

Operacija pokazivanja-i-odabira zasnovana na glasovnom upravljanju

Ecimović, Valentina

Master's thesis / Diplomski rad

2023

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:190:025397>

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

Download date / Datum preuzimanja: **2024-07-10**



Repository / Repozitorij:

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



SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Operacija pokazivanja-i-odabira
zasnovana na glasovnom upravljanju**

Rijeka, rujan 2023.

Valentina Ecimović
0069082653

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni diplomski studij računarstva

Diplomski rad

**Operacija pokazivanja-i-odabira
zasnovana na glasovnom upravljanju**

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

Rijeka, rujan 2023.

Valentina Ecimović
0069082653

Umjesto ove stranice umetnuti zadatak
za završni ili diplomski rad

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradila ovaj rad pod vodstvom mentora izv. prof. dr. sc. Sandija Ljubića.

Rijeka, rujan 2023.

Valentina Ećimović

Zahvala

Ponajprije se zahvaljujem svome mentoru, izv. prof. dr. sc. Sandiju Ljubiću, na podršci tijekom pripreme ovog diplomskog rada, kao i na doprinosu tijekom čitavog studija.

Također, želim izraziti zahvalnost svim kolegama i prijateljima koji su doprinijeli izradi i testiranju projekta opisanog u radu.

Na kraju, zahvaljujem se svojoj obitelji, s posebnim istaknućem za moju majku, koja mi je omogućila sve ovo i bila stalna potpora.

Sadržaj

Popis slika	viii
Popis tablica	x
1 Uvod	1
2 Prepoznavanje govora	3
2.1 Povijest tehnologije prepoznavanja govora	3
2.2 Algoritmi za prepoznavanje govora	4
3 Implementacija	6
3.1 Korištene tehnologije	6
3.1.1 Python	6
3.1.2 Vosk	9
3.1.3 Axialis CursorWorkshop	9
3.2 Modeliranje sustava za glasovno upravljanje pokazivačem	10
3.2.1 Funkcijske značajke modaliteta za glasovno upravljanje pokazivačem	11
3.2.2 Implementacijski detalji modaliteta za glasovno upravljanje pokazivačem	15
3.3 Aplikacija za testiranje modaliteta	23

Sadržaj

4	Eksperimentalno vrednovanje	25
4.1	Fittsov zakon	25
4.2	Postupak testiranja modaliteta	27
4.2.1	Aplikacija FittsStudy	28
4.2.2	Vlastita aplikacija za testiranje	31
4.2.3	Upitnici	31
5	Analiza i rezultati	34
5.1	Ispitna skupina	34
5.2	Analiza rezultata eksperimenta	34
5.2.1	Propusnost	35
5.2.2	Vrijeme	36
5.3	Analiza rezultata upitnika	37
5.3.1	NASA-TLX upitnik	38
5.3.2	Upitnik upotrebljivosti	45
5.3.3	Završni upitnik	50
5.4	Prediktivni modeli	51
6	Zaključak	54
	Bibliografija	56
	Pojmovnik	58
	Sažetak	58
A	Grafički prikaz modaliteta za eksperiment	60

Popis slika

3.1	Dizajn pokazivača miša unutar aplikacije CursorWorkshop	10
3.2	Grafički prikaz određivanja smjera pokazivača bez direktnog određivanja kuta	11
3.3	Grafički prikaz određivanja smjera pokazivača bez direktnog određivanja kuta koristeći animirani pokazivač	12
3.4	Grafički prikaz određivanja smjera pokazivača s direktnim određivanjem kuta	13
3.5	Izbornik za postavljanje parametara unutar vlastite aplikacije za testiranje	24
3.6	Sučelje vlastite aplikacije za testiranje modaliteta interakcije	24
4.1	Balansirani latinski kvadrat	28
4.2	Izbornik za postavljanje parametara eksperimenta unutar aplikacije FittsStudy	29
4.3	Primjer ispitnog zadatka unutar aplikacije FittsStudy	31
5.1	Profilni dijagram (eng. <i>Profile plot</i>) propusnosti	35
5.2	Profilni dijagram (eng. <i>Profile plot</i>) vremena izvršavanja zadatka	37
5.3	Grafička usporedba ocjene mentalne zahtjevnosti po modalitetima	38
5.4	Grafička usporedba ocjene fizičke zahtjevnosti po modalitetima	40
5.5	Grafička usporedba ocjene frustracije po modalitetima	41

Popis slika

5.6	Grafička usporedba subjektivne procijenjene učinkovitosti po modalitetima	42
5.7	Grafička usporedba ocjene truda po modalitetima	44
5.8	Grafička usporedba subjektivno procijenjene lakoće korištenja po modalitetima	45
5.9	Grafička usporedba procijene lakoće učenja po modalitetima	47
5.10	Grafička usporedba ocjena zadovoljstva po modalitetima	48
5.11	Odabir preferiranog modaliteta	50
5.12	Grafovi ovisnosti vremena izvršavanja zadatka o indeksu težine zadatka za modalitete 1, 2 i 3	52
5.13	Grafovi ovisnosti vremena izvršavanja zadatka o indeksu težine zadatka za modalitete 4, 5 i 6	53

Popis tablica

3.1	Modaliteti interakcije i njihove funkcijske značajke	15
-----	----------------------------------------------------------------	----

Popis programskih kodova

3.1	Proces slušanja	17
3.2	Proces prepoznavanja naredbi	18
3.3	Proces za zaustavljanje programa	20
3.4	Proces kontinuiranog pomicanja pokazivača	21
3.5	Funkcija za pomicanje pokazivača	21
3.6	Proces animacije rotirajućeg pokazivača	22
3.7	Funkcija za promjenu pokazivača	23

Poglavlje 1

Uvod

Interakcija čovjeka i računala (HCI, eng. *Human-Computer Interaction*)[1] je multidisciplinarno područje koje istražuje kako ljudi komuniciraju i surađuju s računalima i drugim digitalnim tehnologijama. Fokusira se na optimizaciju interakcije korisnika i računala dizajniranjem interaktivnih računalnih sučelja koja zadovoljavaju potrebe korisnika. Objedinjuje znanja iz područja kao što su računalna znanost, bihevioralne znanosti, kognitivna znanost, ergonomija, psihologija i principi dizajna. Interakcija čovjeka i računala je ključna u dizajniranju intuitivnih sučelja koja omogućavaju pristup osobama različitih sposobnosti i stručnosti posebno u zajednicama koje nemaju formalno obrazovanje za interakciju s određenim računalnim sustavima. Učinkovitim dizajnom sučelja su prilagođena korisnicima i osiguravaju da su interakcije jasne, precizne i prirodne.

U ovom radu, u želji da se predstavi alternativno upravljanje pokazivačem miša, istraženi su sustavi glasovnog upravljanja pokazivačem koji omogućavaju operacije pokazivanja-i-odabira. Prezentirano je šest modaliteta prilagođenih Windows operacijskom sustavu, s varijacijama u načinu pomaka i određivanja smjera pokazivača. Ovaj rad funkcionira kao *proof of concept* (dokaz koncepta), ilustrirajući mogućnost i održivost predloženih modaliteta.

U drugom poglavlju rada predstavljena je tehnologija prepoznavanja govora kroz sažeti osvrt. Treće poglavlje detaljno opisuje funkcionalnosti i implementacijske detalje modaliteta, predstavljajući pritom i vlastitu aplikaciju za testiranje te korištene

Poglavlje 1. Uvod

tehnologije. Četvrto poglavlje posvećeno je eksperimentu s ciljem usporedbe i evaluacije modaliteta. Osim toga, ukratko je predstavljen Fittsov zakon te njegova važnost u ocjenjivanju modaliteta. Ukupno 16 ispitanika sudjelovalo je u eksperimentu, tijekom kojeg su se vrednovala četiri odabrana modaliteta. Koristeći odabrane modalitete, ispitanici su rješavali zadatke unutar aplikacije FittsStudy i vlastite testne aplikacije. Nakon eksperimenta, ispitanicima su podijeljeni određeni upitnici kojima su ispitani subjektivni dojmovi o modalitetima. Peto poglavlje predstavlja statističku analizu podataka prikupljenih u eksperimentu, uspoređujući modalitete na osnovi objektivnih i subjektivnih kriterija.

Polazišna hipoteza jest da će modalitet s najkraćim i najjednostavnijim glasovnim naredbama biti najintuitivniji i najučinkovitiji. Pretpostavlja se da će kraće naredbe biti brže prepoznate, čime će se postići veća responzivnost modaliteta. S druge strane, složenost naredbe i samog funkcioniranja modaliteta može utjecati na učinkovitost, jer korisniku može biti potrebno više vremena da odabere odgovarajuću naredbu. Osim toga, složenije naredbe mogu povećati vjerojatnost pogrešne interpretacije od strane implementiranog sustava za prepoznavanje govora, čime se dodatno može umanjiti učinkovitost modaliteta.

Poglavlje 2

Prepoznavanje govora

Prepoznavanje govora je tehnologija koja omogućuje pretvaranje ljudskog govora u tekstualni format. Smatra se jednim od najsloženijih područja računalne znanosti koje uključuje lingvistiku, matematiku i statistiku. Prepoznavanje govora ima širok raspon primjena, uključujući digitalne asistente poput Siri i Google Assistant, pretraživanje glasom, automatizaciju telefonskih poziva i poboljšanje korisničkog iskustva.

2.1 Povijest tehnologije prepoznavanja govora

Povijest prepoznavanja govora [2] započinje s "Audrey", prvim službenim sustavom za prepoznavanje govora koji su 1950-ih godina dizajnirali Bell Laboratories. Audrey je bila sposobna prepoznati brojeve od 1 do 9 s impresivnom točnošću od 90%. Nakon toga, 1962. godine, IBM-ov "Shoebbox" donosi novi napredak omogućavajući prepoznavanje i razlikovanje 16 riječi. Međutim, i dalje je postojao izazov u prepoznavanju govora jer su raniji sustavi radili na osnovi zvučnih fragmenata (fonema), a korisnici su morali govoriti polako i praviti pauze kako bi osigurali ispravno prepoznavanje.

U ranim 1970-ima, američko Ministarstvo obrane prepoznalo je potencijal tehnologije prepoznavanja govora za vojne i obrambene aplikacije. Zbog toga su uložili pet godina u DARPA-in projekt istraživanja ove tehnologije. Kao rezultat, razvijen je "Harpy" sustav sposoban prepoznati više od 1000 riječi.

Tijekom 1980-ih godina, rječnik za prepoznavanje govora proširio se s nekoliko

Poglavlje 2. Prepoznavanje govora

stotina na nekoliko tisuća riječi. Jedan od ključnih proboja ostvaren je kroz primjenu statističke metode zvane Skriveni Markovljev Model (HMM, eng. *Hidden Markov Model*). Umjesto da se oslanja isključivo na riječi i traženje zvučnih uzoraka, HMM je procjenjivao vjerojatnost da su nepoznati zvukovi u stvarnosti riječi, čime je dodao novu razinu sofisticiranosti i točnosti u tehnologiji prepoznavanja govora.

Prepoznavanje govora napredovalo je 1990-ih velikim dijelom zahvaljujući razvoju osobnih računala koji su imali brže procesore. Do 2001. godine tehnologija prepoznavanja govora postigla je gotovo 80% točnosti.

U 2010. godini, Google je ostvario značajan napredak uvođenjem aplikacije Google Voice Search, koja je prilagođena prema glasu pojedinog korisnika i sposobna se prilagoditi njegovim govornim obrascima radi bolje točnosti. Ovaj iskorak omogućio je razvoj Siri 2011. godine, koja se istaknula svojom izvanrednom sposobnošću interpretacije prirodnih izraza i odgovaranja konverzacijskim stilom. Siri je usmjerila tehnologiju prepoznavanja govora prema vrhu tehnoloških inovacija.

2.2 Algoritmi za prepoznavanje govora

Sustavi za prepoznavanje govora složeni su sklopovi koji se sastoje od nekoliko komponenti, kao što su unos govora, izdvajanje značajki, vektori značajki, dekodirer i izlaz riječi. Unutar ovog procesa, dekodirer koristi akustične modele, rječnik izgovora i jezične modele kako bi odredio odgovarajući izlaz. Ocjenjivanje tehnologije prepoznavanja govora temelji se na točnosti, odnosno stopi pogreške u riječima (WER, eng. *Word Error Rate*) i brzini. Ta točnost može biti podložna različitim faktorima kao što su izgovor, naglasak, ton, glasnoća i pozadinska buka. Glavni cilj ovih sustava je postići razinu točnosti koja je usporediva s onom između dvoje ljudi koji komuniciraju, koristeći različite algoritme i tehnike računanja za prepoznavanje govora u tekst i poboljšanje točnosti transkripcije. U nastavku su kratka objašnjenja nekih od najčešće korištenih metoda [3]:

- Obrada prirodnog jezika (NLP, eng. *Natural Language Processing*) - Područje umjetne inteligencije koje se bavi interakcijom između računalnih algoritama i ljudskog jezika. Cilj obrade prirodnog jezika je omogućiti računalima razumi-

Poglavlje 2. Prepoznavanje govora

jevanje, interpretaciju i generiranje prirodnog jezika na način koji je i koristan i smislen za ljude.

- Skriveni Markovljevi modeli (HMM) - Temelje se na modelu Markovljevih lanaca, gdje vjerojatnost određenog stanja zavisi samo o trenutnom stanju, a ne o svim prethodnim stanjima. Dok je model Markovljevog lanca koristan za analizu opaženih događaja kao što je unos teksta, skriveni Markovljevi modeli pružaju mogućnost uključivanja nevidljivih ili skrivenih događaja, poput oznaka dijelova govora, u probabilistički model. U kontekstu prepoznavanja govora, koriste se kao modeli sekvenci, dodjeljujući oznake svakoj jedinici - bilo da su to riječi, slogovi, rečenice, itd. - unutar sekvence. Ove oznake omogućuju stvaranje mapiranja s određenim ulazom, što modelu dalje omogućuje određivanje najprikladnijeg niza oznaka.
- N-grami - Najjednostavniji tip jezičnog modela (LM, eng. *Language Model*), koji dodjeljuje vjerojatnosti rečenicama ili frazama. N-gram je slijed od N riječi. Gramatika i vjerojatnost određenih nizova riječi koriste se za poboljšanje prepoznavanja i točnosti.
- Neuronske mreže - Prvenstveno se koriste za algoritme dubokog učenja. Obrađuju podatke za obuku oponašajući povezanost ljudskog mozga putem slojeva čvorova. Svaki čvor se sastoji od ulaza, težina, pristranosti (ili praga) i izlaza, a ako izlazna vrijednost premaši zadani prag, čvor se "aktivira" i podaci se prenose sljedećem sloju u mreži. Proces učenja neuronske mreže odvija se putem nadziranog učenja, prilagođavajući se na temelju funkcije gubitka kroz postupak gradijentnog spusta. Iako te mreže imaju tendenciju biti preciznije i sposobne su prihvatiti veću količinu podataka, njihova učinkovitost izvođenja može biti manja u usporedbi s tradicionalnim jezičnim modelima, budući da su obično sporije za obuku.
- Dijarizacija govornika (SD, eng. *Speaker Diarization*) - Algoritmi dijarizacije govornika identificiraju i segmentiraju govor prema identitetu govornika. To pomaže aplikacijama da bolje razlikuju pojedince u razgovoru.

Poglavlje 3

Implementacija

Kao platforma za implementaciju potrebnih aplikacija odabran je operacijski sustav tvrtke Microsoft, Windows 11. Modaliteti interakcije zasnovani na tehnologiji upravljanja govorom implementirani su u programskom jeziku Python [4]. Python je također korišten za izradu jednostavne aplikacije za testiranje modaliteta.

3.1 Korištene tehnologije

U ovom potpoglavlju opisuje se tehnološki stog koji je korišten tijekom izrade spomenutih aplikacija.

3.1.1 Python

Python je programski jezik visoke razine i opće namjene, koji je kreirao Guido van Rossum 1990. godine. Svojim automatskim upravljanjem memorijom, slični jezicima kao što su Perl, Ruby i Smalltalk. Ono što Python čini posebno privlačnim programerima je njegova fleksibilnost u omogućavanju različitih stilova programiranja, uključujući objektno orijentirano, strukturalno i aspektno orijentirano programiranje.

U nastavku je dan kratak pregled Python knjižnica koje su bile korištene tijekom razvoja modaliteta i aplikacije za njihovo testiranje.

Knjižnica `SpeechRecognition`

Za implementaciju prepoznavanja govora u svim modalitetima korištena je knjižnica `SpeechRecognition` [5]. Ona omogućuje jednostavno prepoznavanje govora i pretvaranje izgovorenog teksta u pisani format. Razvijena je kako bi omogućila programerima jednostavno sučelje za rad s različitim tehnologijama prepoznavanja govora. `SpeechRecognition` radi tako da omogućuje programerima pristup više različitih pokretača za prepoznavanje govora, kao što su Google Web Speech API, Microsoft Bing Voice Recognition, IBM Speech to Text, itd.

