey2*. Na taj način neće doći do nikakve međuovisnosti rezultata jedne i druge tipkovnice. Kako bi se smanjio učinak učenja napravljen je latinski kvadrat i prema njemu su svi ispitanici u jednoj grupi testirani različitim redoslijedom. On je prikazan na slici 4.3 gdje slova *H* i *V* označavaju položaj pametnog uređaja (horizontalno, vertikalno), brojevi 1 i 2 označavaju broj prstiju koji se koristi pri unošenju teksta. Svi ispitanici su unosili 2 različita bloka teksta za svaki način korištenja. Sve tipkovnice testirane su na pametnim uređajima veličina 7" i 10".

7"+1+H	10"+1+V	7"+2+H	10"+2+V
10"+1+V	7"+2+H	10"+2+V	7"+1+H
10"+2+H	7"+2+V	10"+1+H	7"+1+V
7"+2+V	10"+1+H	7"+1+V	10"+2+H

**Figure 4.3:** Latinski kvadrat reda 4

#### 4.2.1. Prikaz i interpretacija rezultata

Mjerna jedinica svih prikazanih rezultata je *wpm*. Rezultati su dobiveni prema izrazu 4.1 gdje  $t$  predstavlja izmjereno vrijeme u sekundama,  $n$  broj unešenih riječi, a  $ER$  broj pogrešaka koje je korisnik pri unosu načinio.

$$wpm = \frac{60(n - ER)}{t} \quad (4.1)$$

Tablica 4.1 predstavlja sve dobivene rezultate za *FuryKey1*, tablica 4.2 za *FuryKey2* dok su na tablicama 4.3 i 4.4 prikazani prosječni rezultati za oba bloka teksta. Tablica 4.5 prikazuje rezultate dobivene pri testiranju QWERTZ tipkovnice. Na slikama 4.4 i 4.5 grafički su prikazani rezultati iz prethodnih tablica. Kod oba grafa se može lako uočiti povećanje brzine pri unošenju drugog bloka teksta. Graf 4.6 prikazuje usporedbu svih testiranih tipkovnica. Iz grafa se vidi kako se QWERTZ pokazao dosta učinkovitiji, ali to je i očekivano. Nadalje, može se vidjeti kako se kod nekih načina unošenja teksta boljom pokazala *FuryKey1*, a kod nekih *FuryKey2*. Te rezultate potrebno je još detaljnije ispitati. Na grafovima 4.7 i 4.8 prikazana je usporedba horizontalnog i vertikalnog načina držanja za obe tipkovnice. Vidi se kako je kod obe tipkovnice učinkovitiji vertikalni način. To je zbog toga što je tada međusobna udaljenost tipki manja nego kod horizontalnog načina. Grafovi 4.9 i 4.10 prikazuju usporedbu tipkanja s jednim i dva prsta. Tipkovnica *FuryKey1* namijenjena je za korištenje s jednim prstom, a *FuryKey2* za korištenje s dva prsta. To su rezultati i potvrdili. Konačno, graf 4.11 prikazuje ukupnu usporedbu obe tipkovnice (cjelokupni prosjek svih položaja i načina unošenja pametnog uređaja). Rezultati eksperimenta pokazuju da je *FuryKey2* učinkovitija.

Bitno je napomenuti da su za provedbu eksperimenta tipkovnice testirane nad *samo* 8 ispitanika. Zbog toga ovi podatci ne mogu biti stvarni pokazatelj učinkovitosti testiranih tipkovnica. Potrebno je eksperiment provesti s mnogo više ispitanika, ali to nije napravljeno zbog ograničenosti resursa i vremena. Bez obzira na to, prikazani rezultati i način mjerenja služi kao model koji bi poslužio za provođenje eksperimenta većeg razmjera.

