

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij računarstva

Diplomski rad

Paleta komandi svjesna aplikacije

Rijeka, veljača 2019.

Tomislav Milanović
0069069002

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Diplomski sveučilišni studij računarstva

Diplomski rad

Paleta komandi svjesna aplikacije

Mentor: doc. dr. sc. Sandi Ljubić

Rijeka, veljača 2019.

Tomislav Milanović
0069069002

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Povjerenstvo za završne ispite preddiplomskog sveučilišnog studija računarstva
Klasa: 602-04/17-04/18
Ur. br.: 2170-15-11-17-2
Rijeka, 23.3.2017.

ZADATAK za završni rad

Pristupnik: **Tomislav Milanović**

JMBAG: 0069069002

Naslov zadatka: **Implementacija algoritma za praćenje zrake svjetlosti**

Thesis title: **Implementation of ray tracing algorithm**

Polje znanstvenog područja: **Računarstvo**

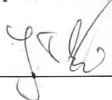
Grana znanstvenog područja: **Programsko inženjerstvo**

Sadržaj zadatka: Implementirati algoritam praćenja zrake svjetlosti u svrhu realističnog prikaza stacionarne scene. Voditi računa o različitim modelima sjenčanja, materijalu objekata u sceni, alias učinku te broju i tipu izvora svjetlosti. Konačno, potrebno je implementirati hijerarhijsku strukturu omeđujućih volumena, čime je moguće znatno ubrzati izvršavanje algoritma.

Zadano: 23.3.2017.

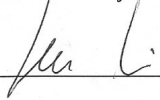
Mentor

izv. prof. dr. sc. Jerko Škifić

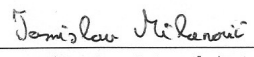


Predsjednik Povjerenstva

izv. prof. dr. sc. Kristijan Lenac



Zadatak preuzeo dana: 27.3.2017.



(potpis pristupnika)

Završni rad mora biti pisan prema uputama koje su objavljene na web stranicama studija.

Dostaviti:

- Pristupnik (x1), Mentor (x1), Studentska služba (x1), Predsjednik Povjerenstva (x1)

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, veljača 2019.

Tomislav Milanović

Zahvala

Ovo je zahvala.

Sadržaj

Popis slika	vii
1 Uvod	1
1.1 Paleta komandi	1
1.2 Evaluacija univerzalne palete komandi	2
2 Realizacija univerzalne palete komandi	4
2.1 Dizajn	4
2.2 Dijagram toka	6
2.3 Implementacija	8
2.4 Prezentacija	9
2.5 Izazovi i moguća poboljšanja	12
2.5.1 Izazovi podrške funkcionalnostima	12
2.5.2 Izazovi ispravne detekcije programa	12
Bibliografija	14
Pojmovnik	15
Sažetak	16

Popis slika

1.1	Paleta komandi u programu Atom	2
2.1	Skica željenog izgleda univerzalne palete komandi	4
2.2	Univerzalna paleta komandi je svjesna aplikacije koju korisnik trenutno upotrebljava	5
2.3	Dijagram toka rada univerzalne palete komandi	7
2.4	Konačni izgled univerzalne palete komandi	10
2.5	Testni način rada	11