Ova knjižnica uzima audio zapis kao ulaz i obrađuje ga kroz odabrani pokretač, koji zatim analizira govor u zapisu, prepoznaje riječi i vraća transkript kao izlaz. Proces se sastoji od nekoliko koraka. Prvo, audio zapis se mora snimiti ili učitati, a zatim se mora odabrati pokretač koji će ga obraditi. Pokretač će zatim izvesti niz algoritama za obradu signala kako bi izvukao značajke govora, poput frekvencije, intenziteta i tempa. Nakon što se izvuku značajke, koriste se statistički modeli, kao što su skriveni Markovljevi modeli ili duboke neuronske mreže, kako bi se prepoznale riječi unutar signala.

Knjižnica `SpeechRecognition` također pruža funkcionalnosti za upravljanje šumom i akcentima te prilagodbu specifičnim domenama ili rječnicima. S obzirom na različite pokretače i metode koje nudi, moguće je podesiti prepoznavanje za različite jezike, naglaske i uvjete okoline.

Knjižnica `multiprocessing`

Modaliteti interakcije su implementirani na način da koriste višeprocesiranje (eng. `multiprocessing`). To je postignuto koristeći `multiprocessing` [6] knjižnicu. Knjižnica `multiprocessing` omogućuje jednostavno paralelno izvršavanje zadataka na više procesora ili jezgri procesora. Olakšava iskorištavanje višejezgrenih kapaciteta suvremenih računalnih sustava, potiče ubrzanje izvođenja programa i omogućuje efikasno rješavanje zadataka koji zahtijevaju intenzivnu računalnu obradu. Korištenjem višeprocesiranja, programi mogu istovremeno izvršavati funkcije ili metode, dijeliti podatke između procesa, izbjevati blokirajuće operacije i poboljšati performanse

Poglavlje 3. Implementacija

za zadatke koji su teški za izvršavanje serijski.

Knjižnica mouse

Operacije pokazivanja-i-odabira unutar implementiranih modela izvedene su korištenjem Pythonove knjižnice po imenu `mouse` [7]. Ova knjižnica nudi programerima mogućnost kontrole pokazivača miša izravno iz Python programa, čime se otvara širok spektar mogućnosti za simulaciju različitih aktivnosti miša, uključujući pokrete pokazivača, pritiske tipki, scrollanje i slično. Primjenom knjižnice `mouse`, programeri mogu automatizirati interakciju s grafičkim korisničkim sučeljima ili provoditi testiranje aplikacija koje ovise o upravljanju mišem.

Knjižnica keyboard

Budući da se modaliteti deaktiviraju pritiskom na tipku "q", korištena je knjižnica `keyboard` [8]. Ova knjižnica omogućuje programerima kontrolu i praćenje događaja uzrokovanih tipkovnicom, pružajući funkcije za simulaciju pritiska i otpuštanja tipki, kao i za slušanje globalnih prečaca ili individualnih događaja tipki. Može se koristiti na različitim operacijskim sustavima poput Windowsa, Linuxa i macOS-a.

Knjižnica ctypes

Promjena sistemskog kursora unutar određenih modaliteta postignuta je korištenjem knjižnice `ctypes` [9]. Ova knjižnica nudi moćno i fleksibilno sučelje za pozivanje funkcija u zajedničkim knjižnicama i DLL-ovima (eng. *Dynamic Link Libraries*) koristeći izvornu C konvenciju poziva. Posebno je korisna kada je potrebno integrirati programski kod u Pythonu s prethodno napisanim C kodom ili kada je potrebno koristiti sustavske API funkcije koje Python ne nudi izravno. Kroz `ctypes`, mogu se učitati C-kompatibilne binarne knjižnice, pozivati funkcije, definirati i instancirati strukture podataka, kao i definirati povratne funkcije. Na taj način, knjižnica gradi most između visokorazinske programske logike Pythona i izravne učinkovitosti C jezika.

Knjižnica tkinter

Vlastita aplikacija za testiranje modaliteta interakcije izrađena je u Python knjižnici `tkinter`[10], koja pruža intuitivan način stvaranja grafičkog korisničkog sučelja (GUI, eng. *Graphical User Interface*). Kao standardna GUI knjižnica u Pythonu, već je integrirana unutar Python instalacije. Nudi razne widgete poput gumba, okvira, polja za unos i mnoge druge, olakšavajući izradu interaktivnih aplikacija. Uz podršku za različite upravitelje rasporeda, elementi se mogu jednostavno organizirati i smjestiti unutar prozora.

3.1.2 Vosk

Za prepoznavanje govora korišten je Vosk [11]. Vosk je otvorena *offline* platforma za prepoznavanje govora. Podržava više od 20 jezika i dijalekata, uključujući engleski, njemački, francuski i španjolski. Njegovi modeli nude kontinuiranu transkripciju s obimnim rječnikom, brzu povratnu informaciju putem API-ja, prilagodljivost rječnika i prepoznavanje govornika. Temelji se na tehnologiji dubokog učenja, točnije na dubokim neuronskim mrežama (DNN, eng. *Deep Neural Networks*). Konkretnije, za svoju osnovu koristi Kaldi, popularni okvir otvorenog koda (eng. *open source framework*) za prepoznavanje govora. Kaldi pruža potrebne alate i funkcionalnosti za obuku modela dubokih neuronskih mreža i izvođenje automatskog prepoznavanja govora (ASR, eng. *Automatic Speech Recognition*). Vosk je kompatibilan s raznim programskim jezicima poput Pythona, Java, C++ i Rusta te je optimiziran za širok spektar uređaja, od Raspberry Pi-a, preko Android telefona, do velikih računalnih grozdova.

3.1.3 Axialis CursorWorkshop

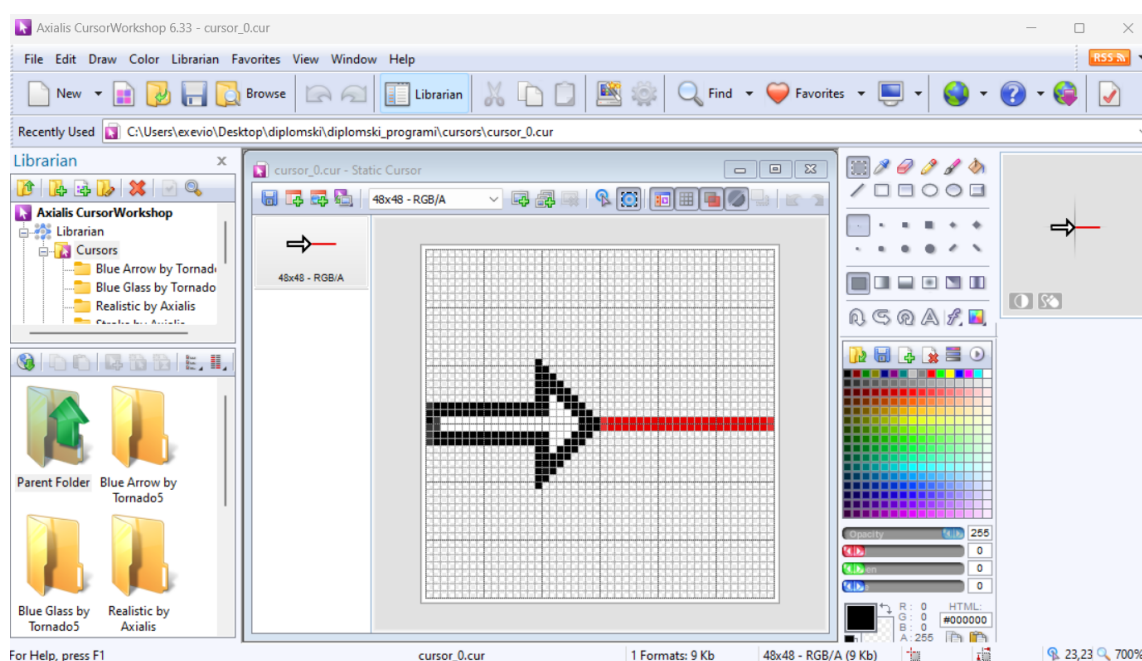
Za potrebe modaliteta koji implementiraju određivanje smjera kretanja pokazivača s podesivim kutem kretanja dizajniran je posebni pokazivač. To je ostvareno pomoću alata Axialis CursorWorkshop [12].

Axialis CursorWorkshop namijenjen je za stvaranje i uređivanje pokazivača miša. Omogućuje korisnicima da stvaraju prilagođene pokazivače za Windows, kompletno

Poglavlje 3. Implementacija

s animacijama i raznim efektima. Korisnici mogu stvarati pokazivače iz slika, raditi sa slojevima i koristiti širok raspon alata kako bi precizno uredili svoje dizajne.

Posebni pokazivač miša dizajniran je kao strelica s crvenom zrakom koja izbija iz nje, omogućujući korisniku da jasnije vidi smjer kretanja pokazivača. Ukupno je kreirano 36 različitih verzija ovog pokazivača, svaka za kutove koji se kreću od 0° do 360° , s intervalom od 10° . Slika 3.1 prikazuje početnu verziju ovog posebnog pokazivača (kut od 0°) unutar Axialis CursorWorkshop alata.



Slika 3.1 Dizajn pokazivača miša unutar aplikacije CursorWorkshop

3.2 Modeliranje sustava za glasovno upravljanje pokazivačem

Sveukupno je implementirano 6 različitih modaliteta za glasovno upravljanje pokazivačem. Svi modaliteti zasnovani su na jednostavnim glasovnim naredbama koje se sastoje od jedne ili više riječi. Odabrani jezik na kojem se izgovaraju naredbe je en-

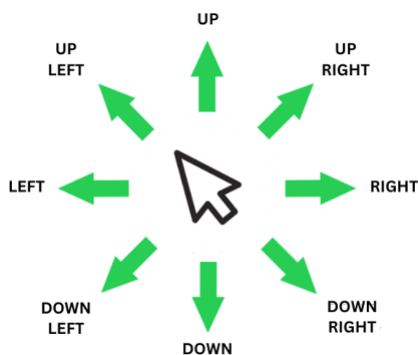
gleski. Modaliteti se razlikuju prema načinu pomaka pokazivača i načinu određivanja smjera kretanja pokazivača.

3.2.1 Funkcijske značajke modaliteta za glasovno upravljanje pokazivačem

Svakim modalitetom može se upravljati pokazivačem putem kontinuiranog ili diskretnog pomaka. U kontinuiranom načinu rada, pokazivač se pokreće jednolikom brzinom od 133,33 piksela po sekundi u odabranom smjeru na određenu naredbu, sve dok sustav ne registrira naredbu za zaustavljanje kretanja. Brzina ovog pomaka nije odabrana na temelju kvantitativnih mjerenja, već je postavljena subjektivno, temeljem osobnih dojmova i iskustva s performansama sustava. To znači da odabir brzine nije bio vođen objektivnim ili mjerljivim kriterijima. S druge strane, diskretni pomak omogućava korisniku da izrazi željeni broj piksela za pomak u određenom smjeru, nakon čega se pokazivač precizno pomakne za odgovarajuću udaljenost.

Određivanja smjera kretanja pokazivača implementirano je na 3 različita načina koji su objašnjeni u nastavku:

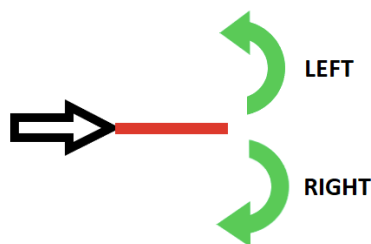
1. Bez direktnog određivanja kuta - Ovaj pristup omogućava korisniku da odredi smjer pokazivača izgovaranjem jedne od naredbi prikazanih na slici 3.2. Pokazivač se tada može pomicati u jednom od 8 mogućih smjerova, gdje svaki smjer odgovara jediničnom kutu od 45 stupnjeva.



Slika 3.2 Grafički prikaz određivanja smjera pokazivača bez direktnog određivanja kuta

Poglavlje 3. Implementacija

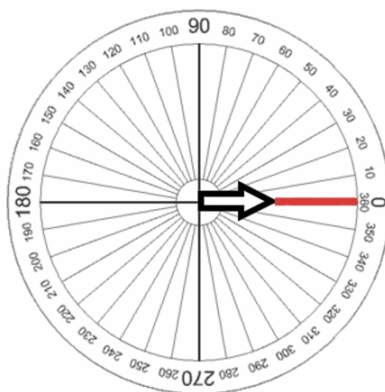
2. Bez direktnog određivanja kuta, koristeći animirani pokazivač - Na početku, specijalizirani pokazivač ukazuje u smjeru koji odgovara kutu od 0 stupnjeva (slika 3.3). Ako korisnik izgovori naredbu "left", pokazivač započinje rotaciju u smjeru suprotnom od kretanja kazaljke na satu, čime se pokreće animacija. Suprotno tome, naredba "right" uzrokuje rotaciju pokazivača u smjeru kazaljke na satu. Animacija se nastavlja sve dok se ne izgovori naredba koja je zaustavlja. S obzirom na to da postoji 36 različitih verzija posebnog pokazivača, mogući smjerovi kretanja su raspoređeni u redovitim intervalima od 10 stupnjeva.



Slika 3.3 Grafički prikaz određivanja smjera pokazivača bez direktnog određivanja kuta koristeći animirani pokazivač

3. S direktnim određivanjem kuta - Ovaj pristup omogućuje korisniku da izgovori željeni kut, koji se zatim prilagodi kako bi bio broj između 0 i 360, djeljiv s 10. Na temelju toga, prilagođeni pokazivač zamjenjuje se verzijom koja pokazuje u točnom smjeru. Na primjer, ako korisnik izgovori "ninety three", program će to interpretirati kao 90, a pokazivač će se zamijeniti onim koji pokazuje u smjeru koji odgovara kutu od 90 stupnjeva.

Poglavlje 3. Implementacija



Slika 3.4 Grafički prikaz određivanja smjera pokazivača s direktnim određivanjem kuta

Popis svih modaliteta i njihovih funkcijskih značajki dan je u nastavku.

Modalitet 1

Prvi modalitet kombinacija je kontinuiranog pomaka pokazivača i određivanja smjera bez direktnog određivanja kuta. Naredbe koje ovaj modalitet prepoznaje su: "up", "down", "left", "right", "up left", "up right", "down left", "down right", "stop", "click". Izgovaranjem jedne od prvih osam navedenih naredbi, pokazivač započinje pomicanje jednolikom brzinom u zadanom smjeru (slika 3.2). Naredba "stop" zaustavlja pokret pokazivača, dok naredba "click" simulira pritisak lijeve tipke miša.

Modalitet 2

Modalitet 2 kombinacija je diskretnog pomaka pokazivača i određivanja smjera bez direktnog određivanja kuta. Naredbe koje ovaj modalitet prepoznaje su: "up", "down", "left", "right", "up left", "up right", "down left", "down right", "click". Prvih 8 navedenih naredbi potrebno je izgovoriti uz broj piksela za koje se korisnik želi pomaknuti u odabranom smjeru. Na primjer, izgovaranjem naredbe "up hundred" pokazivač miša će se pokrenuti sto piksela prema gore od trenutne pozicije. Kao i u prvom modalitetu, naredba "click" simulira pritisak lijeve tipke miša.

Modalitet 3

Ovaj modalitet kombinacija je kontinuiranog pomaka pokazivača i određivanja smjera s direktnim određivanjem kuta. Prepoznaje glasovne naredbe kao što su "go", "stop", "click" te broj koji predstavlja kut. Na primjer, ako korisnik izgovori "one hundred and eighty", modalitet će to interpretirati kao kut od 180 stupnjeva te zamijeniti trenutni pokazivač s onim koji pokazuje u smjeru koji odgovara navedenom kutu. Izgovaranjem naredbe "go" pokazivač započinje kretanje u odabranom smjeru konstantnom brzinom, sve dok korisnik ne zaustavi kretanje izgovaranjem naredbe "stop". Izgovaranjem naredbe "click" simulira se pritisak lijeve tipke miša.

Modalitet 4

Modalitet 4 kombinacija je diskretnog pomaka pokazivača i određivanja smjera s direktnim određivanjem kuta. Naredbe koje prepoznaje su "click", broj koji predstavlja kut i "go" + broj piksela. Na primjer, ako korisnik izgovori "ninety", modalitet će to shvatiti kao kut od 90 stupnjeva te prilagoditi pokazivač da ukazuje u odgovarajućem smjeru. Ako korisnik potom kaže "go two hundred", pokazivač će se pokrenuti za 200 piksela u prethodno odabranom smjeru. Kao i u prethodnim modalitetima, naredbom "click" simulira se pritisak lijeve tipke miša.

Modalitet 5

Peti modalitet je kombinacija kontinuiranog pomaka pokazivača i određivanja smjera bez direktnog određivanja kuta, koristeći animirani pokazivač. Ovaj modalitet reagira na naredbe "go", "stop", "left", "right", i "click". Naredbama "left" i "right" pokreće se animacija rotacije pokazivača; rotira se u smjeru kazaljke na satu za "right" i suprotno od toga za "left". Rotacija se prekida naredbom "stop", čime se odabire smjer kretanja pokazivača. Kretanje pokazivača započinje izgovorom naredbe "go" i zaustavlja se naredbom "stop". Slično ostalim modalitetima, naredba "click" oponaša pritisak lijeve tipke miša.

