

Tipkovnice i dinamička prilagodba

Plavec, Mateo

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:168:320633>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1338

TIPKOVNICE I DINAMIČKA PRILAGODBA

Mateo Plavec

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1338

TIPKOVNICE I DINAMIČKA PRILAGODBA

Mateo Plavec

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1338

Pristupnik: **Mateo Plavec (0036544036)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Dragan Jevtić

Zadatak: **Tipkovnice i dinamička prilagodba**

Opis zadatka:

Različiti aspekti vezani za unos podataka putem tipkovnica mogu se razmatrati s ciljem općeg poboljšanja, ali i namjenske prilagodbe tipkovnica. Općenito se mogu analizirati ili statički i dinamički prilagođavati, ergonomija i udobnost, tipke i prečaci, osvjetljenje vizualizacija, integracija s računalnim okruženjem, dizajn, estetika, trajnost i kvaliteta izrade. Kombinacija navedenih aspekata također djeluje na korisničko iskustvo i zadovoljstvo korisnika tastatura. Vaš je zadatak analizirati postojeća rješenja vizualizacije i prečaca s posebnim naglaskom na mogućnosti automatizacije, odnosno dinamičkog usklađivanja odabranih aspekata tipkovnica. Potrebno je izdvojiti i opisati zapažanja vezano za nedostatke postojećih rješenja te predložiti poboljšanja te izvedbene modele za poboljšanja ili dinamičke namjenske prilagodbe. Verifikaciju provedite na laboratorijskom modelu. Svu potrebnu literaturu i uvjete za rad osigurat će vam Zavod za telekomunikacije.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

Uvod	1
1. Prijašnja istraživanja.....	2
2. Alat	3
2.1. Arhitektura.....	3
2.1.1. Osnovni događaji (razred <code>InputEvent</code>).....	3
2.1.2. Agregirani događaji (razredi <code>CompositeEvent</code> i <code>EventAdapter</code>).....	4
2.1.3. Analiza (razred <code>EventObserver</code>).....	4
2.1.4. Vizualizacija (imenski prostori <code>Graphics</code> i <code>Visual</code>).....	4
2.1.5. Ograničenja.....	5
2.1.6. Primjer uporabe	5
2.1.7. Mogućnosti nadogradnje	6
Zaključak	7
Literatura	8
Sažetak.....	9
Summary.....	10
Privitak	11

Uvod

Tipkovni prečaci su način izdavanja naredbi programu koji se unose uzastopnim pritiskom jedne ili više tipki. [2] Svaki prečac najčešće izdaje jednu naredbu aplikaciji. Ideja ovog rada je dati uvide u efikasne odnosno zadovoljavajuće tipkovne prečace koje korisnici preferiraju koristiti. Zašto se neki prečaci učestalo koriste, u kojem kontekstu.

S obzirom na sastav tipki, odnosno način aktivacije, tipkovne prečace možemo podijeliti u tri kategorije:

- Samo jedna tipka. Na primjer, 'F' za prikaz preko punog zaslona ili 'F5' za osvježavanje prikaza. Ako je aktivno polje za unos teksta, onda neki prečaci iz ove kategorije mogu biti neaktivni.
- Kombinacija tipke i modifikatora. Prečac u užem smislu, na primjer: 'Ctrl-C' za kopiranje.
- Akord (engl. *chord*). Prečac koji se sastoji od više uzastopnih kombinacija. Na primjer, za pokretanje svih testova u *Visual Studio* razvojnom okruženju prvo se pritisne 'Ctrl-R', a potom tipka 'A'. Ovdje spadaju i pristupne tipke (engl. *access keys*). [3]

1. Prijašnja istraživanja

U [1] zaključuju da se tipkovni prečaci rijetko koriste naspram alternativama (gumbi u grafičkom korisničkom sučelju), to je zbog toga što ljudi pronađu metodu koja obavi posao i dovoljno je dobra. No pokazuje se da u nekim slučajevima korisnici ipak češće koriste tipkovne prečace. Naredba 'Traži' se u 90% slučajeva izdaje tipkovnim prečacem, više od svih drugi promatranih naredbi [1]. (Meni je ovaj rezultat bio iznimno zanimljiv.) Ovaj rezultat može se jednostavno objasniti: nakon što izda naredbu 'Traži', korisnik će vrlo vjerojatno unijeti riječi za pretragu upravo tipkovnicom. Stoga je jasno da je dobra opcije ne posezati za mišem, već naredbu izdati tipkovnim prečacem 'Ctrl-F'.

To me navelo da razmislim o tome koje su odrednice dobro dizajniranih tipkovnih prečaca. Specifično vezano uz ovaj rad: koje metrike bi se mogle pratiti.