Poglavlje 1

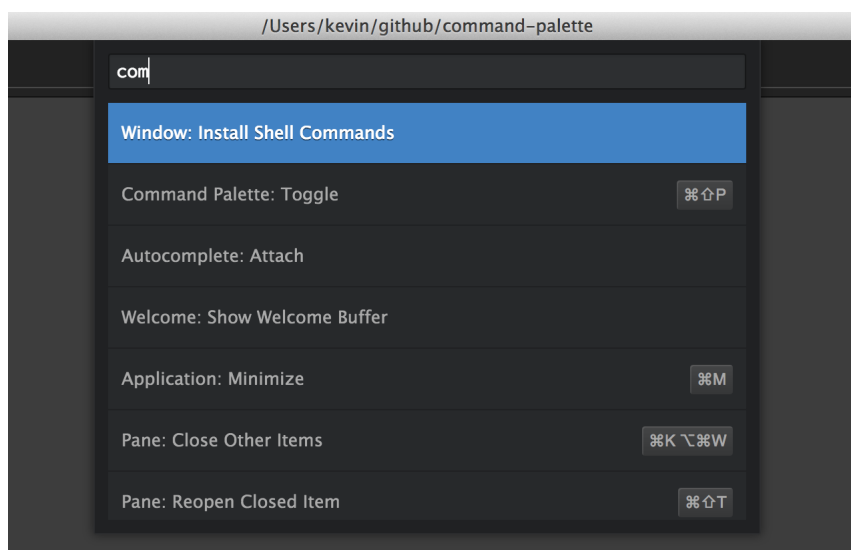
Uvod

Interakcija čovjeka i računala (eng. *HCI (Human-computer interaction)*) istražuje dizajn i korištenje računalne tehnologije, odnosno sučelja između ljudi (korisnika) i računala. Uspješno sučelje za interakciju s računalom ima za cilj postići dobru upotrebljivost (eng. *usability*), koja uključuje olakšan i ubrzan rad te jednostavno savladavanje uporabe samog korisničkog sučelja. Intuitivnost se postiže korištenjem koncepata i mehanizama koji su potencijalnim korisnicima već otprije poznati kroz iskustvo rada s ostalim korisničkim sučeljima.

U ovom radu, fokus je usmjeren na izradi i evaluaciji računalnog programa (**univerzalna paleta komandi**) koji nastoji omogućiti korisniku jednostavan pristup funkcionalnostima **bilo kojeg** drugog programa uz pomoć pretrage po ključnim riječima, umjesto traženja po hijerarhiji izbornika ili ostalim dijelovima sučelja tog programa. Poseban je naglasak stavljen na brzi dohvat rjeđe korištenih funkcionalnosti, za koje korisnik možda ne zna napamet gdje se nalaze.

1.1 Paleta komandi

Paleta komandi (eng. *command palette*) predstavlja listu dostupnih komandi koje se mogu izvršiti unutar programa i polje za unos ključnih riječi za pronalazak željene komande. Na slici 1.1 [1] prikazan je mogući izgled palete u programu *Atom* (uređivač teksta i programskog koda) kada se unese "com". Korisnik može odabrati i izvršiti



Slika 1.1 Paleta komandi u programu Atom

jednu od ponuđenih komandi. Paleta se poziva korištenjem tipkovničkog prečaca *Cmd+Shift+P* na macOS, odnosno *Ctrl+Shift+P* na Linux i Windows operacijskim sustavima.

Dodatna prednost palete komandi je smanjivanje potrebe za prebacivanjem između tipkovnice i miša (eng. *homing* - prebacivanje s jednog na drugi uređaj za upravljanje računalom). U ovom radu implementirana je univerzalna paleta komandi, koja se može koristiti u bilo kojem programu jer je svjesna aplikacije za koju se pokreće.

1.2 Evaluacija univerzalne palete komandi

Upotrebljivost univerzalne palete komandi ispituje se na temelju unaprijed definiranih zadataka koji uključuju rad s popularnim programima:

- **Google Chrome** - web preglednik
- **LibreOffice Writer** - uređivač teksta
- **GIMP** - uređivač slika