Modalitet 6

Zadnji modalitet je kombinacija diskretnog pomaka pokazivača i određivanja smjera bez direktnog određivanja kuta, koristeći animirani pokazivač. Ovaj modalitet reagira na glasovne naredbe kao što su "stop", "left", "right", "click", te broj piksela. Animacija rotacije pokazivača pokreće se naredbama "left" i "right", jednako kao u petom modalitetu, a zaustavlja se naredbom "stop". Nakon što je smjer pokazivača odabran, korisnik može specificirati broj piksela za koliko želi da se pokazivač pomakne u tom smjeru. Naredba "click", sukladno prethodnim modalitetima, simulira pritisak lijeve tipke miša.

Tablica 3.1 prikazuje sve implementirane modalitete interakcije i njihove pripadajuće funkcijske značajke.

Tablica 3.1 Modaliteti interakcije i njihove funkcijske značajke

Modalitet	Način pomaka pokazivača	Određivanje smjera kretanja pokazivača
1	Kontinuirani	Bez direktnog određivanja kuta
2	Diskretni	Bez direktnog određivanja kuta
3	Kontinuirani	S direktnim određivanjem kuta
4	Diskretni	S direktnim određivanjem kuta
5	Kontinuirani	Bez direktnog određivanja kuta, koristeći animirani pokazivač
6	Diskretni	Bez direktnog određivanja kuta, koristeći animirani pokazivač

3.2.2 Implementacijski detalji modaliteta za glasovno upravljanje pokazivačem

Svaki modalitet osmišljen je kao program koji kombinira više paralelnih procesa, koristeći Pythonovu knjižnicu `multiprocessing`. Ključna komponenta ove knjižnice jest razred `Manager`, koji omogućuje komunikaciju i dijeljenje stanja između

Poglavlje 3. Implementacija

različitih procesa. Unutar svakog modaliteta postoje varijable koje se dijele između odgovarajućih procesa, a to su:

- `audio_queue` - Red za pohranjivanje audio zapisa s ograničenim kapacitetom od tri elementa.
- `exit_program` - Booleova varijabla koja kontrolira prekid rada svih procesa. Njena inicijalna vrijednost je `False`, a postavljanje na `True` signalizira svim procesima da treba prekinuti rad.
- `command` - Znakovni niz (*String*) koji se koristi za pohranu trenutne glasovne naredbe.
- `angle` - Cjelobrojna vrijednost koja se koristi za pohranu kuta.

Svaki modalitet sadrži proces zadužen za slušanje i pohranjivanje audio zapisa, proces za prepoznavanje glasovnih naredbi, te proces koji omogućuje zaustavljanje programa. Određeni modaliteti mogu dodatno uključivati procese za kontinuirano pomicanje kursora ili animaciju specifičnog pokazivača. S ciljem modularnosti, svi ovi procesi smješteni su u zasebnu datoteku naziva `shared_processes.py`. Detaljna objašnjenja svih procesa nalaze se u nastavku.

Proces slušanja

Proces slušanja realiziran je unutar funkcije `listen`, koja je prikazana u programskom isječku 3.1. Ova funkcija se oslanja na dva ključna razreda iz knjižnice `SpeechRecognition`. Razred `Recognizer` omogućuje obradu i prevođenje audio zapisa u tekst. S druge strane, razred `Microphone` pruža funkcionalnost za rad s mikrofonom uređaja i snimanje zvuka.

Funkcija `listen` koristi tri parametra: `audio_queue`, `phrase_time_limit` i `exit_program`. Parametar `phrase_time_limit` predstavlja vremenski okvir za snimanje određene fraze, izražen u sekundama. Dok neki modaliteti, koji ne trebaju prepoznavati brojeve, ovaj parametar postavljaju na 1 sekundu, ostali ga definiraju na 3 sekunde. U funkciji se mikrofoni (`mic`) postavljaju kao primarni izvor zvuka, a prepoznavatelj (`r`) se kalibrira za bolje razlikovanje govora od okolne buke. Korisniku se potom daje obavijest da može započeti s govorom. U kontinuiranoj petlji, funkcija

Poglavlje 3. Implementacija

provjerava stanje varijable `exit_program`. Ako je postavljena na `True`, slušanje se zaustavlja uz odgovarajuću obavijest korisniku. U suprotnom, prepoznavač neprestano snima govor prema postavljenom vremenskom ograničenju. Snimljeni audio zapis pokušava se dodati u `audio_queue`. U slučaju da je red pun, korisniku se prikazuje upozoravajuća poruka.

```
1 def listen(audio_queue, phrase_time_limit, exit_program):
2     with mic as source:
3         r.adjust_for_ambient_noise(source)
4         print("Please start speaking.")
5         while True:
6             if exit_program.value:
7                 print("Exit listen process")
8                 break
9             audio = r.listen(source, phrase_time_limit=
phrase_time_limit)
10            try:
11                audio_queue.put_nowait(audio)
12            except queue.Full:
13                print("Audio queue is full")
```

Programski kod 3.1 Proces slušanja

Proces za prepoznavanje naredbi

Proces za prepoznavanje naredbi definiran je funkcijom `keyword_detection` (Isječak programskog koda 3.2). Njegova primarna svrha je analizirati audio snimke nastale tijekom procesa slušanja, prepoznati glasovne naredbe i izvršavati relevantne akcije temeljene na njima. Funkcija `keyword_detection` prima sljedeće parametre: `mode_processor`, `keywords`, `audio_queue`, `command`, `angle` i `exit_program`. Parametar `mode_processor` predstavlja funkcionalnost koja se mijenja ovisno o odabranom modalitetu, svodeći se na skup radnji karakterističnih za taj modalitet, kao što su diskretni pomak kursora ili promjena verzije posebnog kursora. Parametar `keywords` predstavlja mapu prethodno definiranih glasovnih naredbi koje sustav prepoznaje.

Funkcija neprestano prati audio ulaz kroz kontinuiranu *while* petlju koja započinje na liniji 21. Pri svakoj iteraciji petlje, prvo se provjerava status varijable

Poglavlje 3. Implementacija

`exit_program`. Ako je njezina vrijednost postavljena na `True`, proces praćenja audio ulaza prekida se uz odgovarajuće obavještenje upućeno korisniku. Nadalje, unutar petlje, sustav neposredno pokušava izvući audio snimke iz `audio_queue` varijable. Ako nema podataka u redu, iteracija se nastavlja bez dodatnog zastoja. Nakon uspješnog dohvaćanja audio podataka, oni se obrađuju pomoću `SpeechRecognition` knjižnice, koristeći prepoznavač (`r`) i odgovarajući Vosk model te se konvertiraju u tekst. Sustav potom uspoređuje prepoznati tekst s predefiniranom `keywords` mapom radi identifikacije podudaranja. Dodatno, pomoću funkcije `word_to_num` iz `word2number` knjižnice, provjerava se odgovara li izgovorena riječ nekoj numeričkoj vrijednosti. Na temelju ove analize, ažuriraju se varijable `command` i `num_value`. Prepoznata glasovna naredba potom aktivira ugniježđenu funkciju `process_command` kako bi se provele odgovarajuće akcije. Ako se akcije uspješno izvedu, korisniku se kao povratna informacija ispisuje glasovna naredba koju je sustav prepoznao na temelju dobivenog teksta. U slučajevima kad dođe do greške ili iznimke, korisnika se obavještava da ponovi naredbu.

Funkcija `process_command` (linija 2) služi kao mehanizam izravnog odgovora nakon što se detektira glasovna naredba. Ako sustav registrira naredbu "click", simulira se lijevi klik miša. S druge strane, za naredbe koje nisu "stop", usmjerava ih kroz `mode_processor` radi izvođenja specifičnih akcija ili prilagodbi zasnovanih na modalitetu koji upravlja ovim procesom. Funkcija također ima odredbe za ažuriranje vrijednosti varijabli `angle` i `command` ako `mode_processor` preporučuje promjene.

```
1 def keyword_detection(mode_processor, keywords, audio_queue, command,
2   angle, exit_program):
3     def process_command():
4         command_val = command.value
5         if command_val == "click":
6             mouse.click("left")
7             command.value = "stop"
8             return "click"
9         elif command_val != "stop":
10            mode_processor_args = {
11                "command": command_val,
12                "angle": angle.value,
13                "num": num_value,
14            }
15            new_command_val, new_angle_val, cmd_output = mode_processor(
```

Poglavlje 3. Implementacija

```
mode_processor_args)
15
16     if new_angle_val is not None: angle.value = new_angle_val
17     if new_command_val is not None: command.value =
new_command_val
18     return cmd_output
19     return command_val
20
21 while True:
22     if exit_program and exit_program.value:
23         print("Exit keyword detection process")
24         break
25     try: audio = audio_queue.get_nowait()
26     except queue.Empty: continue
27
28     try:
29         text = json.loads(r.recognize_vosk(audio)).get("text", "").
lower()
30         if(text != ""):
31             print(f"text: {text}")
32             matching_keywords = set(text.split()).intersection(
keywords)
33             num_value = None
34
35             try: num_value = w2n.word_to_num(text)
36             except: pass
37
38             if matching_keywords: command.value = keywords[next(
iter(matching_keywords))]
39             elif num_value is not None: command.value = None
40             else: continue
41
42             cmd_output = process_command()
43             if cmd_output: print(f"Command: {cmd_output}")
44
45     except Exception:
46         if not exit_program or not exit_program.value:
47             print("Please speak again.")
```

Programski kod 3.2 Proces prepoznavanja naredbi

Proces zaustavljanja programa

Funkcija `check_key_press` (Isječak programskog koda 3.3) korištena je za definiranje procesa koji služi za zaustavljanje programa i svih procesa koje program implementira. Funkcija prima samo jedan parametar: `exit_program`. Dok se program izvršava, ovaj proces će slušati pritiske tipki i kad detektira da je tipka "q" pritisnuta, ispisat će poruku "Exiting the program" i postaviti će vrijednost varijable `exit_program` na `True`. Ova promjena vrijednosti signalizira ostalim procesima da trebaju zaustaviti izvršavanje, omogućujući korisniku da izađe iz programa jednostavnim pritiskom na tipku "q".

```
1 def check_key_press(exit_program):
2     while True:
3         if keyboard.is_pressed("q"):
4             print("Exiting the program")
5             exit_program.value = True
6             break
```

Programski kod 3.3 Proces za zaustavljanje programa

Proces kontinuiranog pomicanja pokazivača

Kod modaliteta koji implementiraju kontinuirani pomak pokazivača potrebno je implementirati proces koji će to obavljati. Proces kontinuiranog pomaka pokazivača definiran je funkcijom `mouse_movement` (Isječak programskog koda 3.4), koja pokreće miš na temelju detektiranih glasovnih naredbi. Parametri koji su definirani za ovu funkciju su: `cursor_move_keyword`, `cursor_step_size`, `command`, `angle`, `exit_program`. Parametar `cursor_move_keyword` je lista ključnih riječi na koje započinje kontinuirano pomicanje pokazivača. Ta lista je specifična za svaki modalitet. Parametar `cursor_step_size` definira broj piksela za koji se pokazivač pomiče u određenom smjeru.

Unutar beskonačne petlje, funkcija prvo provjerava je li vrijednost `exit_program` postavljena, i ako je istinita, izlazi iz procesa kretanja miša, obavještavajući korisnika putem ispisane poruke. Ako se vrijednost `command` nalazi unutar varijable `cursor_move_keywords`, što ukazuje na detektiranu glasovnu naredbu za pomicanje miša, tada poziva funkciju `move_mouse_cursor`. Ta funkcija prima unaprijed

Poglavlje 3. Implementacija

definiranu veličinu koraka za kretanje (`cursor_step_size`), smjer (`angle.value`) i trajanje pomaka kako bi precizno pomaknula pokazivač miša po ekranu. Trajanje pomaka izračunato je dijeljenjem veličine koraka s konstantom `CURSOR_SPEED` koja označava brzinu kretanja u pikselima po sekundi.

```
1 def mouse_movement(cursor_move_keywords, cursor_step_size, command,
2   angle, exit_program):
3     while True:
4         if exit_program.value:
5             print("Exit mouse movement process")
6             break
7         elif command.value in cursor_move_keywords:
8             move_mouse_cursor(cursor_step_size, angle.value,
9                               cursor_step_size / CURSOR_SPEED)
```

Programski kod 3.4 Proces kontinuiranog pomicanja pokazivača

Funkcija `move_mouse_cursor` (Isječak programskog koda 3.5) upravlja kretanjem pokazivača miša na ekranu. Na temelju zadane veličine koraka (`cursor_step_size`), kuta (`angle`) i trajanja (`duration`), funkcija izračunava horizontalni (`x`) i vertikalni (`y`) pomak miša koristeći trigonometrijske funkcije iz Python knjižnice `math`. Nakon izračuna `x` i `y` vrijednosti, funkcija `move` iz knjižnice `mouse` se koristi za izvršenje samog pomaka miša na zaslonu. Važno je napomenuti da se za vertikalni pomak (`y`) koristi negativna vrijednost kako bi se osigurao ispravan smjer kretanja u skladu s koordinatnim sustavom zaslona.

```
1 def move_mouse_cursor(cursor_step_size, angle, duration):
2     x = cursor_step_size * math.cos(math.radians(angle))
3     y = cursor_step_size * math.sin(math.radians(angle))
4     mouse.move(x, -y, absolute=False, duration=duration)
```

Programski kod 3.5 Funkcija za pomicanje pokazivača

Proces animacije rotirajućeg pokazivača

Funkcijom `cursor_change` (Isječak programskog koda 3.6) definira se proces animacije rotirajućeg pokazivača. Kao i kod prethodnih procesa, ova funkcija implementira beskonačnu petlju. Unutar petlje neprekidno se prati varijabla `command`. Ako varijabla sadrži vrijednosti "left" ili "right", kut pokazivača mijenja se za 10 stupnjeva

Poglavlje 3. Implementacija

u odgovarajućem smjeru. Specifično, ako je naredba "left", kut se povećava za 10 stupnjeva, a ako je "right", smanjuje za 10 stupnjeva. Ova promjena kuta se zatim primjenjuje putem funkcije `set_custom_cursor`, koja prilagođava izgled pokazivača. Također postoje uvjeti koji osiguravaju da kut ostane unutar raspona od 0 do 350 stupnjeva, čime se izbjegava prekoračenje. Petlja se izvodi svakih 0,5 sekundi dok se ne postavi signal za izlazak iz programa putem varijable `exit_program`, što zatim prekida petlju i završava proces.

```
1 def cursor_change(command, angle, exit_program):
2     while True:
3         if exit_program.value:
4             print("Cursor change exit program")
5             break
6         if command.value == "left" or command.value == "right":
7             if command.value == "left":
8                 angle.value += 10
9             else:
10                angle.value -= 10
11
12                if angle.value == -10:
13                    angle.value = 350
14                if angle.value == 360:
15                    angle.value = 0
16
17                set_custom_cursor(angle.value)
18
19                time.sleep(0.5)
```

Programski kod 3.6 Proces animacije rotirajućeg pokazivača

Funkcija `set_custom_cursor` (Isječak programskog koda 3.7) prilagođava izgled pokazivača temeljen na specificiranom kutu. Parametar `angle` koji se prenosi funkciji koristi se za odabir odgovarajuće datoteke pokazivača, čija su imena oblikovana tako da odgovaraju različitim kutovima (npr., "cursor_0.cur", "cursor_10.cur" itd.).

Prvo, funkcija poziva `LoadImage` funkciju iz `win32gui` knjižnice kako bi učitala odgovarajuću datoteku pokazivača s diska. Ova metoda vraća identifikator pokazivača koji se zatim koristi u pozivu `ctypes.windll.user32.SetSystemCursor` funkcije kako bi se postavio kao trenutni pokazivač sustava. Drugi parametar u ovom pozivu, 32512, standardni je identifikator za normalni selektivni kursor. Nakon što

Poglavlje 3. Implementacija

je pokazivač postavljen, zadnji poziv `ctypes.windll.user32.DestroyCursor` oslobađa resurse povezane s pokazivačem koji je upravo bio postavljen. Ovime se osigurava da se resursi oslobode nakon što se pokazivač više ne koristi, čime se izbjegava mogući gubitak memorije ili drugi problemi sa sustavom.

```
1 def set_custom_cursor(angle):
2     custom_cursor = win32gui.LoadImage(
3         0,
4         "cursors/cursor_{}.cur".format(angle),
5         win32con.IMAGE_CURSOR,
6         0,
7         0,
8         win32con.LR_LOADFROMFILE,
9     )
10    ctypes.windll.user32.SetSystemCursor(custom_cursor, 32512)
11    ctypes.windll.user32.DestroyCursor(custom_cursor)
```

Programski kod 3.7 Funkcija za promjenu pokazivača

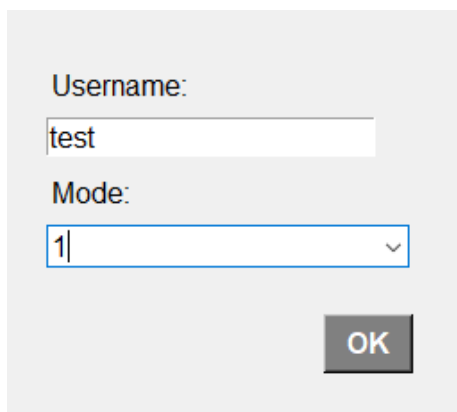
3.3 Aplikacija za testiranje modaliteta

Da bi se procijenilo vrijeme potrebno za obavljanje tipičnog zadatka s mišem, u kontekstu izvršavanja operacije pokazivanja-i-odabira u realnom okruženju, razvijena je vlastita testna aplikacija. Zadatak unutar aplikacije uključuje selekciju pet interaktivnih meta različitih veličina i položaja na zaslonu u određenom redoslijedu. Aplikacija koristi grafičko korisničko sučelje izrađeno pomoću Python knjižnice `tkinter`.

