

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

1. Uvod	1
2. Alternativne tipkovnice	3
2.1. Dvorak	3
2.2. ATOMIK	4
2.3. FITALY	4
2.4. Colemak	4
3. Dizajniranje alternativne tipkovnice	6
3.1. Opis problema	6
3.1.1. Fittsov zakon	7
3.1.2. Genetski algoritam	9
3.2. Rezultati i implementacija	15
3.2.1. Implementacija tipkovnica za Android operacijski sustav	16
4. Provedba eksperimenta i obrada rezultata	17
5. Zaključak	18

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 Wisconsin, 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 bigrame¹ 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.

¹Frekvencija 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.

TODO dodat još o dvorak tipkovnici (bookmarkano)

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*¹. Colemak tipkovnica ima QWERTY bazu, a od nje se razlikuje u 17

¹engl. 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

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	←												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

u neodgovarajućem položaju.

3. Dizajniranje alternativne tipkovnice

Ovo poglavlje obuhvaća opis samog problema te pristup njegovog rješavanja.

TODO još 2-3 rečenice dodat

3.1. Opis problema

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 2.1.1.). 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.

//TODO staviti sliku za PAS

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 razmjesta, što bi vjerojatno potrajalo nekoliko stoljeća. Ovaj problem predstavlja tipičan problem optimizacije, pitanje je kako pronaći minimum funkcije zadane Fittsovim zakonom uzevši u obzir sve moguće razmjeste slova. Postoje razne metode za riješiti taj problem. U ovom radu se koristio genetski algoritam. Genetski algoritam i način na koji je iskorišten detaljno je opisan u poglavlju 2.1.2.

3.1.1. Fittsov zakon

Fittsov zakon je opisni model ljudskog pokreta koji se primarno koristi u *HCI-u*¹ i *ergonomiji*². 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.

$$ID = \log_2(2D/W) \quad (3.1)$$

Fitts je također predložio indeks izvedbe (engl. *index of performance*) kao mjerilo

¹Interakcija čovjeka i računala.

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

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 vremenom 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.

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

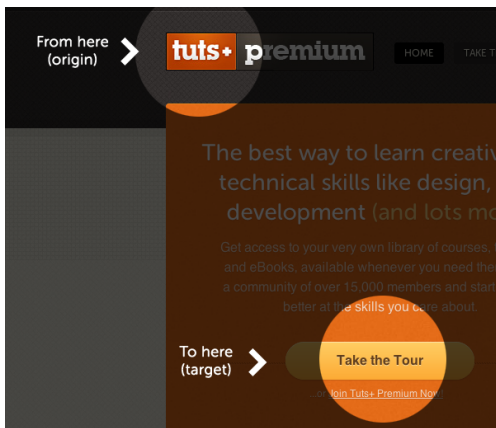


Figure 3.1: Uporaba Fittsovog zakona



Figure 3.2: Uporaba Fittsovog zakona

Fittsov zakon u mom radu

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 cijena 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.

3.1.2. 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

x	ć	z	d	č	ž	q
w	c	j	e	n	u	đ
h	m	l	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	i	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

okolini. Analogija evolucije kao prirodnog procesa i genetskog algoritma kao metode 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.

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.

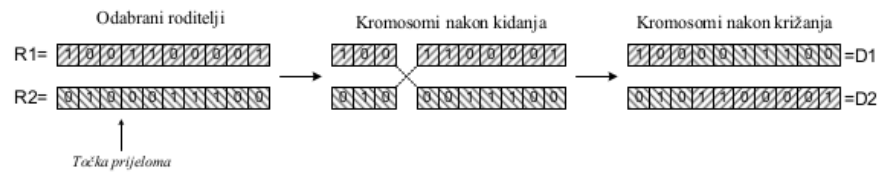


Figure 3.5: Križanje s jednom točkom prijeloma

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. *Jednostavna mutacija* svaki bit kromosoma mijenja s jednakom vjerojatnošću p_m .

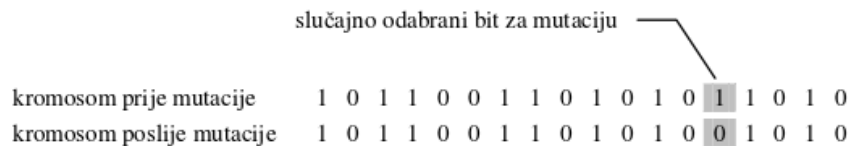


Figure 3.6: Jednostavna mutacija

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

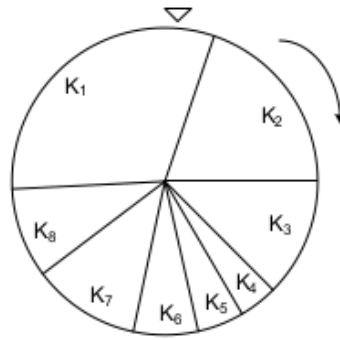


Figure 3.7: Proporcionalna selekcija

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

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.