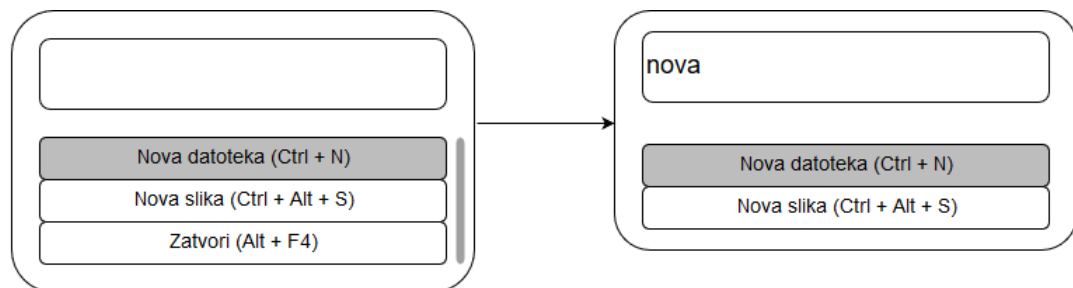
Poglavlje 1. Uvod

Zadaci su osmišljeni tako da predstavljaju uobičajeni, svakodnevni rad na računalu. Sastoje se od kombinacije češće i rjeđe korištenih funkcionalnosti navedenih programa. Uspoređuje se vrijeme potrebno za obavljanje svih zadataka sa i bez univerzalne palete komandi. Također, preko predispitnih i postispitnih anketa želi se dobiti uvid u kojoj mjeri ispitanici koriste tipkovničke prečace i palete komandi iz programa koji ih podržavaju te na kraju njihovo zadovoljstvo s univerzalnom paletom komandi.

Poglavlje 2

Realizacija univerzalne palete komandi

2.1 Dizajn

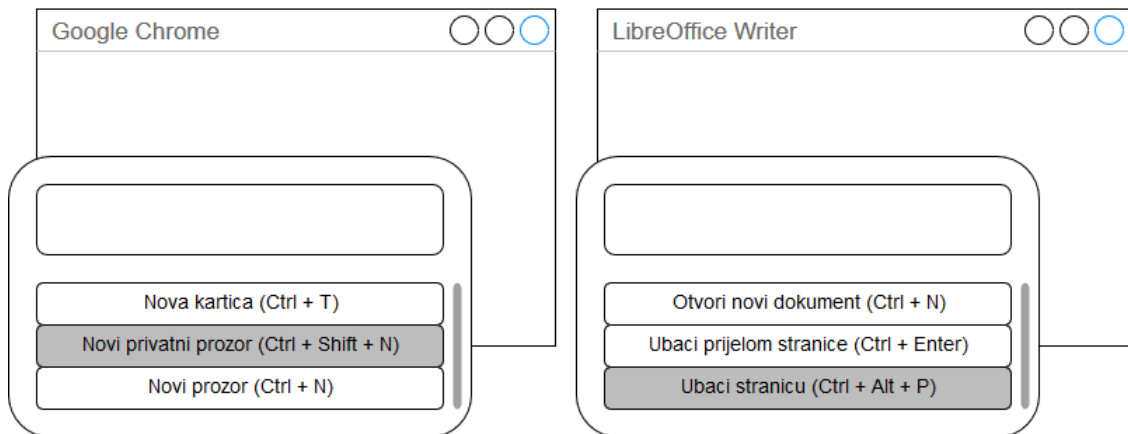


Slika 2.1 Skica željenog izgleda univerzalne palete komandi

Na slici 2.1 prikazan je željeni izgled univerzalne palete komandi koji se sastoji od dvije osnovne cjeline: polja za unos teksta i liste komandi. Pritom, maksimalni broj vidljivih komandi u listi će se moći podesiti (početna postavka je deset komandi). Ukoliko se u listi komandi nalazi više stavki od tog broja, pojavljuje se vertikalna traka za pomicanje (eng. *scrollbar*) kako bi korisnici mogli imati pristup svim komandama. Uvijek je označena samo jedna komanda a unosom određenog pojma prikazuje se lista koja odgovara tom pojmu.

Poglavlje 2. Realizacija univerzalne palete komandi

Paletom će se moći upravljati s tipkovnicom (koristeći strelice za gore i dolje te *Enter* za odabir komande) ili s mišem (korištenjem trake za pomicanje i klikanjem na željenu komandu).