Korisnik započinje test unosom svog korisničkog imena te odabirom željenog modaliteta iz padajućeg izbornika (Slika 3.5). Nakon potvrde, pojavljuje se gumb "Press here to start". Pritiskom na njega započinje zadatak (Slika 3.6) u kojem korisnik mora selektirati mete u predefiniranom redoslijedu. Kada korisnik pritisne gumb u točnom redoslijedu, gumb postaje zelene boje, dok će postati crvene boje ako redoslijed nije točan. Ako korisnik uspješno pritisne sve gumbe u ispravnom redoslijedu, program izračunava koliko je vremena proteklo od početka zadatka i sprema rezultate, zajedno s korisničkim imenom i odabranim načinom rada, u CSV datoteku

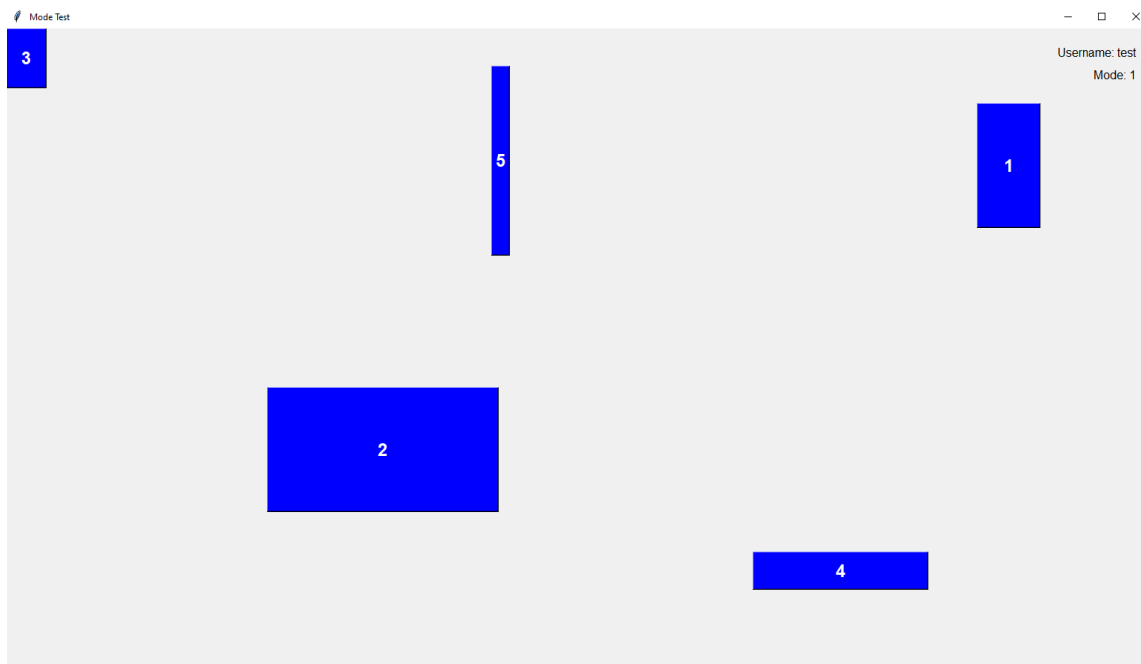
Poglavlje 3. Implementacija

mode_testing_results.csv. Nakon što korisnik završi s trenutnim zadatkom, može ga ponoviti unosom svog imena i novim odabirom modaliteta.



A screenshot of a configuration dialog box. It has a light gray background. At the top, it says "Username:" followed by a text input field containing the word "test". Below that, it says "Mode:" followed by a dropdown menu showing the number "1" and a small downward arrow. At the bottom right, there is a dark gray button with the text "OK" in white.

Slika 3.5 Izbornik za postavljanje parametara unutar vlastite aplikacije za testiranje



Slika 3.6 Sučelje vlastite aplikacije za testiranje modaliteta interakcije

Poglavlje 4

Eksperimentalno vrednovanje

Cilj ovog rada je procijeniti i usporediti prethodno navedene modalitete glasovnog upravljanja pokazivačem prilikom izvođenja operacije pokazivanja-i-odabira na osobnom računalu. Također će se razmotriti i primjena Fittsovog zakona na takve sustave glasovnog upravljanja. Za eksperimentalnu analizu odabrana su prva četiri modaliteta, kako bi se izbjeglo predugo trajanja eksperimenta. Svaki modalitet je pojedinačno evaluiran kroz niz definiranih zadataka. Učinkovitost je mjerena vremenom potrebnim za izvršenje zadatka te propusnošću utemeljenom na Fittsovom zakonu.

4.1 Fittsov zakon

Fittsov zakon [13] prediktivni je model ljudskog pokreta. Primarno se koristi u području interakcije čovjeka i računala i u ergonomiji. Zakon predviđa da je vrijeme potrebno za brzo premještanje u ciljano područje funkcija omjera između udaljenosti do cilja i širine cilja. Koristi se za modeliranje akcije pokazivanja, bilo da je riječ o fizičkom pokazivanju rukom ili virtualnom pokazivanju na ekranu. U praksi se pokazalo da Fittsov zakon vrijedi u različitim uvjetima; koristeći različite udove, različite ulazne uređaje i u različitim fizičkim okolinama i populacijama korisnika.

Originalni rad iz 1954. godine autora Paula Morrisa Fittsa uveo je metriku kojom se može izmjeriti složenost zadataka pri odabiru cilja. Ova metrika bila je osmišljena na temelju informacijske analogije: udaljenost do središta cilja (A) predstavlja signal,

Poglavlje 4. Eksperimentalno vrednovanje

dok širina ili tolerancija cilja (W) predstavlja šum. Naziv za tu metriku je Fittsov indeks težine (ID , izraženo u bitovima) koji se opisuje sljedećim izrazom:

$$ID = \log_2\left(\frac{2A}{W}\right) \quad (4.1)$$

Fitts je također uveo indeks izvedbe (IP , izražen u bitovima po sekundi) kao mjerilo ljudske sposobnosti. Ovaj pokazatelj kombinira indeks težine zadatka (ID) s trajanjem pokreta (MT , u sekundama) pri odabiru cilja. Prosječna količina informacija koju generira niz pokreta je prosječna informacija za svaki pokret podijeljena s njegovim trajanjem, što je prikazano sljedećim izrazom:

$$IP = \frac{ID}{MT} \quad (4.2)$$

Danas IP najčešće nazivamo propusnošću (TP). Također je uobičajeno u izračunu uzeti u obzir prilagodbu za preciznost. Nakon Fittsa, istraživači su počeli formulirati jednadžbe linearne regresije te analizirati korelaciju kako bi ocijenili kvalitetu podudaranja. Sljedeća jednadžba prikazuje vezu između MT i parametara zadatka A i W :

$$MT = a + b \cdot ID = a + b \cdot \log_2\left(\frac{2A}{W}\right) \quad (4.3)$$

Uvedeni su novi parametri, a i b , koji su konstante koje se odnose na izbor ulaznog uređaja. Tipično se utvrđuju empirijski kroz regresijsku analizu. Konstanta a predstavlja točku presjeka na y-osi i često se interpretira kao vremensko zakašnjenje. S druge strane, b je koeficijent nagiba koji opisuje ubrzanje. Ove dvije konstante ilustriraju linearnu zavisnost prema Fittsovom zakonu.

Formulacija Fittsovog indeksa težine koja se najčešće koristi u području interakcije čovjeka i računala naziva se Shannonova formulacija:

$$ID = \log_2\left(\frac{A}{W} + 1\right) \quad (4.4)$$

Scott MacKenzie predložio je ovu formulaciju koja je dobila ime zbog svoje sličnosti sa Shannon-Hartleyevim teoremom. Teorem opisuje prijenos informacija uzimajući u obzir propusnost, jačinu signala i šum. U kontekstu Fittsovog zakona, udaljenost simbolizira jačinu signala, dok širina cilja označava šum.

Poglavlje 4. Eksperimentalno vrednovanje

Značajno poboljšanje Fittsovog zakona predstavio je Crossman 1956. Uz ovu modifikaciju, širina cilja (W) se zamjenjuje efektivnom širinom cilja (We). Vrijednost We izračunava se temeljem standardne devijacije koordinata odabira prikupljenih kroz seriju testiranja za specifični $A - W$ uvjet. Koristeći efektivnu širinu mete, model Fittsovog zakona bolje odražava što su korisnici zapravo radili, umjesto onoga što im je zadano raditi.

Fittsov zakon, izvorno primijenjen na jednodimenzionalne zadatke, proširen je na dvodimenzionalne zadatke, posebno za pokazivanje na računalnim zaslonima. Za precizno određivanje veličine cilja u 2D prostoru, postoji nekoliko metoda, uključujući popularni W-model, iako je prava reprezentacija za nekružne ciljeve složenija.

Iako Fittsov zakon prvotno nije koncipiran za sustave s glasovnim upravljanjem, ovaj rad će razmotriti njegovu moguću primjenu u sustavima gdje se glasom kontrolira pokazivač miša.

4.2 Postupak testiranja modaliteta

Modaliteti su ocijenjeni u eksperimentu sa 16 ispitanika koristeći metodu ponavljanih mjerenja (eng. *repeated measures*). Ova metoda omogućava testiranje različitih eksperimentalnih uvjeta s istom grupom sudionika, gdje svaki ispitanik iskusi različite uvjete u promjenjivom redoslijedu. Glavni izazov metode je mogući umor sudionika. Kako bi se ovaj problem ublažio, uvedene su kratke pauze između svake promjene modaliteta. S obzirom na to da su svi modaliteti ocijenjeni putem istih zadataka, može se javiti efekt učenja (eng. *learning effect*). Ovaj efekt može dovesti do varijacija u kvaliteti izvedbe temeljene na prethodnom iskustvu, što može utjecati na objektivnost ocjene modaliteta. Da bi se ovaj efekt minimalizirao, koristila se metoda uravnoteženja (eng. *counterbalancing*), koja jamči različit redoslijed uvjeta za svakog sudionika. Za određivanje redoslijeda korišten je balansirani latinski kvadrat, kako je prikazano na slici 4.1.

Latinski kvadrat karakterizira to da se svaki uvjet pojavljuje jednom u svakom retku i stupcu. Budući da se redoslijed uvjeta mijenja za svaku skupinu sudionika, efekt učenja se uravnotežuje. Da bi skupine bile iste veličine, ukupan broj sudionika

A	B	D	C
B	C	A	D
C	D	B	A
D	A	C	B

A - Modalitet 1
B - Modalitet 2
C - Modalitet 3
D - Modalitet 4

Slika 4.1 Balansirani latinski kvadrat

treba biti u skladu s brojem uvjeta. U ovom istraživanju, s 4 modaliteta i 16 sudionika, taj kriterij je ispunjen. Iako latinski kvadrat pomaže u balansiranju, ne rješava potpuno problem efekta učenja. Bolja opcija je upotreba balansirano latinskog kvadrata, u kojem svaki uvjet, osim što se pojavljuje točno jednom u svakom retku i stupcu, također slijedi i prethodi svakom drugom uvjetu jednaki broj puta.

Na početku eksperimenta korisniku se detaljno objašnjava procedura te ga se upozna s karakteristikama modaliteta uz pomoć grafičkog prikaza (Dodatak A) za bolje razumijevanje. Informira ga se o predviđenom vremenu trajanja eksperimenta, koje je 60 minuta, te mu se pruža prilika za kratku praksu s modalitetima. Sudionici potom popunjavaju početni upitnik s osnovnim informacijama, kao što su ime, prezime, starost i spol. Ispitni zadaci se zatim provode kroz aplikaciju FittsStudy i ranije spomenutu vlastitu aplikaciju za testiranje. Po završetku, sudionicima se daju upitnici kako bi izrazili svoje dojmove o testiranim modalitetima.

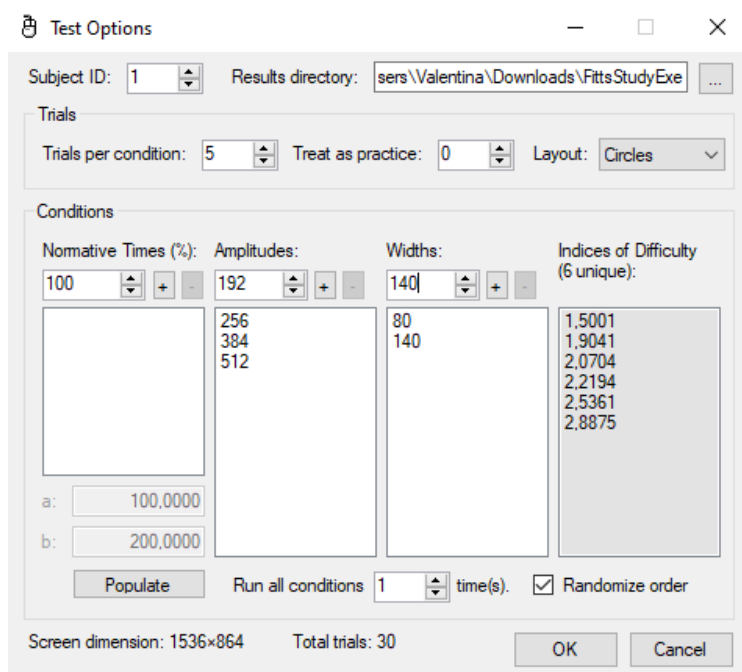
4.2.1 Aplikacija FittsStudy

FittsStudy [14], autora Jacob O. Wobbrock, Susumu Harada, Edward Cutrell i I. Scott MacKenzie, aplikacija je za desktop uređaje. Služi za provođenje i analizu istraživanja vezanih za operaciju pokazivanja-i-odabira u skladu s utvrđenim međunarodnim i akademskim standardima. Osim što podržava manipulaciju A i W ,

Poglavlje 4. Eksperimentalno vrednovanje

podržava i manipulaciju MT s metronomom. Podržani su i 1D i 2D zadaci usmjerenja temeljeni na ISO 9241-9 [15] standardu. Ostale značajke uključuju automatsko izračunavanje modela i mjera, izvoz rezultata u proračunske tablice te alat za vizualno istraživanje pojedinih pokušaja.

Nakon pokretanja aplikacije pritiskom kombinacije tipki "Ctrl + N" prikazuje se izbornik za postavljanje parametara eksperimenta, kao što je prikazano na slici 4.2.



Slika 4.2 Izbornik za postavljanje parametara eksperimenta unutar aplikacije FittsStudy

Opis bitnih parametara dan je u nastavku:

- *Subject ID* - Identifikator ispitanika koja služi kao jedinstvena oznaka.
- *Layout* - Padajući izbornik u kojem se bira između opcija *Ribbons* (1D zadaci) i *Circles* (2D zadaci).
- *Trials per condition* - Broj meta koje ispitanik mora pogoditi po seriji kombinacija udaljenosti i širine meta.

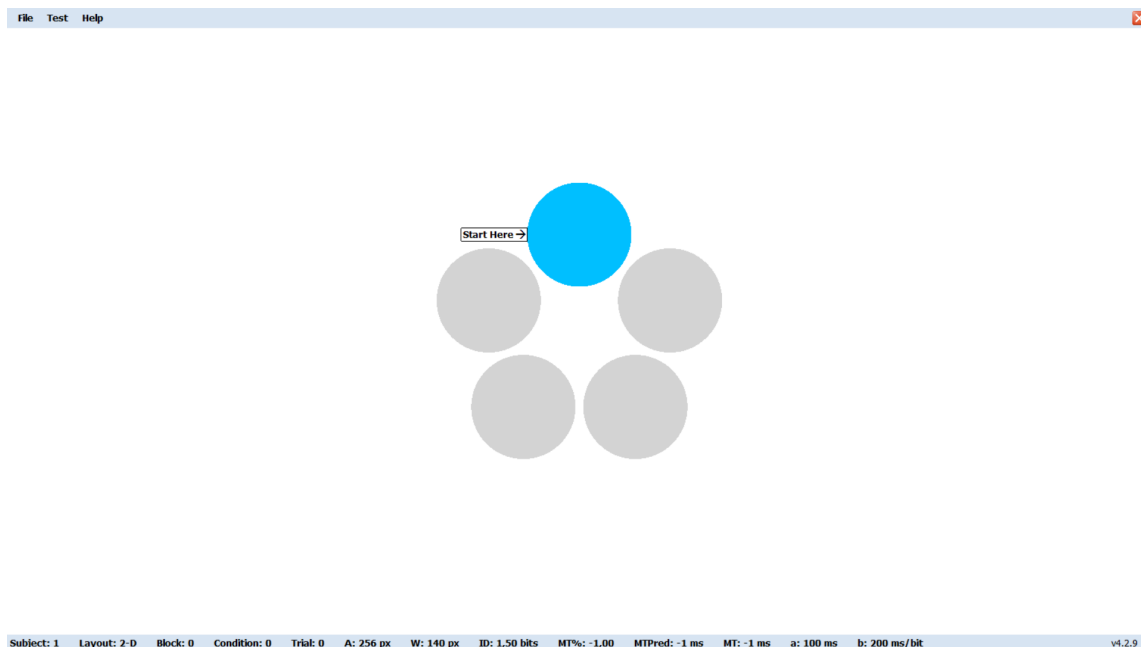
Poglavlje 4. Eksperimentalno vrednovanje

- *Treat as practice* - Označava broj meta koje će se tretirati kao vježba, bez mjerenja podataka.
- *Amplitudes* - Udaljenosti meta koje se pojavljuju u zadacima.
- *Widths* - Širine meta koje se pojavljuju u zadacima.