Operator reprodukcije

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

Algorithm 1 Genetski algoritam - pseudokod

$P = \text{stvariPocetnuPopulaciju(VELICINA)}$

evaluiraj(P)

repeat

 novaPopulacija $P' = \emptyset$

while $\text{velicina}(P') < \text{VELICINA}$ **do**

 odaberi $R1$ i $R2$ iz P

$\{D1, D2\} = \text{krizaj}(R1, R2)$

 mutiraj $D1$, mutiraj $D2$

 dodaj $D1$ i $D2$ u P'

end while

$P = P'$

until uvjetZadovoljen

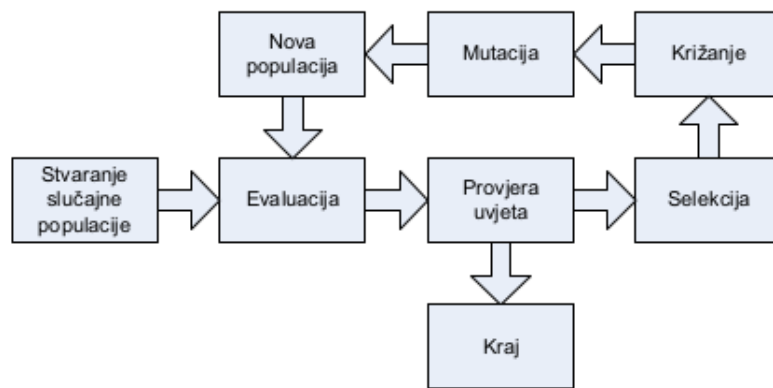


Figure 3.8: Shema genetskog algoritma

Genetski algoritam u mom radu

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 duple 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.

TODO opisat detaljnije ?

