

# gms

Generic Mouse Gestures

Dokumentacija

20. junij 2011

**Avtorji:**

Anže Pečar, Miha Zidar, Matic Potočnik

# Kazalo

<b>1</b>	<b>Uvod</b>	<b>2</b>
1.1	O programu . . . . .	2
<b>2</b>	<b>Navodila za uporabo</b>	<b>3</b>
2.1	Namestitev . . . . .	3
2.1.1	Odvisnosti . . . . .	3
2.1.2	Namestitev . . . . .	3
2.1.3	Posodabljanje . . . . .	3
2.2	Uporaba . . . . .	3
2.2.1	Zagon . . . . .	3
2.2.2	Uporaba . . . . .	4
2.2.3	Nastavitve . . . . .	5
<b>3</b>	<b>Tehnična zasnova</b>	<b>6</b>
<b>4</b>	<b>Nadaljno delo</b>	<b>8</b>

# Poglavje 1

## Uvod

### 1.1 O programu

`gms` je program za upravljanje računalnika s pomočjo miškinih gest. Ustvarjen je bil v sklopu seminarske naloge za predmet `Operacijski Sistemi 2` na univerzitetnem programu Fakultete za računalništvo in informatiko v Ljubljani. Repozitorij z izvorno kodo je na voljo na: <https://github.com/Smotko/gms>

## Poglavje 2

# Navodila za uporabo

### 2.1 Namestitev

Prikazana je namestitev na sistem `Ubuntu 11.04 Natty Narwhal`.

#### 2.1.1 Odvisnosti

Program ima nekaj odvisnosti, ki niso pokrite v privzeti namestitvi Ubuntu 11.04. To popravimo z namestitvijo paketov `python-xlib`, `lineakd`, ter `git`:

```
sudo apt-get install python-xlib lineakd git
```

#### 2.1.2 Namestitev

Kot je navada v Ubuntuju, program naložimo v mapo `/usr/local/bin`:

```
cd /usr/local/bin
sudo git clone git://github.com/Smotko/gms.git
```

Če program želimo zaganjati iz kjerkoli, moramo njegovo mapo dodati v iskalno pot. To naredimo tako, da dodamo naslednji vrstici v datoteko `~/.bash_rc`:

```
PATH=$PATH:/usr/local/bin/gms
export PATH
```

#### 2.1.3 Posodabljanje

Če želimo program kdaj posodobiti, v terminalu zaženemo:

```
cd /usr/local/bin/gms
sudo git pull
```

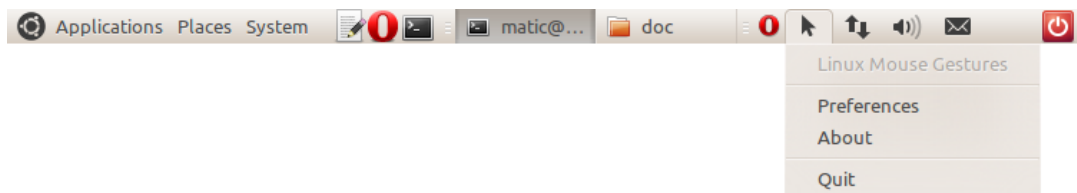
### 2.2 Uporaba

#### 2.2.1 Zagon

Za zagon preprosto zaženemo `gms` v terminalu, po želji pa si lahko ustvarimo tudi bližnjico v kakšnem izmed menijev, ali pa program dodamo med programe, ki se zaženejo ob zagonu operacijskega sistema.

## 2.2.2 Uporaba

Ko program zaženemo, se v zgornji vrstici v kotičku za obveščanju pojavi ikona.



S klikom na ikono se odpre meni, s katerim lahko pridemo do nastavitvev, podatkov o programu, ali pa zapremo program.