Svim ispitanicima postavljena je identična konfiguracija parametara. Za *Layout* parametar odabrana je opcija *Circles*. Definirano je šest serija ispitnih zadataka s kombinacijama različitih udaljenosti i širina meta. Odabrane su tri moguće udaljenosti meta: 256, 384 i 512 piksela te dvije moguće kombinacije širine meta: 80 i 140 piksela. Svaka serija sastoji se od 5 kružnih meta koje ispitanik treba pogoditi. Za svaki modalitet ispitanik ponavlja isti eksperiment. Broj pokušaja za vježbanje postavljen je na nulu jer su ispitanici već imali priliku vježbati prije samih ispitnih zadataka.

Po odabiru parametara, testiranje započinje pritiskom na gumb "OK". Prikazuje se prvi zadatak koji uključuje seriju kombinacija udaljenosti i širine, sastavljen od 5 ciljnih kružnica na virtualnom obodu, prikazan na slici 4.3. Korisnika se porukom "Start here" upućuje gdje započinje s ispitnim zadatkom. Ciljna kružnica koju treba označiti uvijek je plave boje. Mjerenje vremena ne započinje dok korisnik ne označi prvu metu. S pokretanjem mjerenja, moguće su greške, a pogrešno označena meta postaje crvena. Bez obzira je li meta pogođena ili promašena, iduća meta postaje aktivna sve dok se ne pokuša pogoditi svih 5 meta. Time završava jedna od 6 serija. Po završetku svih šest serija, rezultati se spremaju u xml datoteku, a isti postupak ponavlja se za sve modalitete, u poretku određenom postupkom uravnoteženja.

Poglavlje 4. Eksperimentalno vrednovanje



Slika 4.3 Primjer ispitnog zadatka unutar aplikacije FittsStudy

4.2.2 Vlastita aplikacija za testiranje

U poglavlju 3.3 opisana je vlastita aplikacija koja se koristila za mjerenje vremena potrebnog za rješavanje specifičnog jednostavnog zadatka korištenjem određenog modaliteta. Da bi se izbalansirao potencijalni efekt učenja, polovici ispitanika je prvo ponuđeno testiranje kroz FittsStudy aplikaciju, dok je druga polovica prvo koristila ovu aplikaciju.

4.2.3 Upitnici

Nakon uspješnog završetka svih zadataka unutar testnih aplikacija, ispitaniku su predstavljani sljedeći upitnici: NASA-TLX, upitnik o upotrebljivosti i završni upitnik.

NASA-TLX (eng. *NASA Task Load Index*) [16] je upitnik koji se koristi za procjenu percipiranog radnog opterećenja od strane ispitanika. Stvoren od strane

Poglavlje 4. Eksperimentalno vrednovanje

NASA-e, omogućava istraživačima da sagledaju koliko se korisnik osjeća opterećeno tijekom obavljanja raznovrsnih zadataka. Ovaj upitnik obuhvaća pet dimenzija, koje se boduju na skali od 1 do 21, a detaljnije su opisane u nastavku

- Mentalna zahtjevnost (eng. *mental demand*) - Reflektira potrebnu mentalnu aktivnost za obavljanje zadatka, uključujući koncentraciju, odlučivanje i ostale kognitivne funkcije. Veća zahtjevnost može sugerirati složenost ili potrebu za dubljim razmišljanjem, dok manja zahtjevnost upućuje na jednostavnost zadatka.
- Fizička zahtjevnost (eng. *physical demand*) - Označava količinu fizičke aktivnosti ili napora koji je potreban za obavljanje određenog zadatka. To može uključivati aspekte kao što su snaga, izdržljivost, pokreti, ili čak koordinacija prilikom izvođenja aktivnosti.
- Frustracija (eng. *frustration*) - Opisuje osjećaje nezadovoljstva ili frustracije tijekom izvođenja zadatka. Visoka razina sugerira izazovnost ili nejasnoću zadatka, dok niska razina ukazuje na jednostavnost i glatkoću procesa.
- Izvedba (eng. *performance*) - Predstavlja ispitanikovu percepciju o vlastitoj uspješnosti u zadatku. Fokusira se na subjektivni osjećaj postignuća, a ne na stvarnu učinkovitost.
- Trud (eng. *effort*) - Povezuje se s mentalnom i fizičkom energijom koju ispitanik percipira da je uložio. Ovo je subjektivna mjera uloženog truda za postizanje rezultata; veća ocjena upućuje na izazovnost zadatka, dok manja sugerira lakoću izvršenja.

Upitnik o upotrebljivosti služi kao alat za procjenu korisničke percepcije o jednostavnosti i efikasnosti određenog proizvoda ili sustava. Sadrži tri dimenzije, a odgovori su predstavljeni na Likertovoj skali koja se kreće od 1 (u potpunosti se ne slažem) do 7 (u potpunosti se slažem). Detalji o dimenzijama navode se u nastavku:

- Lakoća korištenja (eng. *efficiency*) - Predstavlja koliko je modalitet intuitivan i jednostavan za upotrebu.
- Lakoća učenja (eng. *learnability*) - Označava koliko je brzo i jednostavno korisniku svladati korištenje modaliteta.

Poglavlje 4. Eksperimentalno vrednovanje

- Zadovoljstvo (eng. *satisfaction*) - Mjera koliko je korisnik zadovoljan izvedbom pokazivanja-i-odabira pomoću dotičnog modaliteta.

Završni upitnik postavlja samo jedno pitanje, a to je koji modalitet bi ispitanik preferirao koristiti.

Poglavlje 5

Analiza i rezultati

Analiza je provedena koristeći alat SPSS (*Statistical Package for the Social Sciences*) [17], razvijen od strane IBM-a. IBM SPSS je vodeći softverski paket namijenjen statističkoj analizi podataka koji se široko koristi u društvenim, znanstvenim i poslovnim istraživanjima. SPSS omogućuje korisnicima da provode niz statističkih testova, uključujući deskriptivne statistike, regresijske analize, testiranje hipoteza i mnoge druge.

5.1 Ispitna skupina

Kao što je ranije navedeno, eksperimentu je pristupilo 16 ispitanika. Prosječna dob ispitanika je 25,3125, sa standardnom devijacijom od 1,488. Deset ispitanika (62,5%) pripada muškom spolu, a šest ispitanika (37,5%) pripada ženskom spolu.

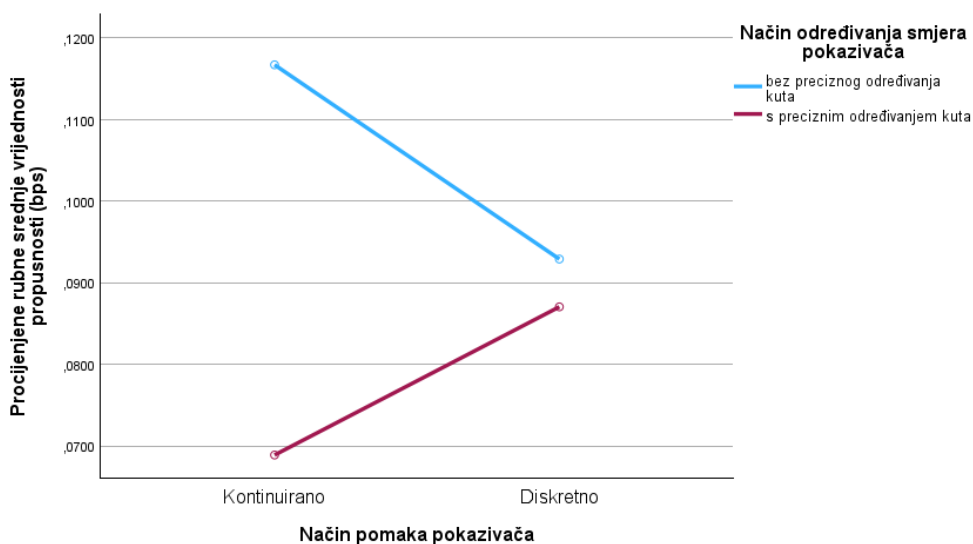
5.2 Analiza rezultata eksperimenta

Eksperiment ima za cilj utvrditi postoji li statistički relevantna razlika između modaliteta za glasovno upravljanje pokazivačem i identificirati najučinkovitiji od njih. Za analizu rezultata eksperimenta korištena je parametarska dvofaktorska ANOVA s ponovljenim mjerenjima (eng. *Two-Way Repeated Measures ANOVA*). Ova metoda omogućava analizu utjecaja dviju nezavisnih varijabli na zavisnu varijablu. Dok

standardna ANOVA analizira razlike temeljem jedne nezavisne varijable, dvofaktorska ANOVA proučava potencijalne interakcije između dvije nezavisne varijable te kako njihova kombinacija može utjecati na ishod. Važno je napomenuti da je dvofaktorska ANOVA parametarski test koji pretpostavlja da su podaci interpretirani na intervalnoj skali i da slijede normalnu distribuciju. Također, s obzirom na ponovljena mjerenja, pretpostavka je da svaki sudionik prolazi kroz sve modalitete. U ovom eksperimentu, nezavisne varijable obuhvaćaju način pomaka i način određivanje smjera pokazivača, s ciljem istraživanja njihovog utjecaja na propusnost i vremensku učinkovitost.

5.2.1 Propusnost

Analizom je utvrđeno da ne postoji statistički značajna razlika ($F(1, 15) = 0,739$, $p = 0,404 > 0,05$) u propusnosti između testiranih modaliteta s obzirom na način pomaka pokazivača. Također je utvrđeno da postoji statistički značajna razlika u propusnosti između modaliteta koji implementiraju različit način određivanja smjera pokazivača: $F(1, 15) = 26,652$, $p < 0,001$.



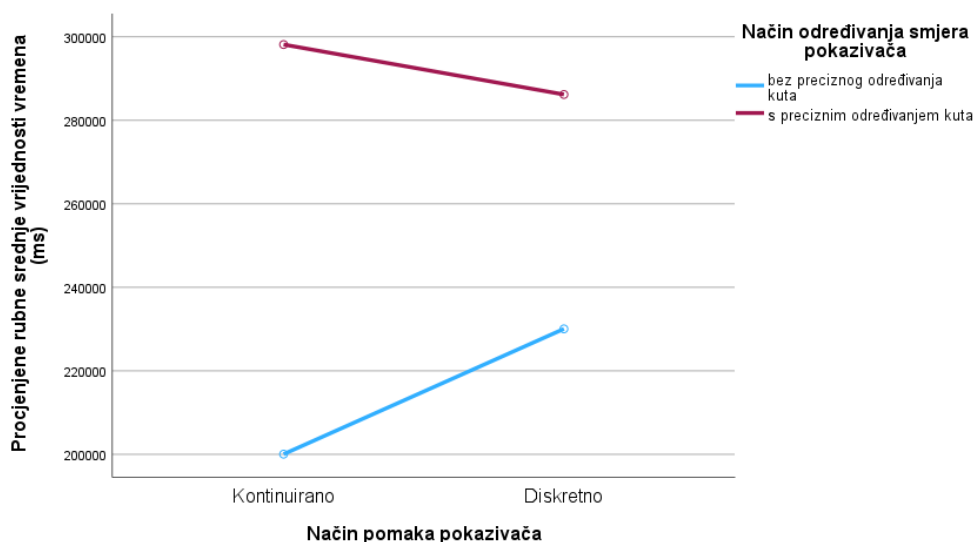
Slika 5.1 Profilni dijagram (eng. *Profile plot*) propusnosti

Na dijagramu na slici 5.1 može se uočiti da modaliteti koji implementiraju određivanje smjera pokazivača bez preciznog određivanja kuta imaju bolje srednje vrijednosti propusnosti. Modalitet 1, koji predstavlja kombinaciju kontinuiranog pomaka i određivanja smjera bez preciznog određivanja kuta, postiže najbolje vrijednosti propusnosti. S druge strane, modalitet 3, koji kombinira kontinuirani pomak i određivanje smjera s preciznim određivanjem kuta, daje najgore vrijednosti propusnosti. Velika razlika u propusnosti između 1. i 3. modaliteta proizlazi iz načina implementacije; treći modalitet zahtijeva više vremena za prepoznavanje ključnih riječi jer prepoznaje i brojeve, što rezultira slabijom responzivnošću i samim time slabijom preciznošću.

5.2.2 Vrijeme

Za procjenu vremenske učinkovitosti, korišteni su podaci o vremenu potrebnom za dovršetak osnovnog zadatka unutar vlastite testne aplikacije. Rezultati analize pokazuju da nema statistički značajne razlike ($F(1, 15) = 0,284$, $p = 0,602 > 0,05$) u izmjenom vremenu između modaliteta s različitim načinima pomaka pokazivača. Međutim, postoji značajna razlika u vremenu potrebnom za dovršetak zadatka među modalitetima s različitim načinima određivanja smjera pokazivača: $F(1, 15) = 18,204$, $p < 0,001$.

Na dijagramu prikazanom na slici 5.2 može se uočiti da modalitet 1, koji kombinira kontinuirani pomak i određivanje smjera bez preciznog određivanja kuta, postiže najkraće vrijednosti izmjenog vremena. Suprotno tome, modalitet 3, koji uključuje kontinuirani pomak i određivanje smjera s točnim određivanjem kuta, ima najdulje izmjereno vrijeme. Dijagram također pokazuje veliku razliku između modaliteta koji primjenjuju različite metode određivanja smjera kursora. Modaliteti koji se oslanjaju na određivanje smjera bez preciznog određivanja kuta su znatno brži. Ovo se može objasniti činjenicom da je potrebno više razmišljanja prilikom određivanja točnog kuta u smjeru kojim će se kursor pomicati, ali i mogućnošću da program ne registrira promptno ispravne vrijednosti.



Slika 5.2 Profilni dijagram (eng. *Profile plot*) vremena izvršavanja zadatka

5.3 Analiza rezultata upitnika

Kao što je istaknuto u prethodnom poglavlju, nakon eksperimenta ispitanici su ispunili upitnik o upotrebljivosti i NASA-TLX upitnik radi prikupljanja subjektivnih dojmova o modalitetima. S obzirom na to da podaci iz upitnika nisu na intervalnoj skali, za njihovu analizu korišten je Friedmanov neparametarski test. Friedmanov test se primjenjuje za identifikaciju razlika u višestrukim testiranjima. Za razliku od analize podataka vezanih uz propusnost i vrijeme izvršenja zadatka, tijekom analize rezultata upitnika svaki modalitet je analiziran zasebno.

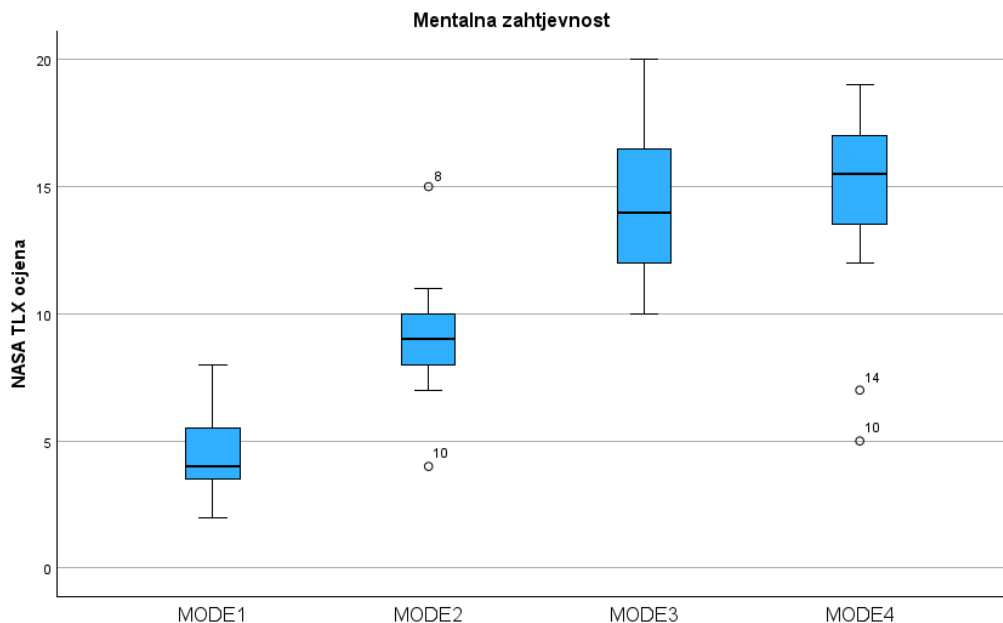
Za svaku stavku unutar upitnika proveden je Friedmanov test s kojim se utvrđuje postoji li statistički značajna razlika u subjektivnom dojmu korisnika, s obzirom na korišteni modalitet glasovnog upravljanja pokazivačem miša. U slučaju da Friedmanov test potvrdi da postoji statistički značajna razlika, provodi se *post-hoc* analiza. Za precizno određivanje između kojih modaliteta postoji statistički značajna razlika, primijenjena je Dunn-Bonferronijeva metoda. Ta metoda odvojeno uspoređuje sve kombinacije modaliteta te korigira *p* vrijednost, čime se smanjuje rizik od lažno pozitivnih rezultata zbog višestrukih usporedbi.