Ispitanik		1 prst		2 prsta	
		Horizontalno	Vertikalno	Horizontalno	Vertikalno
1	Blok 1	7.5	6.46	6.29	8.66
	Blok 2	6.89	9.44	7.92	10
2	Blok 1	8.4	5.87	8.16	6.22
	Blok 2	7.64	5.49	7.78	7.71
3	Blok 1	8.75	9.66	5	7.24
	Blok 2	8.57	10	5.56	7.64
4	Blok 1	7.09	7.78	7.64	6.22
	Blok 2	8.48	8.84	8.75	6.18
Prosjeak	Blok 1	7.94	7.44	6.77	7.09
	Blok 2	7.9	8.44	7.5	7.88

**Table 4.1:** Rezultati mjerenja za FuryKey1

Ispitanik		1 prst		2 prsta	
		Horizontalno	Vertikalno	Horizontalno	Vertikalno
1	Blok 1	5.92	4.38	8.04	9.33
	Blok 2	5	6.36	6.77	8.08
2	Blok 1	7.64	5.22	7.12	7.3
	Blok 2	8.4	5.6	8.84	7.78
3	Blok 1	8.48	10	6.32	8.84
	Blok 2	9.4	13.13	8.32	8.48
4	Blok 1	7.12	7.64	8.32	7.64
	Blok 2	7.92	7.22	9.44	8.67
Prosjeak	Blok 1	7.29	6.81	7.45	8.28
	Blok 2	7.68	8.08	8.34	8.25

**Table 4.2:** Rezultati mjerenja za FuryKey2

Ispitanik	1 prst		2 prsta	
	Horizontalno	Vertikalno	Horizontalno	Vertikalno
1	7.2	7.95	7.11	9.33
2	8.02	5.68	7.97	6.97
3	8.66	9.83	5.28	7.44
4	7.79	8.31	8.2	6.2
Prosjeak	7.92	7.94	7.14	7.48

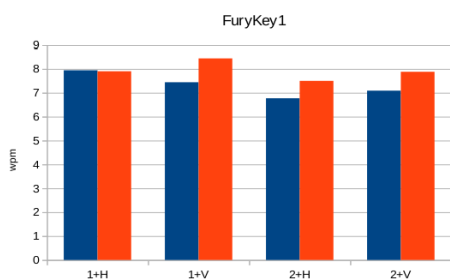
**Table 4.3:** Prosječni rezultati za FuryKey1

Ispitanik	1 prst		2 prsta	
	Horizontalno	Vertikalno	Horizontalno	Vertikalno
1	5.46	5.37	7.41	8.71
2	8.02	5.41	7.98	7.54
3	8.94	11.57	7.32	8.66
4	7.52	7.43	8.88	8.16
Prosjeak	7.49	7.44	7.9	8.27

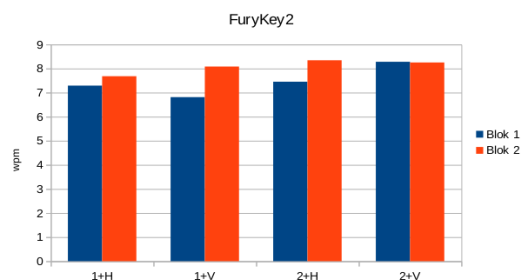
**Table 4.4:** Prosječni rezultati za FuryKey2

Ispitanik	1 prst		2 prsta	
	Horizontalno	Vertikalno	Horizontalno	Vertikalno
1	16.8	22.11	22.7	23.33
2	18.67	20.49	20	17.5
3	21.54	21.54	15.55	24
4	21.54	28	24	23.33
5	12	19.54	27.1	19.53
6	16.47	21	28	20
7	19.53	26.25	22.11	26.25
8	22.11	23.33	15.56	24
Prosjeak	18.58	22.78	21.88	22.24

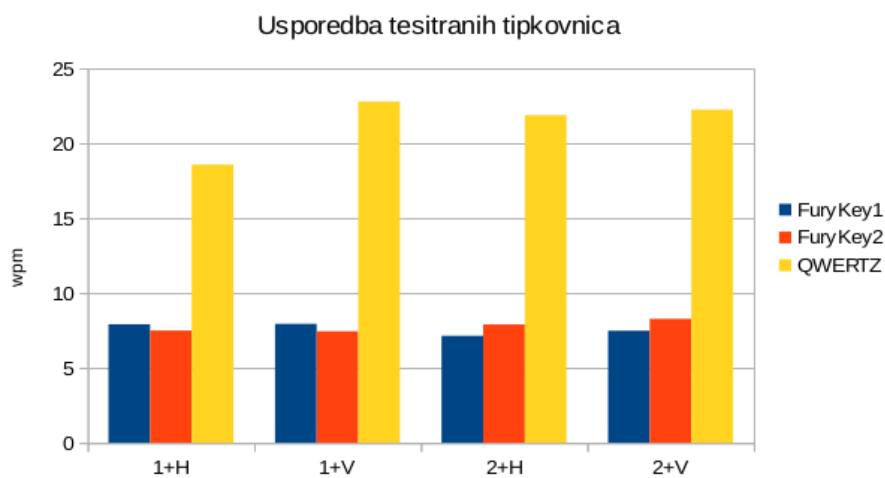
**Table 4.5:** Rezultati mjerenja za QWERTZ