Slika 2.2 Univerzalna paleta komandi je svjesna aplikacije koju korisnik trenutno upotrebljava

Glavna značajka univerzalne palete komandi je njena svjesnost aplikacije koju korisnik trenutno upotrebljava, što je vidljivo na slici 2.2 (prikazane komande su samo za demonstraciju koncepta). Ovisno o aplikaciji, izbor komandi se razlikuje.

Dodatno, univerzalna paleta komandi će pamtit i frekvenciju korištenja svake komande, kako bi one koje se najčešće koriste bile prikazane na vrhu liste. Nakon iznesenih zahtjeva, nameću se sljedeći izazovi:

- Kako prepoznati aplikaciju koja se trenutno koristi?
- Koji je univerzalan način na koji neki program može izvršiti bilo koju radnju u nekom drugom programu?

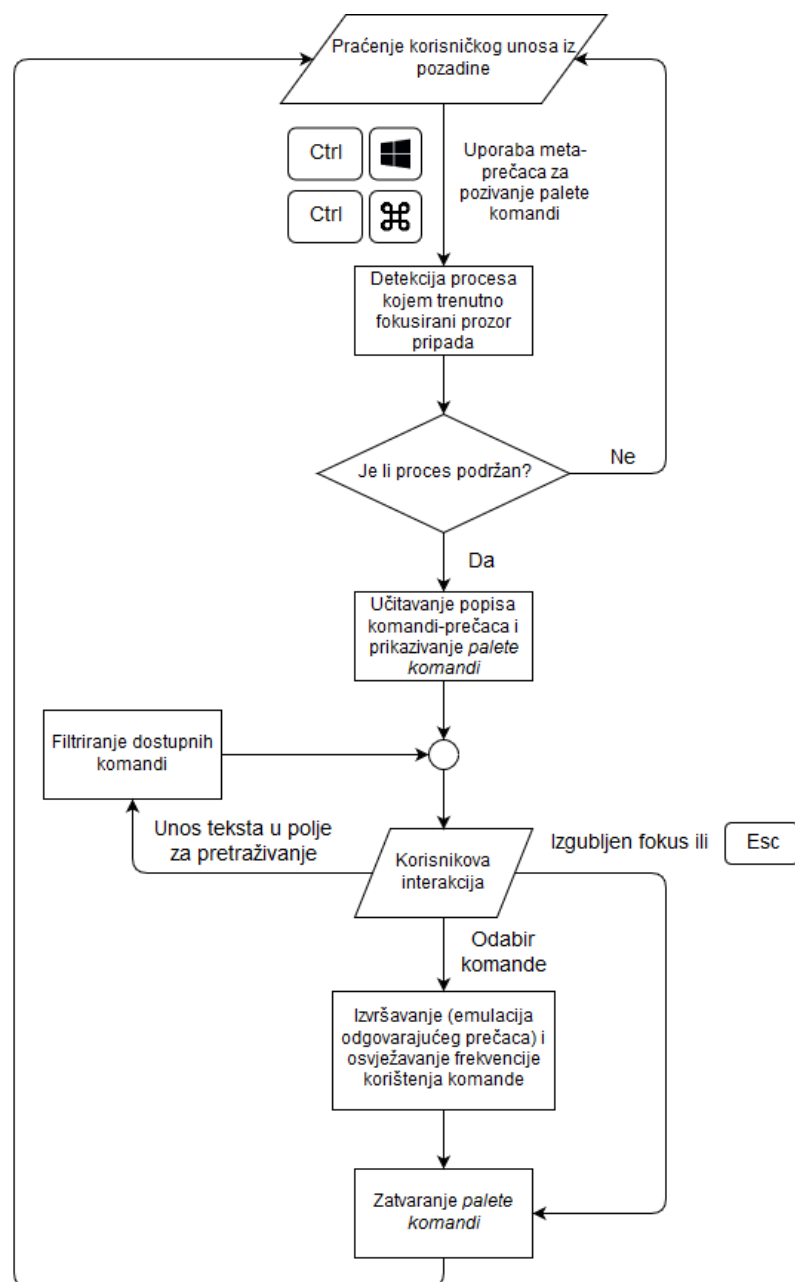
Prvi problem je riješen tako da se koristi alat za prepoznavanje procesa kojem trenutno fokusirani prozor pripada te se u skladu s time učitava odgovarajuća lista komandi. Kod drugog izazova, izvršavanje komandi se postiže emulacijom tipkovničkih prečaca. Primjerice, kod izvršavanja komande *Otvori datoteku*, univerzalna paleta će emulirati tipkovnički prečac *Ctrl + O*.

