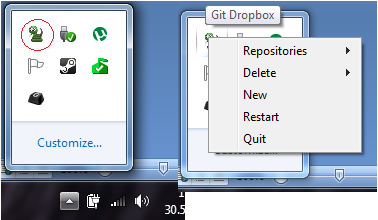
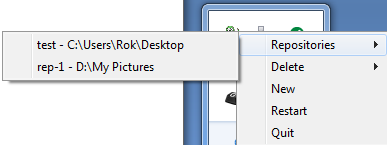
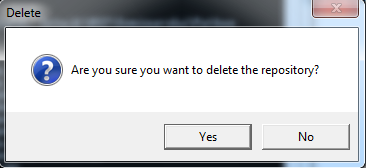
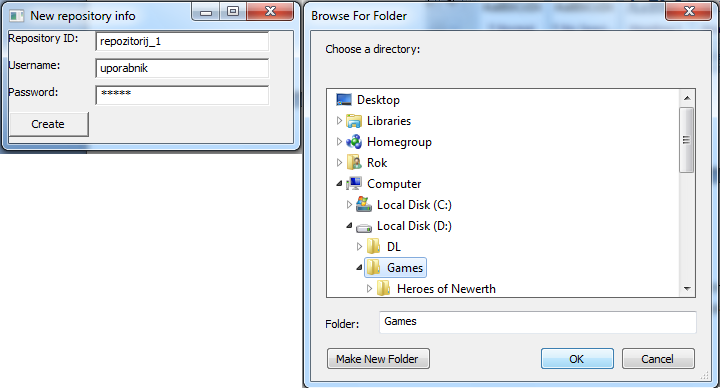
# Windows system tray

Za operacijske sisteme Windows ima git-dropbox realiziran tudi grafični vmesnik v obliki ikone v system trayu.

## Interakcija



Vmesnik ima naslednje opcije

* Repositories: v podmeniju izpiše seznam repozitorijev, klik na enega od njih pa odpre novo okno windows explorerja na tisti lokaciji. 
* Delete: ponovno se odpre enak podmenij kot pri prejšnji opciji, le da tokrat klik na željen repozitorij le tega izbriše. Pred izbrisom se pojavi tudi klasično YES/NO potrditveno okence. 
* New: opcija za kreiranje novega repozitorija. Tu se pojavita dve okni; najprej okno s tremi polji za vnos, ki sprešuje po imenu repozitorija, uporabniškem imenu, in geslu, nato pa klasičen windows file browser. Po uspešnem vnosu podatkov se repozitorij kreira. 
* Restart: resetira aplikacijo.
* Quit: zapre aplikacijo.

## Ozadje

Sedaj ko smo opisali kako git-dropbox zgleda na zunaj, pa si lahko pogledamo kako deluje.

Ob zagonu programa se iz datoteke »repositories.dataz« preberejo podatki o vseh trenutno obstoječih repozitorijih. Če datoteke ni se kreira nova prazna datoteka.

Za vsak tako pridobljen repozitorij se potem zažene nov thread, ki na določeni lokaciji posluša za kakršnekoli spremembe datotek. Hkrati se iz podatkov o repozitorijih kreirajo potrebne podatkovne strukture za menije v system trayu. Nato se požene del kode ki naredi in operira s celotnim system trayem. Velika večina te kode je skopirana iz interneta in spremenjena za naše potrebe, zato je tudi hkrati edini večji kos kode v tem projektu ki ni dobro pokomentiran.

V tej točki program čaka na interakcijo z uporabnikom, threadi pa poslušajo svoje direktorije za spremembe. V primeru spremembe thread pokliče primerne funkcije za sinhronizacijo repozitorijev, ki so obravnavane v drugem delu te dokumentacije.

Ob izhodu iz aplikacije se najprej ugasne system tray ikona, nato pa se požene metoda ki konča vse threade.

## Izzivi

V tem delu bom izpostavil nekaj tehničnih izzivov, s katerimi sem imel opravka.

Za še največji izziv se je izkazalo sprotno updatanje seznama repozitorijev v meniju po dodajanju oziroma brisanju repozitorijev. Edini način, ki sem ga našel, je da se ob vsaki spremembi seznama repozitorijev celoten system tray del programa ugasne in ponovno zažene.

Tak način »brute force updatanja« pa ni bil tako slab, saj tako lahko hkrati tudi ustavimo in ponovno zaženemo vse threade – z novimi podatki. Tako bo thread, ki nadzira repozitorij ki smo ga ravnokar odstranili, takoj ugasnjen, hkrati pa bo ravnokar kreiran repozitorij takoj dobil svoj thread.

Naslednji izziv se je pojavil zaradi načina implementacije threadov. Izkazalo se je, da so threadi z neskončno zanko zelo odporni na konvencionalne metode ugašanja. Tudi tu je bila na koncu implementirana bolj »brute force« metoda, ki povzroči izhod iz neskončne zanke, nato pa se thread izvede do konca in ustavi.

Na koncu pa je tu še edini trenutno znan bug tega dela seminarske: kakršnokoli resetiranje system tray ikone (torej kreiranje, brisanje ali pa reset opcija) povzroči crash pythona če poskušaš nato odpreti kakšnega od repozitorijev prek system tray menija. Ker sta ta dva dela kode (reset in odpiranje folderja) dve popolnoma različni in ločeni kodi resnično ne vem kaj bi ta bug lahko povzročalo, hkrati pa mi za nadaljnje testiranje zmanjkuje časa.

## Zahteve

Za delovanje potrebujemo Python 2.7.1 in dve knjižnici za python: pywin in wxPython, povezave do uradnih strani pa so v nadaljevanju.

Python 2.7.1 - [link](http://www.python.org/download/releases/2.7.1/)

Pywin - [link](http://sourceforge.net/projects/pywin32/files/pywin32/Build216/)

wxPython - [link](http://www.wxpython.org/)