Na primjer, ako se prečac izdaje korištenjem obje ruke (npr. ako pretpostavimo da se miš koristi desnom rukom, onda je lijeva ruka često na tipkovnici, pripravna za izdavanje prečaca. Tada se prečac 'Ctrl-K' koristi s dvije ruke, jer je neergonomično samo lijevom rukom pritisnuti 'Ctrl' i 'K') i ako korisnik i prije i poslije izdavanja često koristi miš, onda znači da je morao ruku pomaknuti s miša na tipkovnicu. To je onda nepoželjna situacija i taj prečac nije najbolje osmišljen.

2. Alat

U sklopu rada razvijen je alat za analizu uporabe tipkovnih prečaca u kontekstu interakcije s računalom pomoću miša i tipkovnice. Alat prikuplja i agregira osnovne događaje (pomicanje miša, pritisak tipke), analizira te nudi različite poglede na podatke. Pogledi koji su dostupni:

- `EventVisualiser` – Prikazuje popis agregiranih događaja kako dolaze, korišten prilikom razvoja za otklanjanje grešaka.
- `LogEvents` – U zapisnik (`runtime.log`) i na zaslon ('Ctrl-F10' za prikaz) zapisuje događaje, korišten prilikom razvoja za otklanjanje grešaka.
- `ShortcutContext` – Za svaki tipkovni prečac računa frekvencijsku analizu konteksta u kojima se pojavljuje (u ovoj pojednostavljenoj implementaciji kontekstom se smatraju događaji neposredno prije i nakon danog prečaca) te prikazuje prečac, kontekst i frekvenciju na zaslon.
- `ShortcutPairs` – Prikazuje prečace koji se često koriste u paru, sortirano po učestalosti.

Alat je namijenjen da se pokrene i da prikuplja podatke u pozadini dok se računalo normalno koristi.

2.1. Arhitektura

Izvorni kod alata je oblikovan u skladu sa oblikovnim obrascima i načelima. Tako da se bez promjene postojećeg koda lako može dodati novi pogled koji će raditi neku novu analizu.

2.1.1. Osnovni događaji (razred `InputEvent`)

Osnovni događaji su nedjeljivi događaji koji se stvaraju prilikom interakcije korisnika s ulaznim jedinicama (miš i tipkovnica). Specifično, to su klikanje i pomicanje miša, pritisak i otpuštanje tipka na tipkovnici. Alat registrira `SetWindowsHookEx` te od operacijskog sustava prima sve događaje, neovisno o aktivnoj aplikaciji. Kako bi se osigurala responzivnost sustava, osnovni događaji se prikupljaju u zasebnoj dretvi i stavljaju na red čekanja za analizu (razred `EventQueue`). Analiza događaja može biti skup proces, te je nepoželjno da blokira obradu događaja na razini operacijskog sustava.

Osim informacija o samom događaju, prikuplja se vrijeme događaja, aplikacija koja je bila aktivna, te je li događaj umjetno ubačen. Ove informacije su važne za daljnju analizu. Umjetno ubačeni događaji je ubacio neki program treće strane, odnosno nije stvoren izravnom interakcijom korisnika. Takve događaje najčešće ubacuju aplikacije za prilagodbu i automatizaciju tipkovnice, na primjer: *PowerToys Keyboard Manager* [6] ili *AutoHotkey* [7].

2.1.2. Agregirani događaji (razredi `CompositeEvent` i `EventAdapter`)

Osnovi događaji nisu pogodni za direktnu analizu. Zato se prvo agregiraju u kompozitne događaje kojima je lakše baratati. Za agregaciju su zaduženi `EventAdapter`-i. Trenutno implementirani agregirani događaji su:

- Tipkovni prečaci – Agregiraju više tipkovničkih događaja te razumiju razliku između vrsti tipki (modifikatori, tipke za unos teksta).
- Sažeti događaji miša – Sažimaju vremenski bliske događaje miša u jedan događaj radi lakše analize.
- Unos teksta – Sažimaju vremenski bliske događaje unosa teksta (pritisaka određenih tipki) u jedan događaj radi lakše analize.

2.1.3. Analiza (razred `EventObserver`)

Agregirani i sažeti događaji su pogodni za analize koje provode `EventObserver`-i. Temeljem sličnosti i skupom pravila događaji se grupiraju te se izvode analize.

2.1.4. Vizualizacija (imenski prostori `Graphics` i `Visual`)

Rezultati analize se u stvarnom vremenu na interaktivan način prikazuju na zaslonu.