Tipkovnički prečaci su koncept koji je široko rasprostranjen u softveru na klasičnim, *desktop* računalima, gdje se koriste miš i tipkovnica. Gotovo da nema aplikacije koja ih ne koristi kako bi svojim korisnicima omogućila ubrzani rad s aplikacijom. Zbog toga, emuliranje prečaca donosi **univerzalnost** ovom programu.

2.2 Dijagram toka

Cjelokupni način rada univerzalne palete komandi prikazan je na slici 2.3. Većinu vremena, ovaj program će provesti kao pozadinski proces (eng. *daemon*) te će osluškivati korisnikove unose na tipkovnici. U trenutku kada korisnik upotrijebi *meta-prečac*, koji je u ovom radu podešen na *Ctrl + Win*, program će provjeriti prvo je li aplikacija koji se trenutno koristi podržana. Ako nije, neće se ništa dogoditi, a u suprotnom slučaju, učitat će se lista komandi i paleta će se prikazati na ekranu.

Korisnik može pretražiti i izvršiti željenu komandu ili može zatvoriti paletu korištenjem tipke *Esc* ili gubitkom fokusa na paletu (klikom izvan njenog prozora).



Slika 2.3 Dijagram toka rada univerzalne palete komandi

2.3 Implementacija

Univerzalna paleta komandi implementirana je za sustave bazirane na Linuxu. Konkretno, koristila se Antergos distribucija s **KDE** okolišem. Temelji se na programskom jeziku Python 3.7.2 i sljedećim modulima:

- *Tkinter* - *de-facto* standardni paket za razvoj GUI aplikacija, temeljen na **Tcl/Tk** alatima [2]. Služi za implementaciju korisničkog sučelja univerzalne palete komandi.
- *PyAutoGUI* - modul za programatsko izvršavanje (emulaciju) uporabe miša i tipkovnice. Koristi se za emulaciju tipkovničkih prečaca, čime univerzalna paleta komandi može izvršiti neku radnju u bilo kojoj drugoj aplikaciji.
- *pynput* - knjižnica funkcija za osluškivanje korisnikovog unosa na tipkovnici. Omogućava uporabu meta-prečaca za pozivanje univerzalne palete komandi.

Za prepoznavanje aplikacije koju korisnik trenutno upotrebljava koristi se Linux alat *xdotool* [3], uz čiju se pomoć može saznati PID (eng. *Process ID*) trenutno korištene aplikacije. S tom informacijom, od poznate *shell* naredbe *ps* doznajemo ime procesa s tim PID-om.

Komande se spremaju kao JSON datoteke za svaku aplikaciju zasebno. Nazivi tih datoteka odgovaraju imenima procesa tih aplikacija u Linux sustavima.

U isječku koda 2.1 možemo vidjeti kako bi izgledao JSON zapis komandi za program Google Chrome kada bismo imali samo dvije stavke. Pri tome, važno je napomenuti kako oznake za tipke koje čine određeni tipkovnički prečac (*ctrl*, *shift*, *del* i slični) moraju odgovarati oznakama koje se pojavljuju u dokumentaciji za modul *PyAutoGUI* [4].

Prvo polje (u ovom primjeru *Bookmark* i *Clear Browsing Data*) predstavlja ime komande, odnosno ono što će se korisniku univerzalne palete komandi prikazati na ekranu.