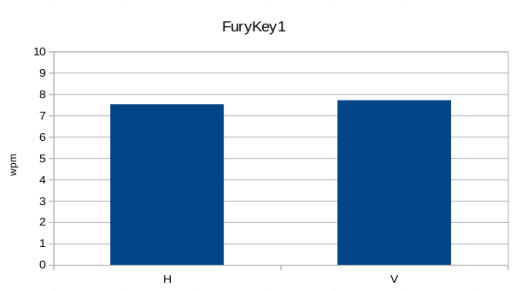
**Figure 4.4:** Grafički prikaz rezultata za FuryKey1



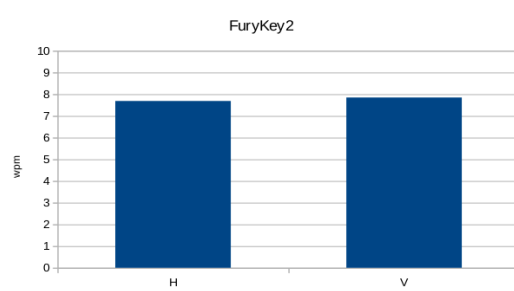
**Figure 4.5:** Grafički prikaz rezultata za FuryKey2



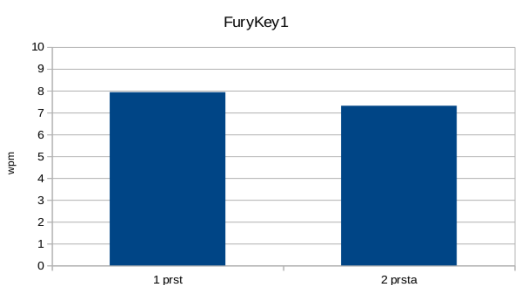
**Figure 4.6:** Usporedba FuryKey1, FuryKey2 i QWERTZ



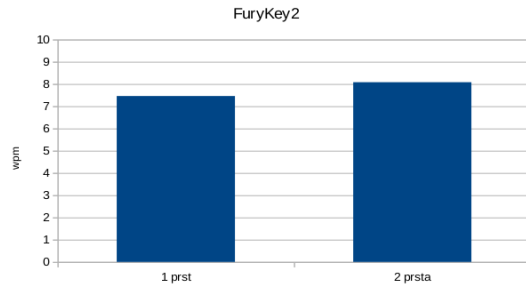
**Figure 4.7:** Usporedba za horizontalni i vertikalni način držanja ( $H = 7.52wpm$ ,  $V = 7.71wpm$ )



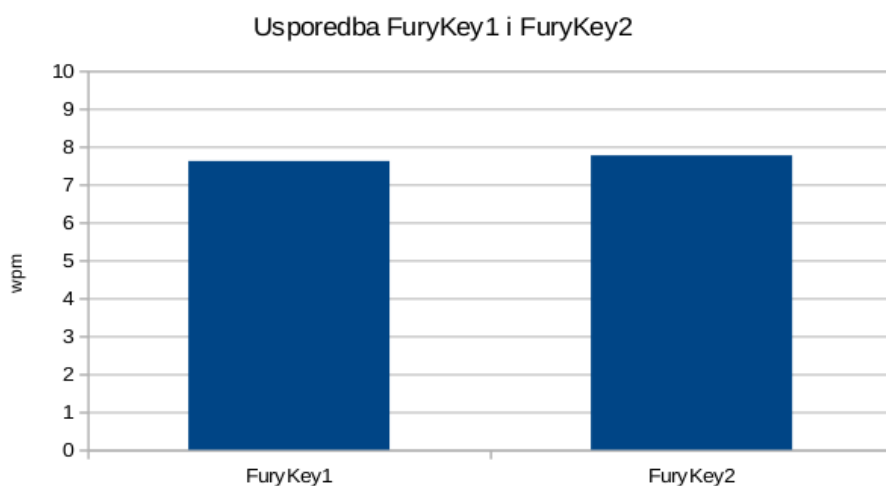
**Figure 4.8:** Usporedba za horizontalni i vertikalni način držanja ( $H = 7.69wpm$ ,  $V = 7.85wpm$ )