5.3.1 NASA-TLX upitnik

U ovom potpoglavlju, provedena je analiza rezultata dobivenih putem NASA-TLX upitnika.

Mentalna zahtjevnost

Grafička usporedba ocjene mentalne zahtjevnosti prikazana je na slici 5.3. Na grafu je vidljivo da su modaliteti 3 i 4, koji primjenjuju određivanje smjera s preciznim određivanjem kuta, mentalno zahtjevniji.



Slika 5.3 Grafička usporedba ocjene mentalne zahtjevnosti po modalitetima

Friedmanov test utvrdio je da postoji statistički značajna razlika u mentalnoj zahtjevnosti između modaliteta ($\chi^2(3) = 41,925$, $p < 0,001$). Kako bi se saznalo između kojih modaliteta postoji razlika, provedena je spomenuta *post hoc* analiza. Rezultati su dani u nastavku:

- MODALITET 1 ($4,7 \pm 1,78$) vs. MODALITET 2 ($9,1 \pm 2,33$) s rezultatom Z

Poglavlje 5. Analiza i rezultati

$$= -2,328, p = 0,120$$

- MODALITET 1 ($4,7 \pm 1,78$) vs. MODALITET 3 ($14,3 \pm 3,00$) s rezultatom $Z = -5,066, p = 0,000$
- MODALITET 1 ($4,7 \pm 1,78$) vs. MODALITET 4 ($14,6 \pm 3,9$) s rezultatom $Z = -5,751, p = 0,000$
- MODALITET 2 ($9,1 \pm 2,33$) vs. MODALITET 3 ($14,3 \pm 3,00$) s rezultatom $Z = -2,739, p = 0,037$
- MODALITET 2 ($9,1 \pm 2,33$) vs. MODALITET 4 ($14,6 \pm 3,9$) s rezultatom $Z = -3,423, p = 0,004$
- MODALITET 3 ($14,3 \pm 3,00$) vs. MODALITET 4 ($14,6 \pm 3,9$) s rezultatom $Z = -0,685, p = 1,000$

Analizom je utvrđeno da postoji statistički značajna razlika između svih kombinacija modaliteta osim između modaliteta 1 i 2 ($p = 0,120 > 0,05$) i modaliteta 3 i 4 ($p = 1,000 > 0,05$).

Fizička zahtjevnost

Na slici 5.4 prikazana je grafička usporedba ocjene fizičke zahtjevnosti po modalitetima. Jasno se uočava da modaliteti sa složenijim glasovnim naredbama iziskuju veći fizički napor.

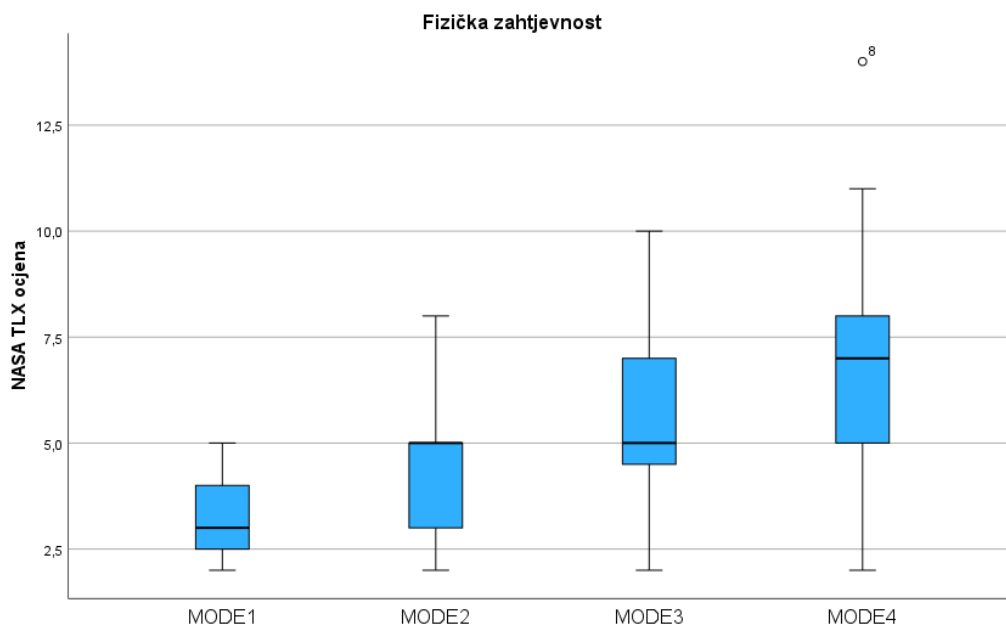
Analizom je utvrđeno da postoji statistički značajna razlika ($\chi^2(3) = 26,036, p < 0,001$) u fizičkoj zahtjevnosti između modaliteta. *Post hoc* analiza dala je sljedeće rezultate:

- MODALITET 1 ($3,3 \pm 1,06$) vs. MODALITET 2 ($4,2 \pm 1,68$) s rezultatom $Z = -1,369, p = 1,000$
- MODALITET 1 ($3,3 \pm 1,06$) vs. MODALITET 3 ($5,6 \pm 2,22$) s rezultatom $Z = -2,944, p = 0,019$
- MODALITET 1 ($3,3 \pm 1,06$) vs. MODALITET 4 ($6,6 \pm 3,07$) s rezultatom $Z = -4,450, p = 0,000$

Poglavlje 5. Analiza i rezultati

- MODALITET 2 ($4,2 \pm 1,68$) vs. MODALITET 3 ($5,6 \pm 2,22$) s rezultatom $Z = -1,575$, $p = 0,692$
- MODALITET 2 ($4,2 \pm 1,68$) vs. MODALITET 4 ($6,6 \pm 3,07$) s rezultatom $Z = -3,081$, $p = 0,012$
- MODALITET 3 ($5,6 \pm 2,22$) vs. MODALITET 4 ($6,6 \pm 3,07$) s rezultatom $Z = -1,506$, $p = 0,792$

Rezultati analize pokazuju statistički značajnu razliku ($p < 0,05$) među sljedećim kombinacijama modaliteta: 1 i 3, 1 i 4 te 2 i 4.



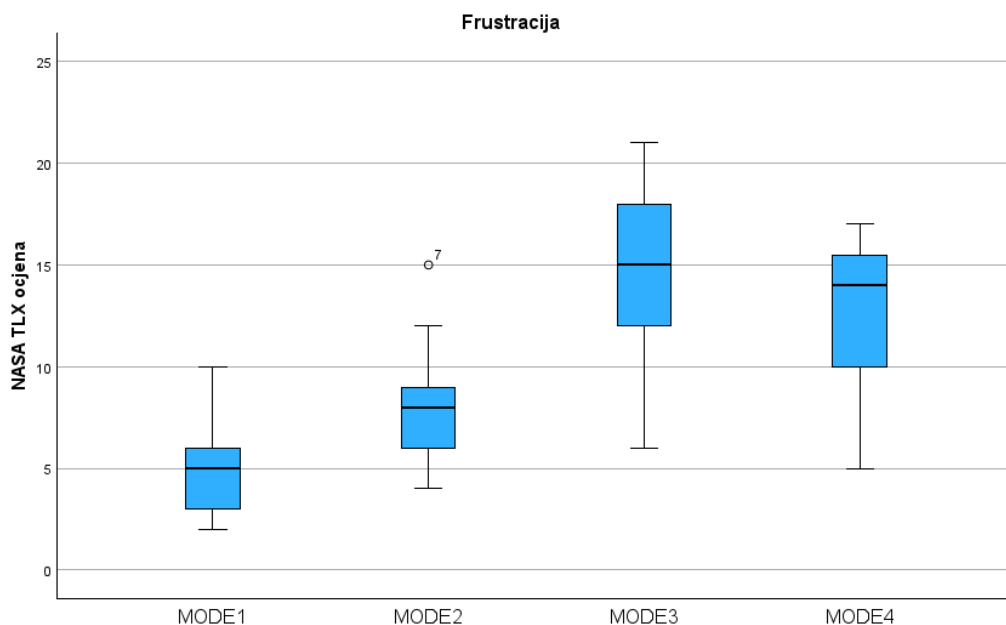
Slika 5.4 Grafička usporedba ocjene fizičke zahtjevnosti po modalitetima

Frustracija

Na grafu prikazanom na slici 5.5 ilustrirana je grafička usporedba ocjene frustracije po modalitetima. Vidljivo je da modaliteti koji koriste brojeve kao glasovne naredbe često uzrokuju veću frustraciju. To je posljedica mogućih pogrešaka u interpretaciji

Poglavlje 5. Analiza i rezultati

izgovorenog broja od strane sustava ili korisnikove pogrešne procjene kuta i udaljenosti. Dodatno, modalitet 3 (s kontinuiranim pomakom) pokazuje se kao posebno frustrirajući zbog svoje sporije reakcije na naredbe što ga čini manje responsivnim.



Slika 5.5 Grafička usporedba ocjene frustracije po modalitetima

Friedmanov test utvrdio je da postoji statistički značajna razlika u frustraciji između modaliteta ($\chi^2(3) = 35,620$, $p < 0,001$). *Post hoc* analizom dobiveni su sljedeći rezultati:

- MODALITET 1 ($4,9 \pm 2,13$) vs. MODALITET 2 ($7,9 \pm 2,91$) s rezultatom $Z = -1,780$, $p = 0,450$
- MODALITET 1 ($4,9 \pm 2,13$) vs. MODALITET 3 ($15,0 \pm 3,86$) s rezultatom $Z = -4,519$, $p = 0,000$
- MODALITET 1 ($4,9 \pm 2,13$) vs. MODALITET 4 ($12,5 \pm 4,05$) s rezultatom $Z = -5,203$, $p = 0,000$
- MODALITET 2 ($7,9 \pm 2,91$) vs. MODALITET 3 ($15,0 \pm 3,86$) s rezultatom $Z = -2,739$, $p = 0,037$

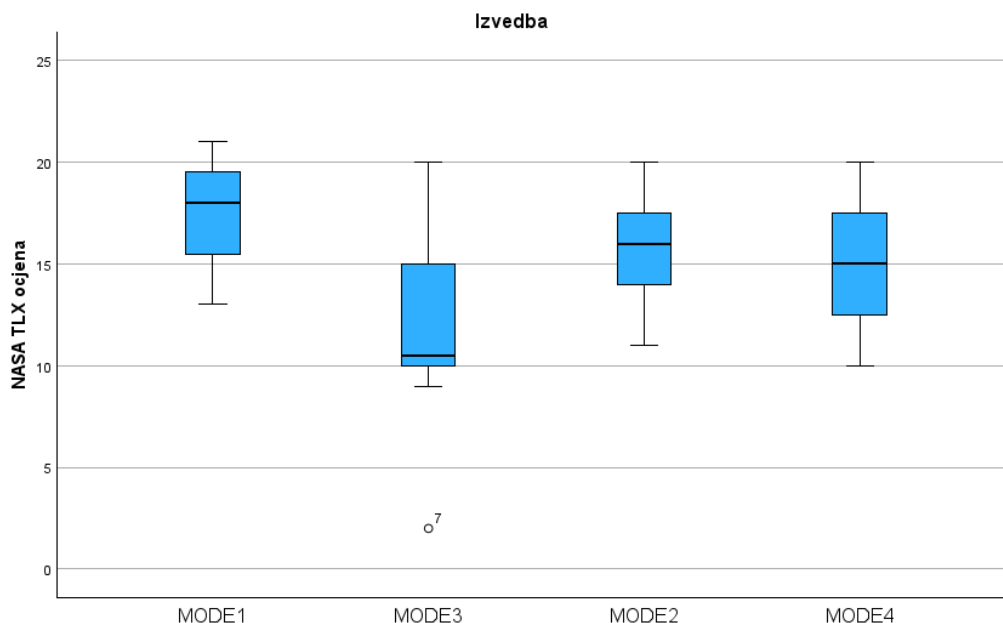
Poglavlje 5. Analiza i rezultati

- MODALITET 2 ($7,9 \pm 2,91$) vs. MODALITET 4 ($12,5 \pm 4,05$) s rezultatom $Z = -3,423$, $p = 0,004$
- MODALITET 3 ($15,0 \pm 3,86$) vs. MODALITET 4 ($12,5 \pm 4,05$) s rezultatom $Z = 0,685$, $p = 1,000$

Rezultati analize pokazuju postojanje statistički značajne razlike ($p < 0,05$) među svim kombinacijama modaliteta, osim između modaliteta 1 i 2 te 3 i 4.

Izvedba

Grafička usporedba subjektivne procijenjene učinkovitosti po modalitetima prikazana je na slici 5.6. Na grafu je vidljivo da nema značajnih razlika u percipiranoj učinkovitosti između različitih modaliteta. No, vrijedno je napomenuti da modalitet 3 bilježi nešto slabije performanse, što vjerojatno objašnjava visoku razinu frustracije koju korisnici izražavaju pri njegovoj upotrebi.



Slika 5.6 Grafička usporedba subjektivne procijenjene učinkovitosti po modalitetima

Poglavlje 5. Analiza i rezultati

Analizom je utvrđeno da postoji statistički značajna razlika ($\chi^2(3) = 30,336$, $p < 0,001$) u izvedbi između modaliteta. *Post hoc* analiza dala je sljedeće rezultate:

- MODALITET 3 ($11,8 \pm 4,25$) vs. MODALITET 4 ($15,1 \pm 3,15$) s rezultatom $Z = -2,533$, $p = 0,068$
- MODALITET 3 ($11,8 \pm 4,25$) vs. MODALITET 2 ($15,8 \pm 2,56$) s rezultatom $Z = 3,834$, $p = 0,001$
- MODALITET 3 ($11,8 \pm 4,25$) vs. MODALITET 1 ($17,4 \pm 2,39$) s rezultatom $Z = 5,135$, $p = 0,000$
- MODALITET 4 ($15,1 \pm 3,15$) vs. MODALITET 2 ($15,8 \pm 2,56$) s rezultatom $Z = 1,301$, $p = 1,000$
- MODALITET 4 ($15,1 \pm 3,15$) vs. MODALITET 1 ($17,4 \pm 2,39$) s rezultatom $Z = 2,602$, $p = 0,056$
- MODALITET 2 ($15,8 \pm 2,56$) vs. MODALITET 1 ($17,4 \pm 2,39$) s rezultatom $Z = 1,301$, $p = 1,000$

Analički rezultati otkrivaju statistički značajne razlike ($p < 0,05$) između modaliteta 1 i 3 te 2 i 3.

Trud

Na grafu prikazanom na slici 5.7 ilustrirana je grafička usporedba ocjene truda po modalitetima. Budući da modaliteti 1 i 2 nude intuitivniji pristup i ne zahtijevaju precizno određivanje kuta prilikom odabira smjera, vidljivo je da korisnici ulažu manje truda tijekom rada s njima.

Friedmanov test utvrdio je da postoji statistički značajna razlika u uloženom trudu između modaliteta ($\chi^2(3) = 30,336$, $p < 0,001$). *Post hoc* analizom dobiveni su sljedeći rezultati:

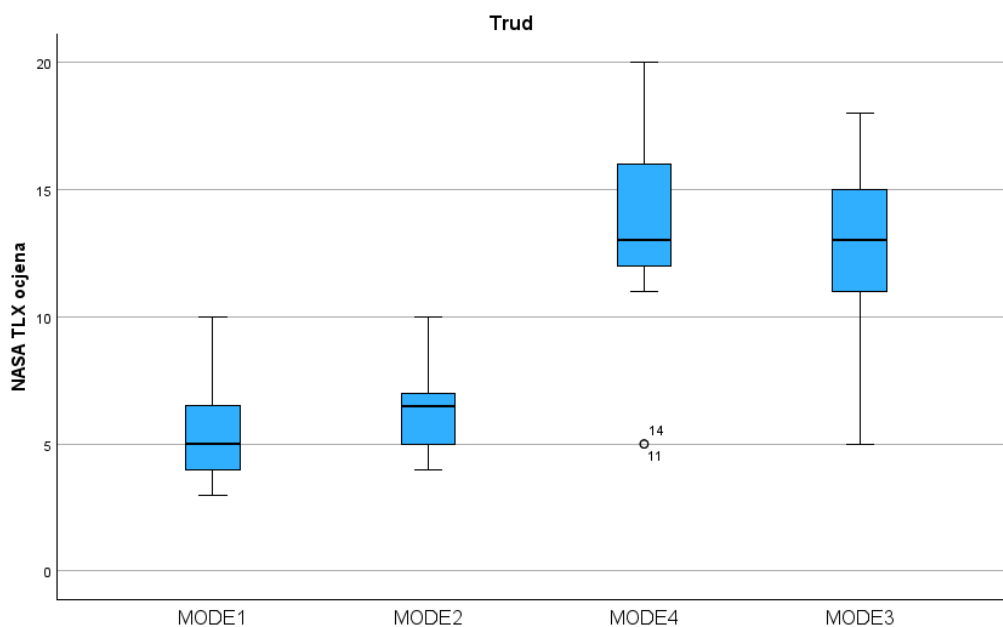
- MODALITET 1 ($5,5 \pm 1,86$) vs. MODALITET 2 ($6,4 \pm 1,45$) s rezultatom $Z = -1,301$, $p = 1,000$
- MODALITET 1 ($5,5 \pm 1,86$) vs. MODALITET 3 ($12,7 \pm 3,03$) s rezultatom

Poglavlje 5. Analiza i rezultati

$$Z = -4,519, p = 0,000$$

- MODALITET 1 ($5,5 \pm 1,86$) vs. MODALITET 4 ($13,2 \pm 4,02$) s rezultatom $Z = -4,861, p < 0,000$
- MODALITET 2 ($6,4 \pm 1,45$) vs. MODALITET 3 ($12,7 \pm 3,03$) s rezultatom $Z = -3,218, p = 0,008$
- MODALITET 2 ($6,4 \pm 1,45$) vs. MODALITET 4 ($13,2 \pm 4,02$) s rezultatom $Z = -3,560, p < 0,002$
- MODALITET 3 ($12,7 \pm 3,03$) vs. MODALITET 4 ($13,2 \pm 4,02$) s rezultatom $Z = -0,342, p = 1,000$

Analiza rezultata ukazuje na statistički značajne razlike ($p < 0,05$) između kombinacija modaliteta, osim kada uspoređujemo modalitete 1 i 2 te 3 i 4.