Polje *frequency* označava koliko je puta komanda bila korištena otkad je korisnik počeo upotrebljavati univerzalnu paletu komandi. Može se zaključiti kako je jednostavno podešavati tu vrijednost, ukoliko postoji potreba za time.

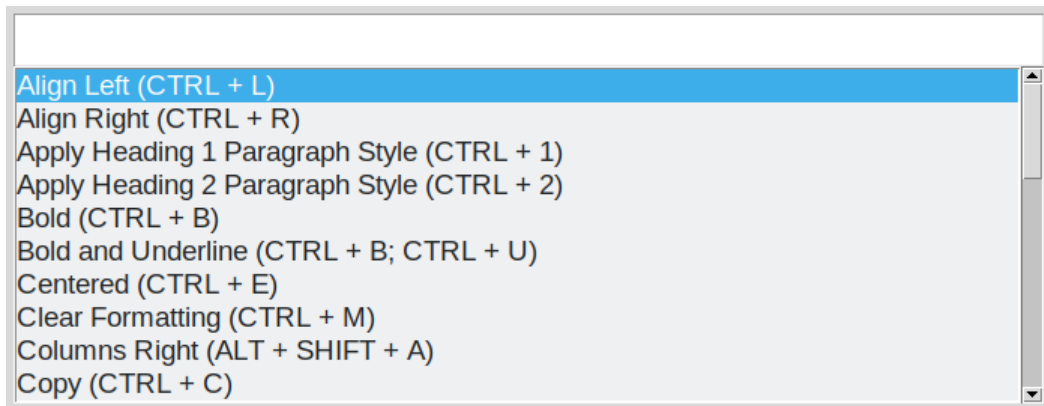
Kod 2.1 Pokazni primjer moguće JSON datoteke za konfiguraciju komandi u programu Google Chrome

```
1 {
2   "Bookmark": {
3     "type": "shortcut",
4     "shortcuts": [
5       [
6         "ctrl",
7         "d"
8       ]
9     ],
10    "frequency": 0
11  },
12  "Clear Browsing Data": {
13    "type": "shortcut",
14    "shortcuts": [
15      [
16        "ctrl",
17        "shift",
18        "del"
19      ]
20    ],
21    "frequency": 0
22  }
23 }
```

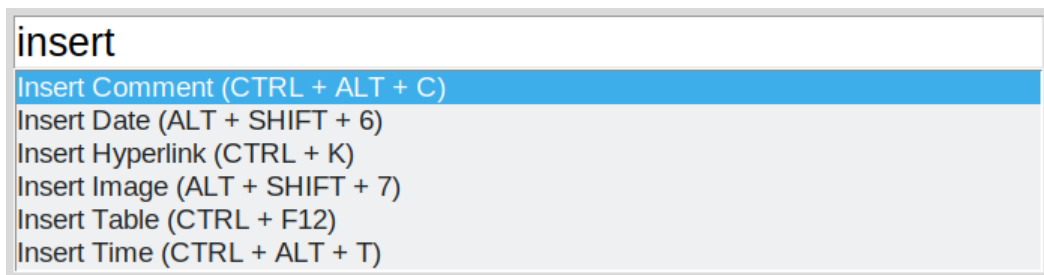
Sve u svemu, JSON predstavlja prikladan format za zapis podataka ovakvog tipa, jer je jednostavan za uporabu i jasno čitljiv ljudima.

2.4 Prezentacija

Konačni izgled univerzalne palete komandi možemo vidjeti na slikama 2.4. Svaka komanda pored svog naziva ima i odgovarajući prečac, što je uvedeno u nadi da će korisnik kroz rad upamtiti odgovarajuće prečace te ih s vremenom početi i koristiti, kako bi u kraćem roku dolazio do funkcionalnosti koje mu trebaju.