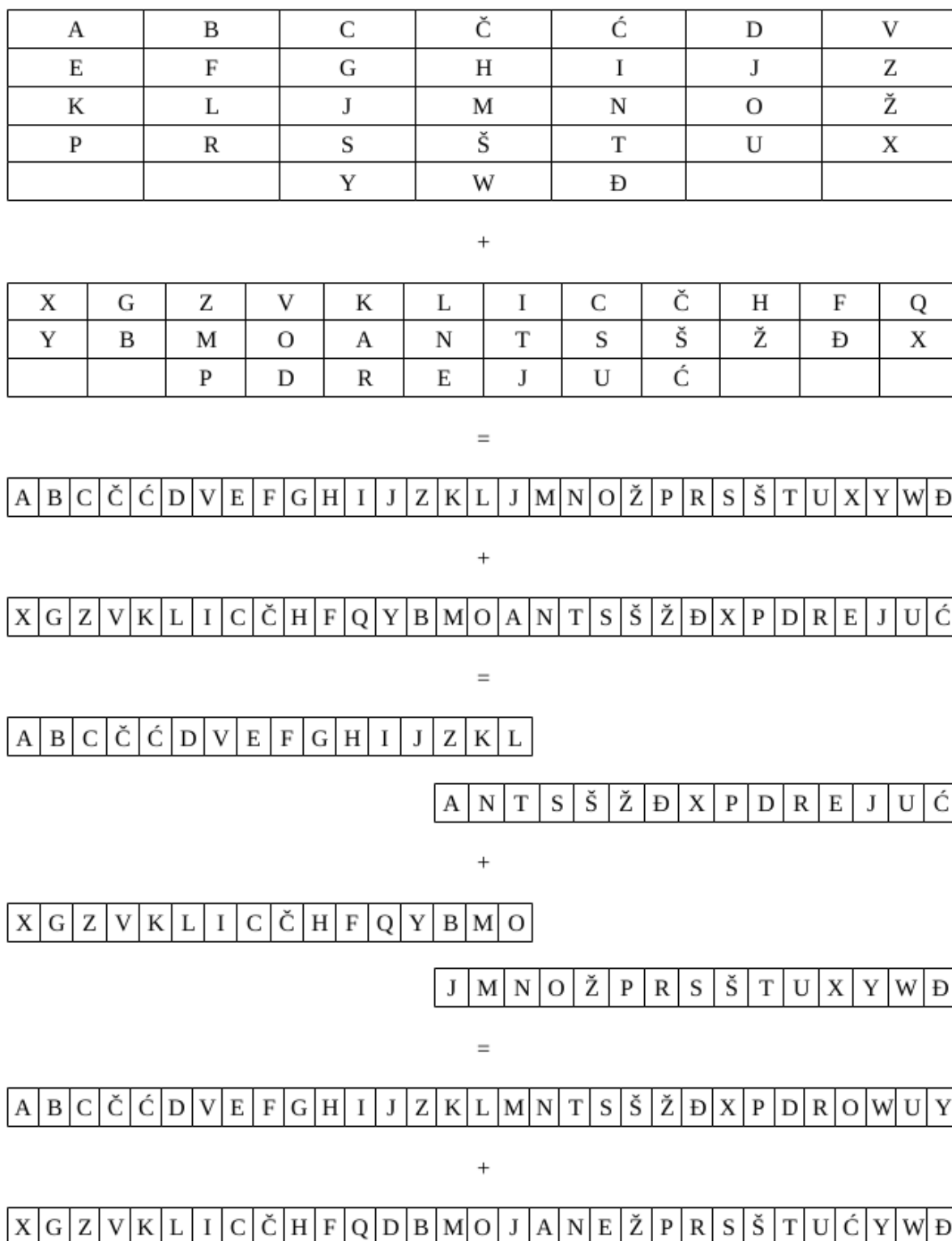


Figure 3.9: Primjer križanja tipkovnica

3.2. Rezultati i implementacija

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. Neke od njih prikazane su na sljedećim slikama //TODO ref bla bla

Moguće je da postoje i neki 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. 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 evaluirati obe tipkovnice. Provedeni eksperiment je opisan u poglavlju 3.

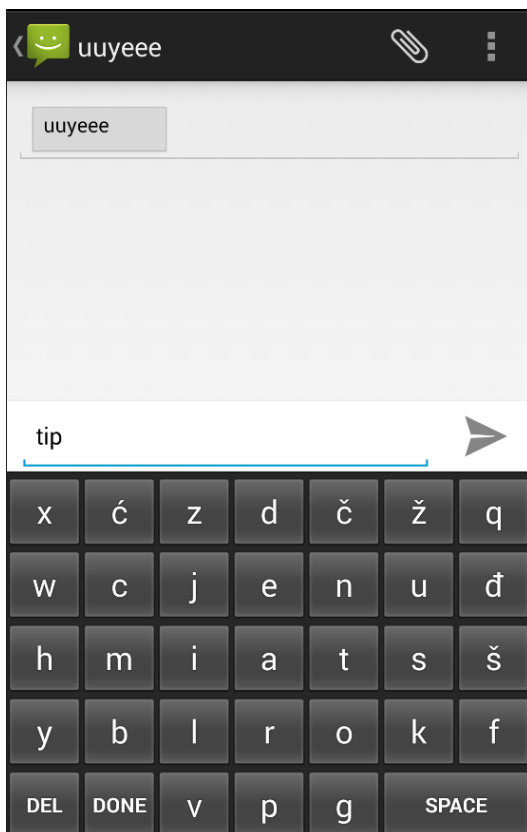


Figure 3.10: FuryKey1 - tipkovnica namijenjena za tipkanje jednim prstom ili stilusom

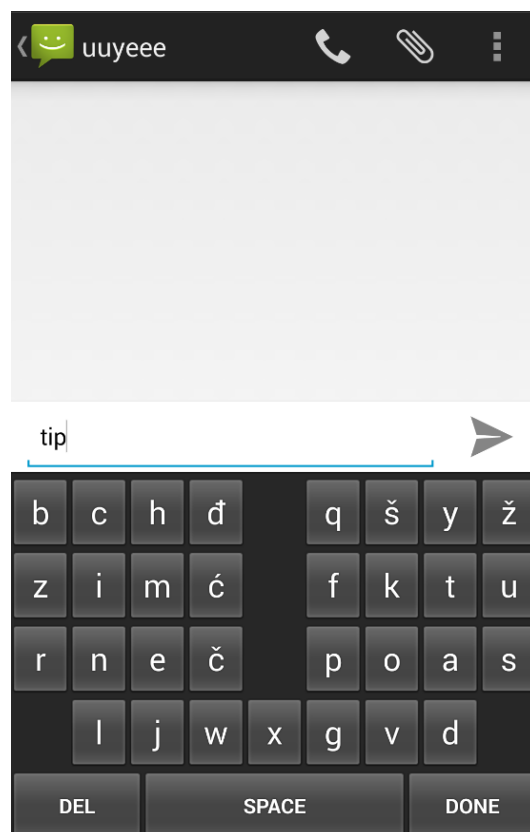


Figure 3.11: FuryKey2 - tipkovnica namijenjena za tipkanje s dva prsta

3.2.1. Implementacija tipkovnica za Android operacijski sustav

Obe tipkovnice implementirane su za pametne uređaje s *Android* operacijskim sustavom. //TODO napisat što je to Android ? napisat ponešto o soft keyboardima i način implementacije soft keyboarda ? (pitat mentora)

4. Provedba eksperimenta i obrada rezultata

5. Zaključak

Zaključak.

Alternativna tastatura za zaslone osjetljive na dodir temeljena na Fittsovom zakonu

Sažetak

Sažetak na hrvatskom jeziku.

Ključne riječi: Ključne riječi, odvojene zarezima.

Alternative Touchscreen Keyboard Based on Fitts Law

Abstract

Abstract.

Keywords: Keywords.