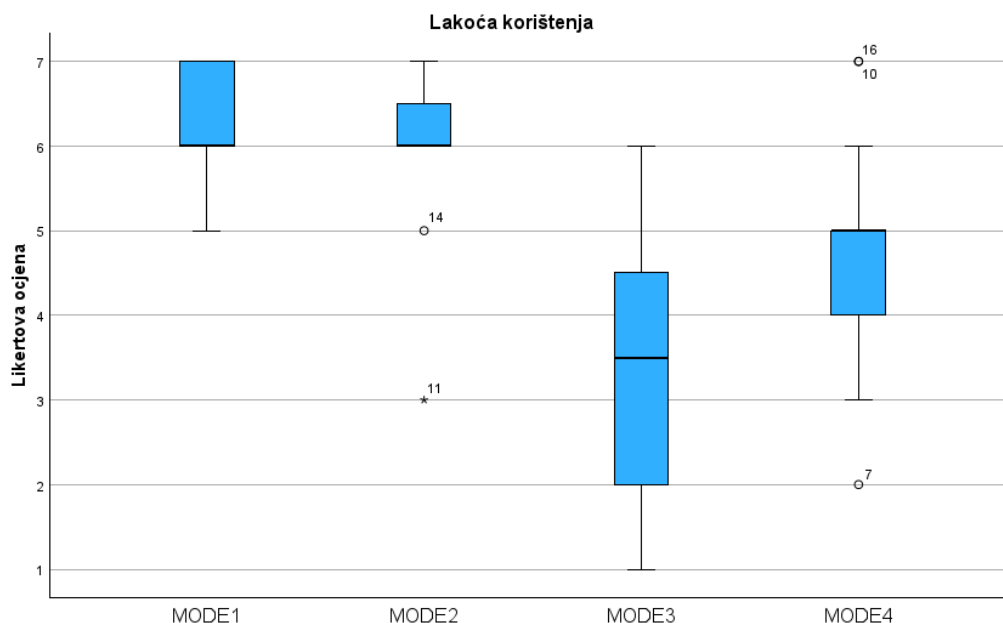
Slika 5.7 Grafička usporedba ocjene truda po modalitetima

5.3.2 Upitnik upotrebljivosti

Rezultati dobiveni putem upitnika o upotrebljivosti objašnjeni su u ovom potpoglavljju.

Lakoća korištenja

Grafička usporedba subjektivno procijenjene lakoće korištenja po modalitetima prikazana je na slici 5.8. Modaliteti 1 i 2 su se izdvojili svojom intuitivnošću i jednostavnošću uporabe.



Slika 5.8 Grafička usporedba subjektivno procijenjene lakoće korištenja po modalitetima

Analizom je utvrđeno da postoji statistički značajna razlika ($\chi^2(3) = 33,640$, $p < 0,001$) u lakoći korištenja između modaliteta. *Post hoc* analizom dobiveni su sljedeći rezultati:

- MODALITET 3 ($3,4 \pm 1,41$) vs. MODALITET 4 ($4,8 \pm 1,29$) s rezultatom Z

Poglavlje 5. Analiza i rezultati

$$= -1,849, p = 0,387$$

- MODALITET 3 ($3,4 \pm 1,41$) vs. MODALITET 2 ($6,0 \pm 0,97$) s rezultatom $Z = 3,971, p = 0,000$
- MODALITET 3 ($3,4 \pm 1,41$) vs. MODALITET 1 ($6,3 \pm 0,70$) s rezultatom $Z = 4,861, p = 0,000$
- MODALITET 4 ($4,8 \pm 1,29$) vs. MODALITET 2 ($6,0 \pm 0,97$) s rezultatom $Z = 2,122, p = 0,203$
- MODALITET 4 ($4,8 \pm 1,29$) vs. MODALITET 1 ($6,3 \pm 0,70$) s rezultatom $Z = 3,012, p = 0,016$
- MODALITET 2 ($6,0 \pm 0,97$) vs. MODALITET 1 ($6,3 \pm 0,70$) s rezultatom $Z = 0,890, p = 1,000$

Rezultati ističu statistički značajne razlike ($p < 0,05$) između kombinacija modaliteta 1 i 3, 1 i 4 te 2 i 3.

Lakoća učenja

Na slici 5.9 prikazan je graf koji uspoređuje procijene lakoće učenja po modalitetima. Graf sugerira da su modaliteti 1 i 2 najlakši za naučiti koristiti, vjerojatno zbog njihove uporabe određivanja smjera bez potrebe za preciznim definiranjem kuta.

Analizom je utvrđeno da postoji statistički značajna razlika ($\chi^2(3) = 30,137, p < 0,001$) u lakoći učenja između modaliteta. *Post hoc* analizom dobiveni su sljedeći rezultati:

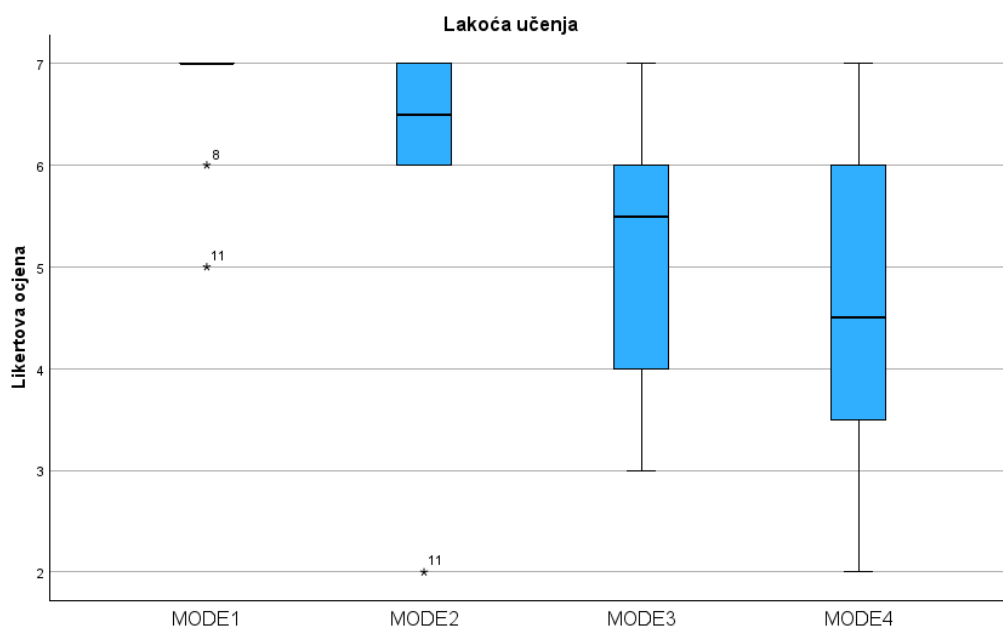
- MODALITET 4 ($4,7 \pm 1,58$) vs. MODALITET 3 ($5,2 \pm 1,28$) s rezultatom $Z = 1,095, p = 1,000$
- MODALITET 4 ($4,7 \pm 1,58$) vs. MODALITET 2 ($6,3 \pm 1,24$) s rezultatom $Z = 3,149, p = 0,010$
- MODALITET 4 ($4,7 \pm 1,58$) vs. MODALITET 1 ($6,8 \pm 0,54$) s rezultatom $Z = 4,519, p = 0,000$
- MODALITET 3 ($5,2 \pm 1,28$) vs. MODALITET 2 ($6,3 \pm 1,24$) s rezultatom Z

Poglavlje 5. Analiza i rezultati

= 2,054, $p = 0,240$

- MODALITET 3 ($5,2 \pm 1,28$) vs. MODALITET 1 ($6,8 \pm 0,54$) s rezultatom $Z = 3,423$, $p = 0,004$
- MODALITET 2 ($6,3 \pm 1,24$) vs. MODALITET 1 ($6,8 \pm 0,54$) s rezultatom $Z = 1,369$, $p = 1,000$

Analiza je istaknula statistički značajne razlike ($p < 0,05$) među modalitetima 2 i 4, 1 i 4 te 1 i 3.

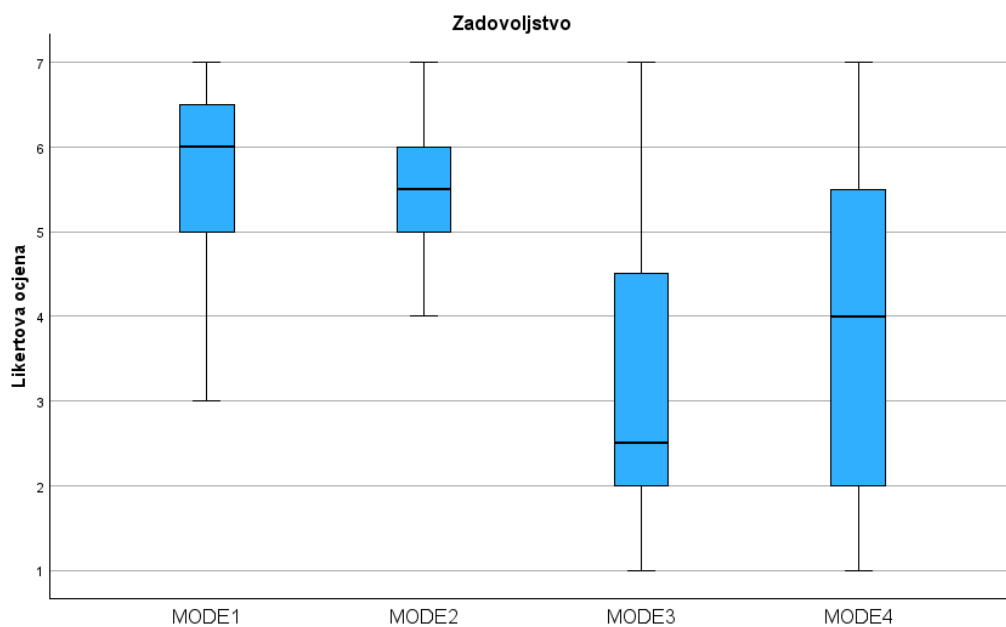


Slika 5.9 Grafička usporedba procijene lakoće učenja po modalitetima

Zadovoljstvo

Grafička usporedba ocjena zadovoljstva po modalitetima prikazana je na slici 5.10. Vidljivo je da su korisnici zadovoljniji svojom izvedbom tijekom korištenja modaliteta koji implementiraju određivanje smjera bez točnog određivanja kuta.

Poglavlje 5. Analiza i rezultati



Slika 5.10 Grafička usporedba ocjena zadovoljstva po modalitetima

Friedmanov test utvrdio je da postoji statistički značajna razlika u zadovoljstvu između modaliteta ($\chi^2(3) = 23,672$, $p < 0,001$). *Post hoc* analizom dobiveni su sljedeći rezultati:

- MODALITET 3 ($3,2 \pm 1,87$) vs. MODALITET 4 ($3,8 \pm 1,97$) s rezultatom $Z = -1,095$, $p = 1,000$
- MODALITET 3 ($3,2 \pm 1,87$) vs. MODALITET 2 ($5,4 \pm 0,81$) s rezultatom $Z = 3,081$, $p = 0,012$
- MODALITET 3 ($3,2 \pm 1,87$) vs. MODALITET 1 ($5,8 \pm 1,05$) s rezultatom $Z = 4,039$, $p = 0,000$
- MODALITET 4 ($3,8 \pm 1,97$) vs. MODALITET 2 ($5,4 \pm 0,81$) s rezultatom $Z = 1,985$, $p = 0,283$
- MODALITET 4 ($3,8 \pm 1,97$) vs. MODALITET 1 ($5,8 \pm 1,05$) s rezultatom $Z = 2,944$, $p = 0,019$

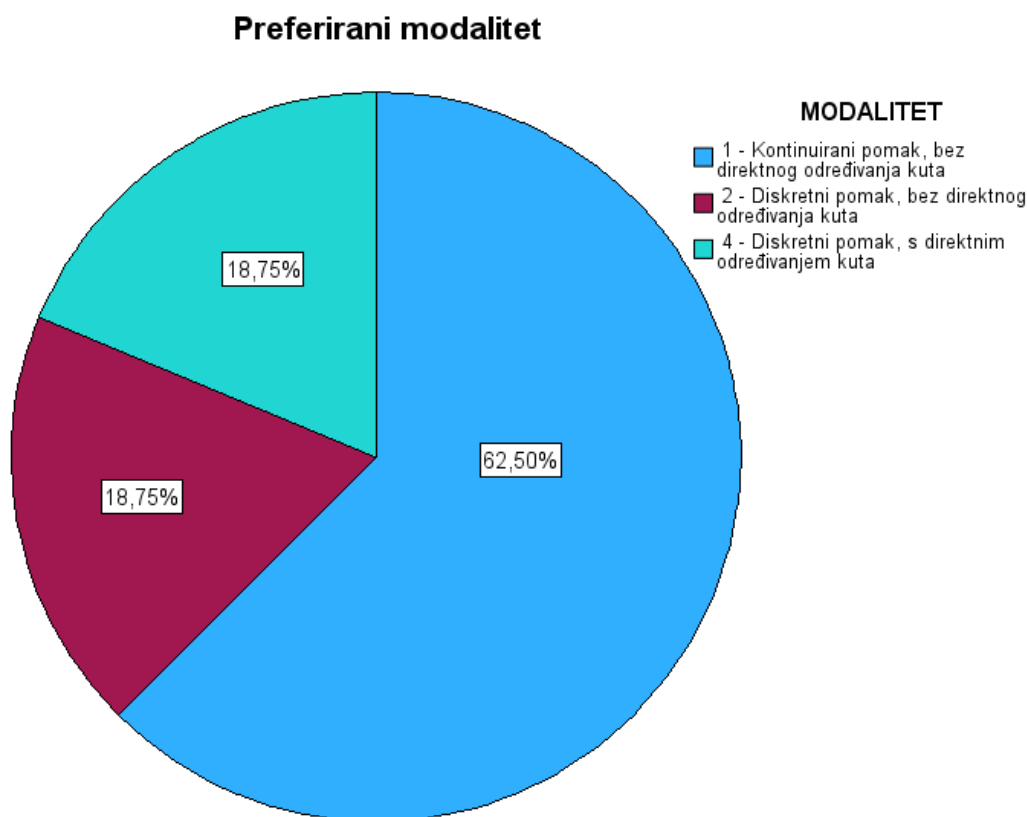
Poglavlje 5. Analiza i rezultati

- MODALITET 2 ($5,4 \pm 0,81$) vs. MODALITET 1 ($5,8 \pm 1,05$) s rezultatom $Z = 0,959$, $p = 1,000$

Rezultati ističu statistički značajne razlike ($p < 0.05$) između kombinacija modaliteta 2 i 3, 1 i 3 te 1 i 4.

5.3.3 Završni upitnik

U završnom upitniku ispitanici su morali odgovoriti na pitanje koji bi modalitet najviše preferirali koristiti. Na grafu na slici 5.11 vidljivi su njihovi odabiri. Modalitet koji ispitanici najviše preferiraju je modalitet 1 (62.5 %), što je bilo i očekivano s obzirom na rezultate istraživanja. Također nije iznenađujuće što nijedan korisnik nije odabrao modalitet 3 s obzirom na rezultate eksperimenta i upitnika. Modaliteti 2 i 4 jednako su zastupljeni (18.75 %), iako bi se po rezultatima moglo očekivati da će modalitet 2 biti više preferiran, no ako korisnik ima dobru mogućnost procjene kuta i udaljenosti od mete, onda mu je modalitet 4 možda i bolji za koristiti.