(a) Bez unosa



(b) S unosom

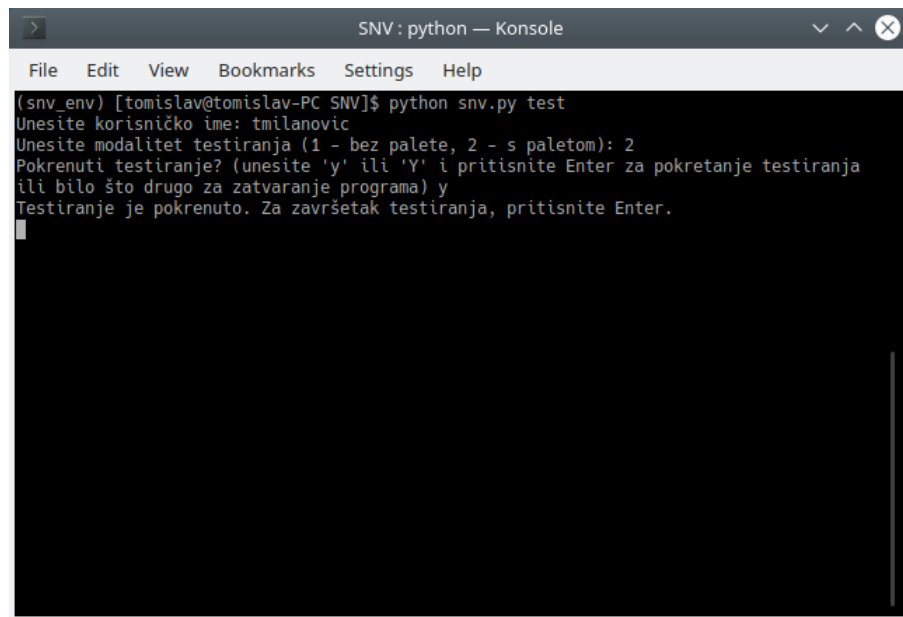
Slika 2.4 Konačni izgled univerzalne palete komandi

Glavni jezik programa je engleski, jer je pretpostavka da će ispitanici (studenti računarstva i profesori Zavoda za računarstvo) biti bolje upoznati s radom na softveru koji je podešen za originalni, engleski jezik.

Dodatno, u svrhu provođenja eksperimenta i ispitivanja upotrebljivosti univerzalne palete komandi uveden je i **testni način rada**, što se može vidjeti na slici 2.5. Predviđena su dva modaliteta testiranja, gdje je u jednom modalitetu korištenje palete dano na raspolaganje a u drugom nije (ovdje se program pokreće samo za svrhu mjerenja vremena obavljanja ispitnog zadatka).

Nakon završetka testiranja (pritiskom na tipku *Enter*) stvorit će se tekstualna datoteka s imenom oblika *korisnickoime_modalitet_report.txt*, čiji je primjer dan u isječku 2.2. Ovdje se nalazi popis svih komandi koje su bile upotrijebljene za vrijeme testiranja (ime Linux procesa za kojeg su bile iskorištene, njihovo ime i vremenski

Poglavlje 2. Realizacija univerzalne palete komandi



Slika 2.5 Testni način rada

Kod 2.2 Primjer stvorenog izvještaja eksperimenta (datoteka *tmilanovic_2-report.txt*)

```
Username: tmilanovic
Start Timestamp: 1358.316003

Commands:
soffice.bin: Align Left — 1365.989010
soffice.bin: Apply Heading 2 Paragraph Style — 1367.789081
soffice.bin: Find and Replace — 1370.069251

End Timestamp: 1373.265085

Number of executed commands: 3
Measured time: 14.949082 s
```

trenutak u kojem su izvršene).

Vremenske pečate (eng. *timestamps*) ispisuje Python metoda *time.monotonic()* [5]. Definirani su u sekundama i nemaju referentnu točku već je potrebno gledati

razliku između dva pečata za mjerenje vremena. Glavna karakteristika *monotonic* satova je nemogućnost da idu nazad, odnosno na njih ne utječe promjena sistemskog sata.

Izveštaj za prvi modalitet rada (bez palete) razlikuje se u izostanku popisa komandi.