Pojašnjenje boja (Slika 2.1):

- Crveno-narančasta – Uobičajen tipkovni prečac.
- Zatamnjeno crveno-narančasta – Ponavljanje tipkovnog prečaca (uslijed zadržavanja tipke pritisnutom)
- Plavo-zelena – Umjetno ubačeni tipkovni prečac (zatamnjeno – ponavljanje)
- Svijetlo plava – Unos teksta tipkovnicom

- Roza-placa – Pomak i klikanje mišem



Slika 2.1 Primjer vizualizacije frekvencijske analize tipkovnih prečaca i relevantnog konteksta

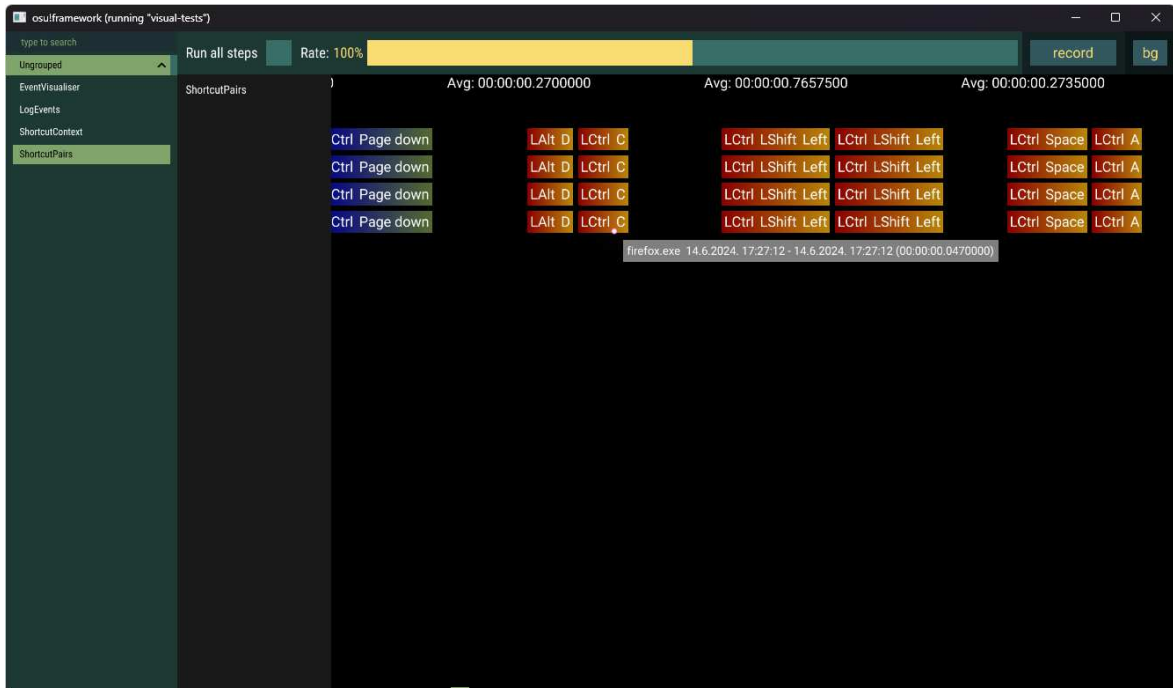
2.1.5. Ograničenja

Zbog jednostavnosti implementacije, razvijen alat bilježi samo prečace koji koriste tipke za modifikiranje ili one koje ne koriste tipke koje se uobičajeno koriste za upis teksta. (Alat ne može razlikovati unos teksta 'f' i tipkovni prečac za prikaz u punom zaslonu 'F'). Također nisu izravno podržani akordi (implicitno su prikazani u pogledu `ShortcutContext` kao prečac + <typing 1>).

2.1.6. Primjer uporabe

Za primjer ću uzeti uporabu alata koja je meni osobno bila korisna. Tokom dana sam normalno koristio računalo i imao sam u pozadini uključen alat za analizu prečaca. Tu i tamo bih provjerio što ima novoga u pogledu `ShortcutPairs`. Primijetio sam da se kombinacija 'Alt-D', 'Ctrl-C' često pojavljivala prilikom korištenja web-preglednika (Slika 2.2). Konceptualno ta kombinacija prečaca kopira URL trenutne stranice, no u stvarnosti prvo se fokusira adresna traku, a potom kopira tekst odnosno URL trenutne stranice. Korisna kombinacija, ali ima nuspojavu prebacivanja fokusa. Često kopiram URL i zalijepim ga u neku drugu aplikaciju, ali po povratku u web-preglednik, fokus je na adresnoj traci, a ne na sadržaju web-stranice. Zbog toga bi bilo korisno imati prečac koji direktno kopira URL trenutne stranice. *Firefox* nema ugrađenu mogućnost dodavanja i mijenjanja prečaca, no

postoje proširenja (poput *Vimium*-a [4]) koja omogućuju korisniku potpunu kontrolu nad prečacima.