Slika 5.11 Odabir preferiranog modaliteta

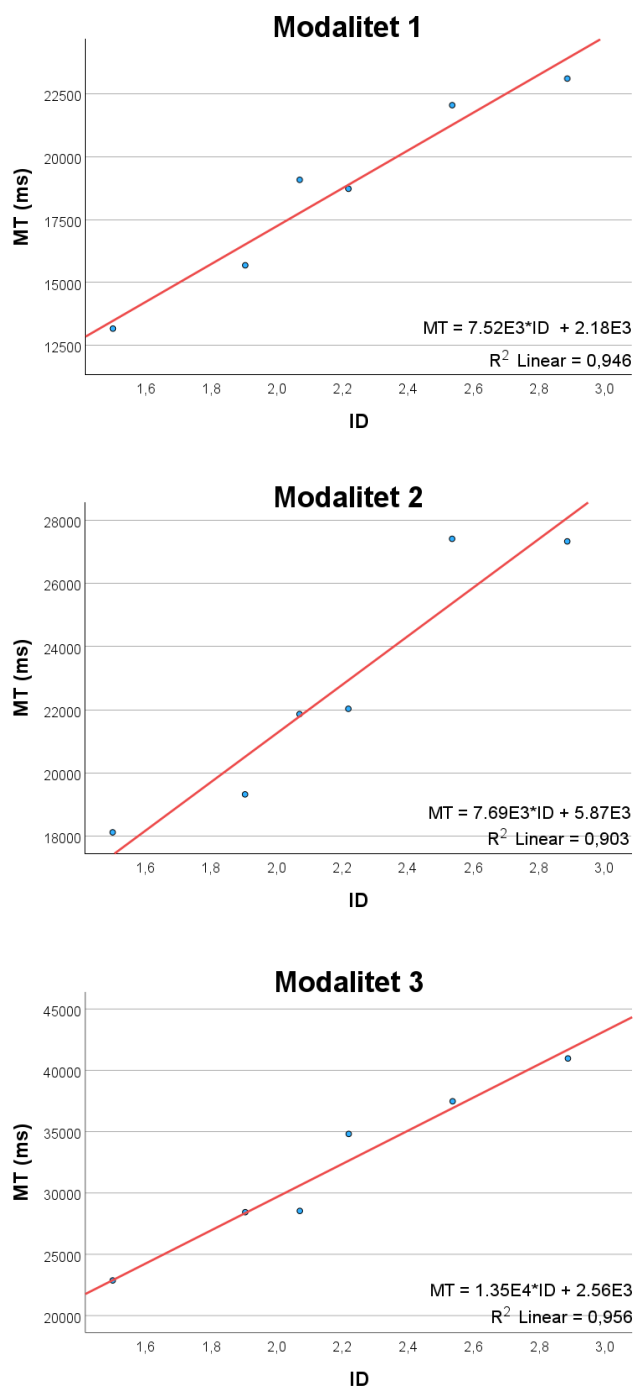
5.4 Prediktivni modeli

U ovom dijelu razmatraju se prediktivni modeli temeljeni na karakteristikama Fittsovog zakona. Za svaki modalitet interakcije zasnovan na glasovnom upravljanju izrađen je graf koji prikazuje potrebno vrijeme (MT) za obavljanje zadataka s poznatim indeksom težine (ID). Pomoću ovih podataka konstruirana je linija linearne regresije, sa svrhom demonstriranja koliko se dobro testirani modaliteti mogu opisati Fittsovim zakonom. Osim četiri modaliteta testirana s više ispitanika, predstavljeni su i prediktivni modeli za modalitete 5 i 6, iako su oni temeljeni samo na testiranju jednog ispitanika koji je imao dosta prakse s njima kroz aplikaciju FittsStudy i nisu jednako robusni kao ostali. Relevantni grafovi dostupni su na slikama 5.12 i 5.13.

Grafovi ilustriraju jednadžbe linearne regresije uz prateću statističku mjeru R^2 , koja ocjenjuje kvalitetu regresijskog modela. Linija regresije služi kao prediktivni model te označava očekivano vrijeme potrebno za završetak zadatka s određenim indeksom težine. Manji koeficijent nagiba ukazuje na sporiji porast vremena potrebnog za složenije zadatke, čineći taj modalitet učinkovitijim.

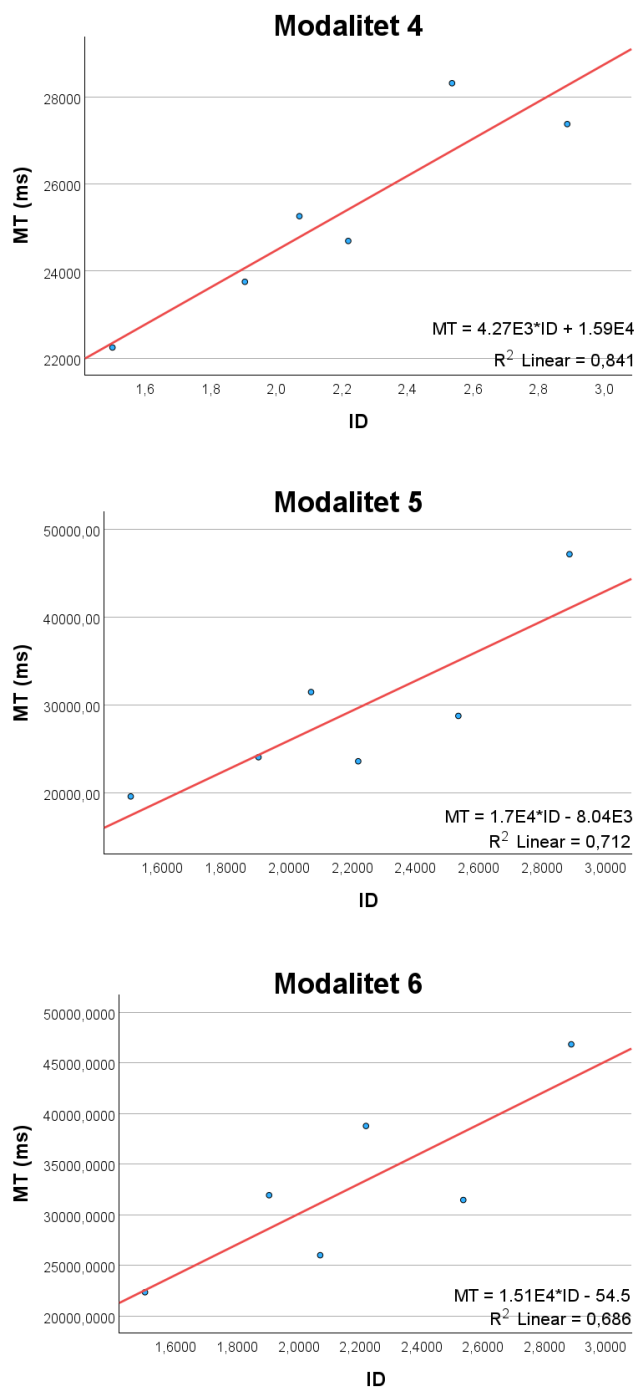
Grafovi potvrđuju primjenjivost Fittsovog zakona na analizirane modalitete, naglašavajući modalitete 1 i 3 kao najbolje usklađene s tim zakonom. Ovo se može objasniti time što je za modalitete s kontinuiranim pomakom izazovnije precizno ciljati manje mete zbog njihove inherentne nepreciznosti u usporedbi s modalitetima s diskretnim pomakom. Također, što je meta udaljenija, korisniku će trebati više vremena da je nacilja. Modaliteti s diskretnim pomakom teoretski bi mogli omogućiti korisnicima, uz dovoljno iskustva u njihovoj uporabi, da u istom vremenskom intervalu ciljaju kako bliže, tako i udaljenije mete.

Poglavlje 5. Analiza i rezultati



Slika 5.12 Grafovi ovisnosti vremena izvršavanja zadataka o indeksu težine zadatka za modalitete 1, 2 i 3

Poglavlje 5. Analiza i rezultati



Slika 5.13 Grafovi ovisnosti vremena izvršavanja zadataka o indeksu težine zadatka za modalitete 4, 5 i 6

Poglavlje 6

Zaključak

S ciljem pružanja alternativnog načina upravljanja pokazivačem miša putem glasovnih naredbi, razvijeno je šest jedinstvenih modaliteta za glasovno upravljanje pokazivačem. Svaki modalitet se razlikuje po načinu pomicanja pokazivača te po načinu određivanja njegova smjera. Eksperimentalno vrednovanje četiri odabrana modaliteta obavljeno je sa 16 sudionika koristeći FittsStudy aplikaciju u kombinaciji s vlastitom testnom aplikacijom.

Na temelju statističke analize podataka, utvrđeno je da modaliteti koji određuju smjer bez potrebe za preciznim određivanjem kuta ostvaruju bolju propusnost i vremensku efikasnost. Nadalje, način pomicanja kursora ne pokazuje statistički značajnu razliku na navedene performanse. Iz subjektivnih povratnih informacija prikupljenih putem upitnika, utvrđeno je da korisnici najviše preferiraju modalitet 1 zbog njegove jednostavnosti i intuitivnosti; koristi se s kratkim i jasnim naredbama koje sustav lako prepoznaje. To potvrđuje početnu pretpostavku da će najintuitivniji i najučinkovitiji modalitet biti onaj s najjednostavnijim naredbama. Da bi rezultati bili pouzdaniji, potrebno je provesti istraživanje s većim uzorkom ispitanika, te u idealnom slučaju podijeliti ih u dvije grupe kako bi se testirali svi modaliteti, a ne samo četiri odabrana.

Realizacija glasovnog upravljanja nudi prostor za unaprjeđenje. Za početak, za svaki modalitet može se istrenirati model koji prepoznaje isključivo njegove specifične glasovne naredbe. Za modalitete koji koriste precizno određivanje kuta pomoću

Poglavlje 6. Zaključak

specijaliziranog pokazivača, mogla bi se razviti verzija s 360 različitih opcija – svaka za kuteve od 0° do 360° s korakom od 1° . To bi značajno povećalo preciznost detekcije smjera. Dodatna korisna funkcionalnost bila bi vizualni prikaz izgovorene naredbe na zaslonu, omogućavajući korisniku da provjeri je li sustav ispravno prepoznao glasovnu naredbu.

Bibliografija

- [1] Human–computer interaction. , s Interneta, https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction , kolovoz 2023.
- [2] A short history of speech recognition. , s Interneta, <https://sonix.ai/history-of-speech-recognition> , kolovoz 2023.
- [3] What is speech recognition? , s Interneta, <https://www.ibm.com/topics/speech-recognition> , kolovoz 2023.
- [4] Python. , s Interneta, <https://www.python.org/> , kolovoz 2023.
- [5] Python SpeechRecognition knjižnica. , s Interneta, <https://pypi.org/project/SpeechRecognition/> , kolovoz 2023.
- [6] Python multiprocessing knjižnica. , s Interneta, <https://docs.python.org/3/library/multiprocessing.html> , kolovoz 2023.
- [7] Python mouse knjižnica. , s Interneta, <https://pypi.org/project/mouse/> , kolovoz 2023.
- [8] Python keyboard knjižnica. , s Interneta, <https://pypi.org/project/keyboard/> , kolovoz 2023.
- [9] Python ctypes knjižnica. , s Interneta, <https://docs.python.org/3/library/ctypes.html> , kolovoz 2023.
- [10] Python tkinter knjižnica. , s Interneta, <https://docs.python.org/3/library/tkinter.html> , kolovoz 2023.
- [11] vosk. , s Interneta, <https://alphacephei.com/vosk/> , kolovoz 2023.
- [12] Axialis CursorWorkshop. , s Interneta, <https://www.axialis.com/cursorworkshop/> , kolovoz 2023.

Bibliografija

- [13] Fitts's law. , s Interneta, https://en.wikipedia.org/wiki/Fitts%27s_law , kolovoz 2023.
- [14] FittsStudy. , s Interneta, <https://depts.washington.edu/accelab/proj/fittsstudy/index.html> , kolovoz 2023.
- [15] ISO 9241-9. , s Interneta, <https://www.iso.org/standard/30030.html> , rujan 2023.
- [16] NASA-TLX. , s Interneta, <https://en.wikipedia.org/wiki/NASA-TLX> , kolovoz 2023.
- [17] IBM SPSS Statistics. , s Interneta, <https://www.ibm.com/products/spss-statistics> , kolovoz 2023.

Sažetak

U ovome radu su predstavljeni i implementirani alternativni modaliteti interakcije putem glasovnih naredbi za upravljanje pokazivačem miša. Operacija pokazivanja-i-odabira na zaslonu stolnog računala omogućena je beskontaktnom interakcijom koristeći predložene modalitete glasovnog upravljanja. Modaliteti se međusobno razlikuju po metodi kretanja pokazivača (kontinuirano ili diskretno) te po načinu određivanja smjera pokazivača (direktnim ili indirektnim određivanjem kuta kretanja). Eksperimentalno vrednovanje implementiranih modaliteta provedeno je s naglaskom na standardnu metriku propusnosti, vrijeme izvršavanja zadatka, upotrebljivost te radno opterećenje interakcije. Dodatni aspekt rada usmjeren je na provjeru primjenjivosti Fittsovog zakona na operaciju pokazivanja-i-odabira u kontekstu glasovnog upravljanja.

Ključne riječi — interakcija čovjeka i računala, operacija pokazivanja-i-odabira, glasovno upravljanje, beskontaktna interakcija, propusnost

Abstract

In this thesis, alternative interaction modalities in which the mouse pointer is controlled by voice commands are presented and implemented. Point-and-select operation on the desktop screen is enabled by contactless interaction with the proposed voice control modalities. The modalities differ from each other by the way of moving the cursor (continuous or discrete) and by the way of determining the direction of the cursor (direct or indirect determination of the angle of movement). Experimental evaluation of the implemented modalities has been performed focusing on standard throughput metrics, task execution time, usability, and interaction workload. Another aspect of the work aims to verify the applicability of Fitts' law to point-and-select operation in the context of voice control.

Keywords — **Human-Computer Interaction, point-and-select, voice control, contactless interaction, throughput**

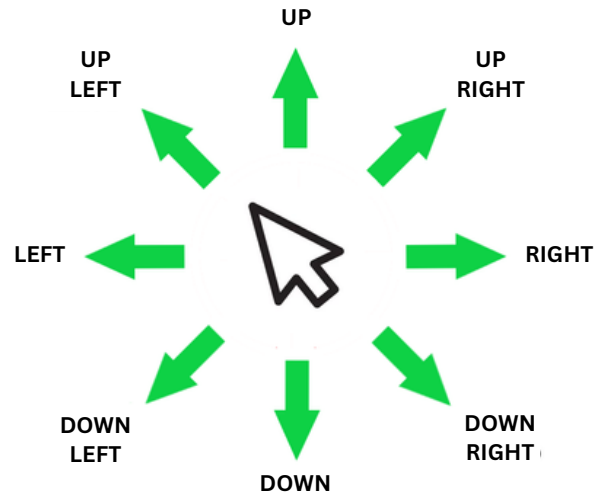
Dodatak A

Grafički prikaz modaliteta za eksperiment

MODALITETI

MODALITET 1

- kontinuirano pomicanje kursora
- **NAREDBE:** UP, DOWN, LEFT, RIGHT, UP LEFT, UP RIGHT, DOWN LEFT, DOWN RIGHT, STOP, CLICK

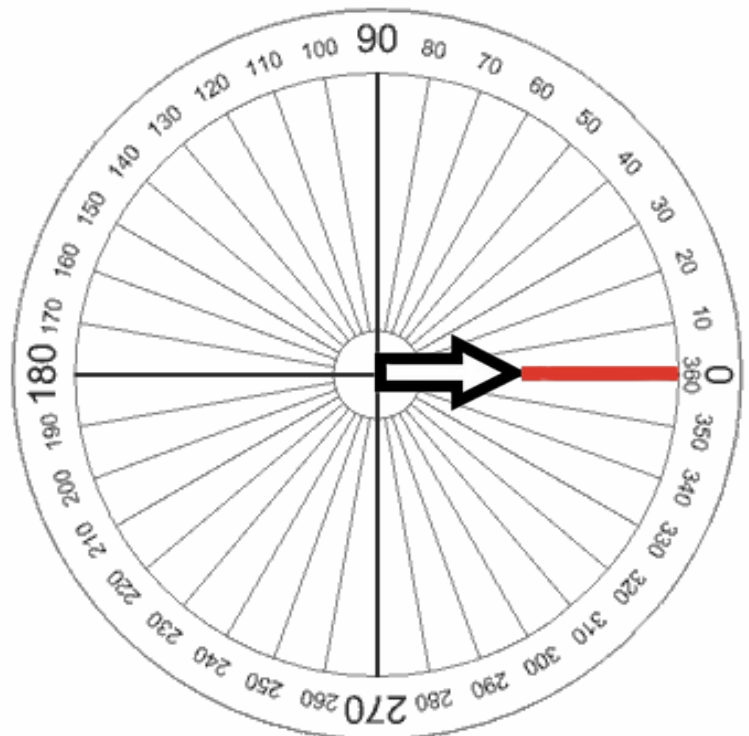


MODALITET 2

- diskretno pomicanje kursora
- **NAREDBE:**
 - {UP, DOWN, LEFT, RIGHT, UP LEFT, UP RIGHT, DOWN LEFT, DOWN RIGHT} + broj piksela (npr. UP 50)
 - CLICK

MODALITET 3

- kontinuirano pomicanje kursora
- **NAREDBE:**
 - GO, STOP, CLICK
 - broj stupnjeva



MODALITET 4

- diskretno pomicanje kursora
- **NAREDBE:**
 - CLICK
 - GO + broj piksela
 - broj stupnjeva