Program nato uporabljamo prek miškinih gest. Podprte so naslednje geste:

- Gor-Levo



- Desno



- Dol-Desno



- Gor



- Dol



- Levo



- Gor-Desno



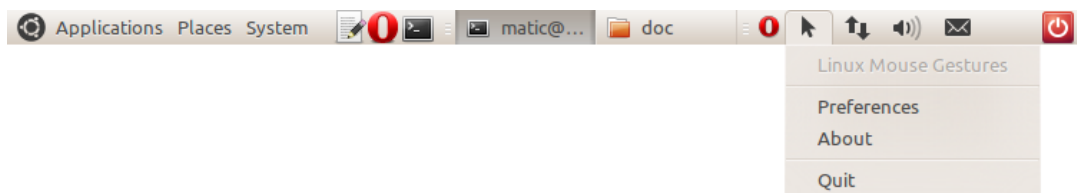
- Dol-Levo



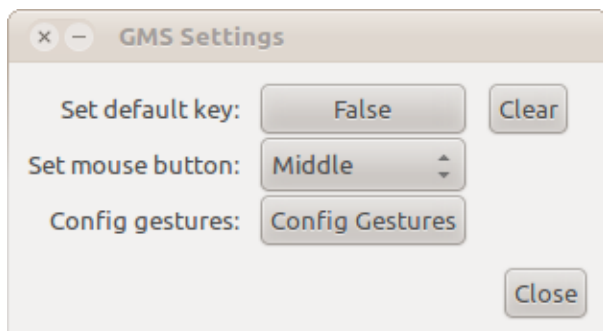
Akcije, ki se izvedejo po prepoznavi gest so nastavljive (glej 2.2.3), obsegajo pa povsem poljubne pritiske tipk tipkovnice. Tako lahko upravljamo z vsakim programom, ki podpira kake bližnjice na tipkovnici, nekatere bližnjice (npr. za zapiranje programov), pa so podprte že s strani operacijskega sistema.

### 2.2.3 Nastavitve

Do nastavitve pridemo s klikom na ikono v zgornji vrstici in izbiramo **Preferences**



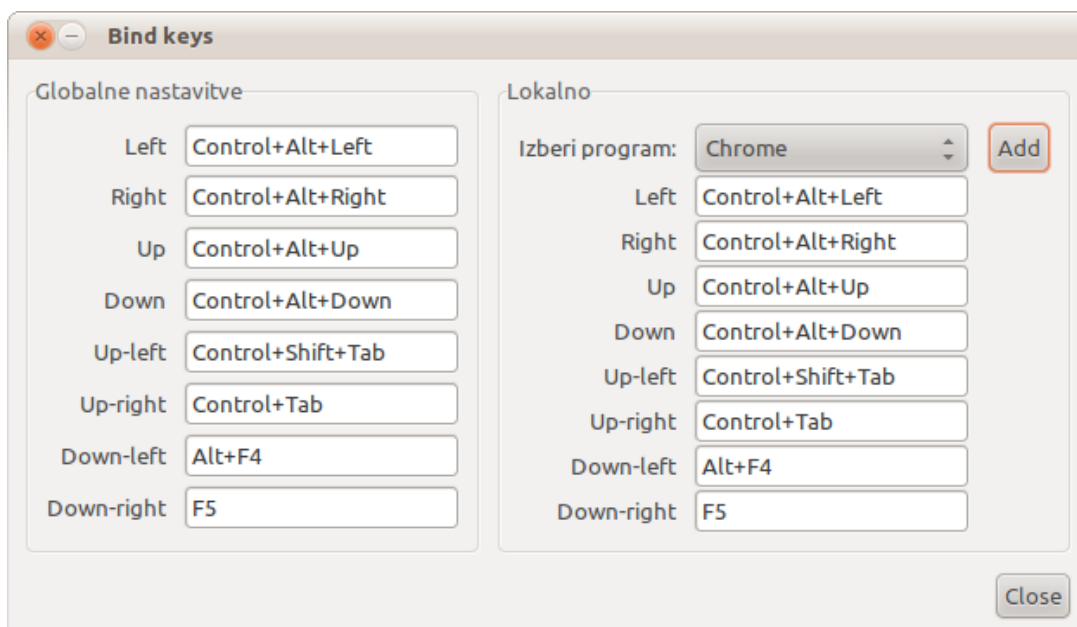
V oknu z nastavitvami imamo naslednje možnosti:



S klikom na gumb ob **Set default key** nastavimo, katero tipko je potrebno držati med izvajanjem geste, pri čemer privzeta nastavev **False** pomeni, nobene.

S **Set mouse button** nastavimo, katero miškino tipko je potrebno držati med izvajanjem geste.

S klikom na **Config gestures** pa se odpre okno, z nastavitvami akcij.



V tem oknu nastavimo, katera akcija se bo izvedla ob kateri gesti, lahko pa dodamo tudi posebne nastavitve za posamezne programe, ki imajo morda svoje bližnjice, katere jih želimo uporabiti.