2.5 Izazovi i moguća poboljšanja

2.5.1 Izazovi podrške funkcionalnostima

Nisu sve funkcionalnosti koje se mogu koristiti u nekoj aplikaciji pokrivene odgovarajućim tipkovničkim prečacem. Međutim, programi u svojim postavkama često imaju mogućnost podešavanja tipkovničkih prečaca za one funkcionalnosti koje ga inače nemaju. Na taj način može se ostvariti proširenje opsega funkcionalnosti koje se mogu izvršiti uz pomoć univerzalne palete komandi. Dodatna prednost kod takvih programa bila bi mogućnost spremanja postavki tipkovničkih prečaca u datoteke koje se zatim mogu dijeliti sa zajednicom korisnika tog programa.

Programi često sadržavaju mnogobrojne funkcionalnosti i mogućnosti, što znači da izrada JSON datoteke za univerzalnu paletu komandi može iziskivati određeno vrijeme i trud. Srećom, tu je datoteku dovoljno napisati samo jednom, a zatim se ona može dijeliti unutar potencijalne zajednice korisnika univerzalne palete komandi. Kako bi proces pisanja JSON datoteke bio pristupačan korisnicima (eng. *user friendly*), postoji opcija izrade aplikacije koja bi služila uređivanju takvih datoteka, bez potrebe da se direktno uređuje JSON zapis za njih.

2.5.2 Izazovi ispravne detekcije programa

Prema iznesenom dizajnu univerzalne palete komandi, detekcija aplikacije koju korisnik trenutno upotrebljava vrši se uz pomoć imena procesa u Linux sustavu. Tijekom implementacije podrške za program *LibreOffice Writer* uočeno je da svi programi u *LibreOffice* softverskom paketu imaju identično ime procesa: *soffice.bin*.

To predstavlja problem prilikom učitavanja odgovarajuće JSON datoteke ako se želi ostvariti podrška za sve programe koji čine taj paket.

Ovaj problem se može riješiti tako da se u razmatranje prilikom detekcije programa uzme i naslov prozora. Oni se prilično razlikuju, najčešće sadržavaju ime otvorene datoteke ili u slučaju preglednika interneta i ime stranice koju korisnik trenutno pretražuje. Važno je u tom slučaju programski obraditi taj naslov tako da se iz njega iščita ime aplikacije, koje se najčešće nalazi na kraju naslova prozora, odvojeno sa crticom (znak "-" ili "--", kratka i duga crtica).

Na kraju, treba uzeti u obzir i činjenicu kako se s vremenom pojavljuju nove verzije aplikacija, koje mogu imati nove funkcionalnosti i samim time potencijalne nove komande za paletu. U Linux sustavima se verzija često može dobiti pokretanjem komande koja odgovara programu u konzoli s argumentom "-v" ili "--version". U Windows sustavima se ta informacija može dobiti iz izvršnih datoteka programa.

Ako se do te informacije ne može doći programatski, preostaje ponuditi korisniku izbor odgovarajuće verzije preko postavki univerzalne palete komandi.

Bibliografija

- [1] Atom, Command Palette package. , s Interneta, <https://atom.io/packages/command-palette> , 03.03.2019.
- [2] TkInter, Python. , s Interneta, <https://wiki.python.org/moin/TkInter> , 04.03.2019.
- [3] xdotool. , s Interneta, <https://github.com/jordansissel/xdotool> , 04.03.2019.
- [4] PyAutoGUI, Dokumentacija. , s Interneta, <https://pyautogui.readthedocs.io/en/latest/keyboard.html#keyboard-keys> , 04.03.2019.
- [5] Time Monotonic, Python 3 Dokumentacija. , s Interneta, <https://docs.python.org/3/library/time.html#time.monotonic> , 07.03.2019.

Pojmovnik

AACP Application-Aware Command Palette. 3

HCI Human-computer interaction. 1

Sažetak

Hrvatski sažetak.

Ključne riječi — interakcija čovjeka i računala

Abstract

English abstract.

Keywords — human-computer interaction