Slika 2.2 ShortcutPairs vizualizacija s prikazom para 'Alt-D' i 'Ctrl-C'

2.1.7. Mogućnosti nadogradnje

Korisno bi bilo dodati funkcionalnost izvoza podataka kako bi se oni mogli analizirati u drugim alatima. Izvoz u standardne formate nije kompliciran.

Najveća nadogradnja bi bila da se alatu omogući bogatiji pristup informacijama o trenutnoj aplikaciji. Primjerice, da alat zna koji gumb na alatnoj traci je korisnik kliknuo, koja je akcija i koji je tipkovni prečac za tu akciju. Očekujem da se nešto od toga može dobiti koristeći API za pristupačnost (isti API koji koriste čitači zaslona koji slabovidnim osobama omogućuju korištenje računala). No kvaliteta ovog pristupa jako ovisi o kvaliteti sučelja pristupačnosti koje pojedina aplikacija nudi. Praktičnost implementacije ove funkcionalnosti nisam istražio.

Zaključak

Dinamička prilagodba tipkovnica odnosno optimizacija tipkovnih prečaca da budu efikasni i zadovoljavajući za korištenje je težak problem. Ovaj rad postavlja temelj za daljnja istraživanja: alat koji neprimjetno prikuplja podatke o tipkovnim prečacima i kontekstu u kojem se nalaze te nudi osnovne analize prikupljenih podataka. Također se postavlja ideja za daljnji razvoj alata temeljena na API-jima koje nude operacijski sustavi.

Dolaskom na snagu Europskog zakona o pristupačnosti 28. lipnja 2025. [5] očekuje se porast pristupačnih aplikacija i usluga, a samim time i primjenjivost budućih inačica alata zasnovanom na API-ju za pristupačnost.

Literatura

- [1] Tak, S., Westendorp P., van Rooij, I., *Satisficing and the Use of Keyboard Shortcuts: Being Good Enough Is Enough?*, *Interacting with Computers*, 25, 5, (2013), str. 404-416.
- [2] *Keyboard shortcut*, Wikipedia. Poveznica: https://en.wikipedia.org/wiki/Keyboard_shortcut; pristupljeno 13. lipnja 2024.
- [3] *Access keys*, Microsoft Learn. Poveznica: <https://learn.microsoft.com/en-us/windows/apps/design/input/access-keys>; pristupljeno: 14. lipnja 2024.
- [4] *Vimium*. Poveznica: <https://vimium.github.io/>; pristupljeno: 14. lipnja 2024.
- [5] *Pristupačnost proizvoda i usluga*, EUR-Lex. Poveznica: <https://eur-lex.europa.eu/HR/legal-content/summary/accessibility-of-products-and-services.html>; pristupljeno: 14. lipnja 2024.
- [6] *Keyboard Manager utility*, Microsoft Learn. Poveznica: <https://learn.microsoft.com/en-gb/windows/powertoys/keyboard-manager>; pristupljeno: 14. lipnja 2024.
- [7] *AutoHotkey*. Poveznica: <https://www.autohotkey.com/>; pristupljeno 14. lipnja 2024.

Sažetak

Tipkovnice i dinamička prilagodba

Rad daje repliku na prijašnja istraživanja o zadovoljivosti i korištenju tipkovnih prečaca. Opisuje funkcionalnost i razvoj alata za prikupljanje podataka i mjerenja tipkovnih prečaca. Ručnom analizom podataka može se doći do rješenja koja optimiziraju korištenje prečaca i ubrzavaju svakodnevni rad na računalu. Također se daje ideja za upotrebu API-ja za pristupačnost u nekonvencionalne svrhe prikupljanja podataka.

tipkovnice; tipkovni prečaci; interakcija čovjeka i računala; vizualizacija; pristupačnost

Summary

Keyboards and dynamic customization

This thesis replies to previous research into satisficing and the use of keyboard shortcuts. The functionality and development of a tool to collect data and measuring keyboard shortcuts is given. By manually analysing the data, it's possible to optimize to use of keyboard shortcuts and speed up everyday computer use. The idea for using accessibility API for unconventional uses of data gathering is given.

keyboard; keyboard shortcuts; human-computer interaction; visualization; accessibility

Privitak

Instalacija programske podrške

Alat podržava Windows 10 i novije.

1. Instalirati .NET 8 razvojno okruženje. Dostupno na <https://get.dot.net/>.
2.

```
cd KeyboardTester  
dotnet run --project KeyboardTester.Game.Tests
```