## Poglavje 3

# Tehnična zasnova

Program gms je napisan v programskem jeziku `Python`. Za uporabniški vmesnik nastavitvev je uporabljeno ogrodje `GTK`, za dodajanje ikone v kotichek za obveščanje je uporabiljen `IndicatorApplet`, za zajemanje premikov miške in tipk tipkovnice in tudi za izvajanje akcij, pa smo uporabili knjižnico `Xlib`. Za prepoznavanje miškinih gest smo uporabili lasten algoritem.

Za `Python` smo se odločili, ker kljub svoji visoko-nivojski naravi, prek množice knjižnic, omogoča dostop do nizko-nivojskih dogajanj operacijskega sistema. Knjižnica, ki smo jo za implementacijo najbolj potrebovali je `Xlib` – z njo je implementirano tako zajemanje miške in tipkovnice (zajemanje gibanja in pritiska tipk), kot tudi izvajanje akcij (pošiljanje tipk tipkovnice).

V modulu `pyxhook` s pomočjo modula `record`, ki se nahaja v knjižnici `XLib`, lovimo pritisnjene tipke. Prav tako imamo v `pyxhook` izdelan vmesnik za priklopljanje (hookanje) dogodkov, ki ga izkoriščamo v modulu `gms`. `gms` je glavni modul naše aplikacije, saj se tu priklopimo na dogodke `keyboard button down`, `keyboard button up`, `mouse button down`, `mouse button up`, ki so definirani v modulu `pyxhook` in na podlagi teh dogodkov preverjamo geste in izvajamo akcije. Ko se en od teh dogodkov sproži dobimo povratno informacijo, na katero se odzovemo tako, da preverimo, če ima pritisnjena tipka pomen v uporabniških nastavitvah. Če je bila npr. pritisnjena tipka za začetek geste, začnemo shranjevati pozicijo miške v namenski seznam koordinat, če pa je bila pritisnjena tipka za konec geste pa preverimo, če koordinate v seznamu predstavljajo neko gesto. Če je algoritem za detekcijo gest prepoznal izvedeno gesto preda podatek o izvedeni gesti modulu za izvedbo akcij. Modul za izvedbo akcij prebere akcijo iz nastavitvev in jo izvede s pomočjo knjižnice `xsendkeys`.

Nastavitve shranjujemo v `sqlite 3` podatkovno bazo, do nje pa dostopamo preko namenskega modula `models`. V modulu `models` imamo napisan preprost vmesnik za dostop do nastavitvev npr. `set`, `get`, `delete` metode za delo z nastavitvami in namenske metode za delo s shranjenimi gestami. Za modul `models` smo tudi napisali preprost razred `dbutils`, ki abstrahira delo s podatkovno bazo tako, da nam v modulu `models` ni bilo potrebno skrbeti za povezovanje na podatkovno bazo. `dbutils` nam prav tako omogoča, da preprosto zamenjamo podatkovno bazo `sqlite 3`, s katero drugo, če bi bilo to potrebno.

Grafični vmesnik smo izdelali v ogrodju `GTK`, razdeljen pa je v dva modula. Prvi skrbi za prikaz globalnih nastavitvev, v drugem pa je napisana logika za dodajanje in spreminjanje gest. Oba modula komunicirata z modulom `models`, preko katerega prikazujeta in shranjujeta nastavitve.





## Poglavje 4

# Nadaljno delo

Program se kljub temu, da je že pripravljen za uporabo, širi v praktično vse smeri.

Na eni strani lahko napišemo boljši algoritem za prepoznavanje gest, ki bo prepoznaval tudi višje-nivojske geste (npr. gesto `Gor-Dol-Gor-Levo-Desno`). Možno bi bilo zajemati podatke na nižjem nivoju, kar bi omogočalo da preprečimo programu, da bi prejel miškin klik in izvedel privzeto akcijo. Lahko bi poboljšali tudi izvajanje gest, ki so trenutno omejene na pošiljanje tipk tipkovnice, ne pošiljajo pa signalov operacijskemu sistemu naj, npr. minimizira neko okno, ali zažene nek program ali skripto. Lahko bi tudi izboljšali uporabniško izkušnjo tako, da bi uporabnik videl grafično sled miške, medtem, ko bi izvajal gesto.