**Figure 4.9:** Usporedba za korištenje s jednim i s dva prsta ( $1prst = 7.93wpm$ ,  $2prsta = 7.31wpm$ )



**Figure 4.10:** Usporedba za korištenje s jednim i s dva prsta ( $1prst = 7.46wpm$ ,  $2prsta = 8.08wpm$ )



**Figure 4.11:** Ukupna usporedba učinkovitosti ( $FuryKey1 = 7.62wpm$ ,  $FuryKey2 = 7.77wpm$ )

## 5. Zaključak

Pronalazak idealnog razmještaja znakova na tipkovnici predstavlja velik problem. Kao što je već opisano, standardna QWERTY tipkovnica ima mnogo mana, a postoji još i mnogo drugih tipkovnica koje je nastoje zamijeniti. Nadalje, tipkovnicama je često potrebna lokalizacija te će krajnji korisnici težiti uređajima s boljom lokalnom podrškom ukoliko krivulja učenja (engl. *learning curve*) nije prevelika. Zbog toga je potrebno proučiti statistiku za ciljani jezik i iz toga pokušati oblikovati što kvalitetniju tipkovnicu. U ovom radu je hrvatski jezik uzet kao primjer. Fittsov model predstavlja jednostavan ali učinkovit model pomoću kojeg se mogu popboljšati razna korisnička sučelja. Osim njega mogu se koristiti još mnogi drugi koji nisu navedeni u ovom radu. Provođenje eksperimenata se također pokazao kako jedan od ključnih čimbenika u oblikovanju bilo kakvog korisničkog sučelja. Ipak su takva sučelja namijenjena samim korisnicima pa se ispitivanjem mora provjeriti njihova kvaliteta.

Ovaj rad se može proširiti na dosta načina. Potrebno je pokrenuti genetski algoritam s još mnogo različitih parametara kako bi se pronašli oni najoptimalniji. Nadalje, moguće je implementirati još više vrsta selekcije, mutacije i križanja. Što se tiče internacionalizacije, u algoritam bi se vrlo lako dali uvesti bigrami za neki drugi jezik pa vidjeti kakva bi bila "najbolja" tipkovnica za primjerice njemački ili španjolski jezik. Osim toga, za budući rad se predviđa provođenje opsežnijeg eksperimenta kako bi se tipkovnice što kvalitetnije vrednovale.

## **Alternativna tastatura za zaslone osjetljive na dodir temeljena na Fittsovom zakonu**

### **Sažetak**

Predstavlja se problem pronalaska idealne tipkovnice za pojedini jezik te se ukazuju mane standardne QWERTY tipkovnice. Kako bi se pronašlo rješenje problema koristi se jednostavan Fittsov zakon pomoću kojeg se brojčano vrednuje svaka tipkovnica. Zbog velikog broja mogućih rješenja koristi se genetski algoritam u svrhu optimizacije i pronalaska tipkovnice s optimalnim razmještajem. Navodi se važnost eksperimentalne metode u potvrđivanju hipoteza.

**Ključne riječi:** tipkovnica, QWERTY, eksperiment, Fittsov zakon, genetski algoritam, jezik

## **Alternative Touchscreen Keyboard Based on Fitts Law**

### **Abstract**

Problem of finding ideal keyboard for specific language is introduced, also a lot of disadvantages of standard QWERTY keyboard are mentioned. Fitts' Law is used in order to evaluate each keyboard. Because of many possible solutions, genetic algorithm is used to find keyboard with optimal layout and arrangement of keys. Furthermore, significance of experimental method is mentioned in order to test different hypothesis.

**Keywords:** keyboard, QWERTY, experiment, Fitts' law, genetic algorithm, language